Максим Кульгин

ДЛЯ ПРОФЕССИОНАЛОВ

КОМПЬЮТЕРНЫЕ СЕТИ

Практика построения

2-Е ИЗДАНИЕ

ПИТЕР®

Москва · Санкт-Петербург · Нижний Новгород · Воронеж
Ростов-на-Дону · Екатеринбург · Самара
Киев · Харьков · Минск
2003

ББК 32.988.02
УДК 681.324
К 90

К 90 **Компьютерные сети. Практика построения. Для профессионалов. 2-е изд. /** М. В. Кульгин. — СПб.: Питер, 2003. — 462 с.: ил.

ISBN 5-94723-563-3

При создании и обслуживании любой сети возникает масса больших и малых проблем, о решении которых ничего не говорится в теоретических фолиантах. Оказывается, многие из них можно решить, применяя совокупность нескольких технологий или внедряя некоторые технологии там, где они в соответствии со своей целевой функцией не могут быть использованы. В книге основной упор сделан именно на практическую сторону построения и обслуживания сетей, рассмотрены те устройства и технологии, которые на сегодняшний день являются базовыми и основополагающими: маршрутизаторы компании Cisco Systems и их настройка, технология организации очередей, трансляции адресов, построения защитного экрана, настройка протоколов маршрутизации, построение защищенных виртуальных сетей и т. п. Книга рассчитана не только на начинающих администраторов, которые хотят разобраться в основных принципах работы с сетевыми программными и аппаратными комплексами, — опытные специалисты также найдут здесь немало ценной информации.

ББК 32.968.02
УДК 681.324

ISBN 5-94723-563-3

Краткое содержание

От автора 9
Введение 10
Глава 1. Начальная настройка маршрутизатора Cisco Systems 14
Глава 2. Настройка протокола IP 72
Глава 3. Дальнейшая настройка маршрутизатора 126
Глава 4. Протокол ICMP 139
Глава 5. Настройка протокола TCP 189
Глава 6. Трансляция адресов и настройка очередей 238
Глава 7. Настройка протоколов маршрутизации 273
Глава 8. Технология Frame Relay в построении распределенной сети . . 320
Глава 9. Настройка базовой безопасности 344
Глава 10. Построение виртуальных частных сетей 374
Приложение А. Протоколы 442
Алфавитный указатель 447

Содержание

От автора **9**

Введение **10**

Организация книги 11
От издательства 13

Глава 1. Начальная настройка маршрутизатора Cisco Systems . . . **14**

Общие сведения 14
Аппаратная архитектура маршрутизаторов Cisco Systems 17
Введение в Cisco IOS 21
Начальная настройка маршрутизатора 22
Пошаговая настройка маршрутизатора 25
Интерфейс командной строки Cisco IOS 27
Команды Cisco IOS 34
Работа с аппаратными интерфейсами маршрутизатора 40
Терминальные порты 50
Проверка аппаратных ресурсов маршрутизатора 52
Таблицы, буферы и журналы маршрутизатора 53
Синхронизация по времени 58
Файлы конфигурации маршрутизатора 59
Имена и заголовки 66
Обеспечение безопасности доступа к маршрутизатору 68

Глава 2. Настройка протокола IP **72**

Протокол IP 72
Классы адресов и их маски 76
Подсети 82
Маска подсети 83
Практика выделения IP-подсетей 93
Маска подсети переменной длины 96
Применение маски подсети переменной длины 100
Технология CIDR 108
Пример использования технологии CIDR 114
Настройка протокола IP на маршрутизаторе Cisco Systems 116
Протокол ARP 119
Возможность использования протокола ARP для сетевых атак 123

Глава 3. Дальнейшая настройка маршрутизатора 126
Разрешение адресов маршрутизатором 126
Работа с интерфейсами и портами маршрутизатора 128
Работа с маршрутизатором через сеанс Telnet 133
Протокол CDP 135
Работа со встроенным отладчиком 137

Глава 4. Протокол ICMP 139
Введение 139
Перехват информации в сетях 141
Сообщения протокола ICMP 147
Сообщение ICMP echo request 151
Сообщение ICMP source quench (подавление источника) 160
Сообщение ICMP destination unreachable 164
Сообщение ICMP redirect 171
Информационные запросы протокола ICMP 176
Сообщение ICMP parameter problem 180
Сообщение ICMP time exceeded 181
Фрагментация дейтаграмм IP 183

Глава 5. Настройка протокола TCP 189
Введение в протокол TCP 189
Атака TCP SYN flooding 194
Блок управления передачей протокола TCP 204
Скользящее окно 205
Узкие места в сети 208
Пропускная способность TCP 210
Изменение размера окна на маршрутизаторе 214
Управление потоком 214
Стратегии отправления и приема данных 217
Выборочное подтверждение SACK 219
Задержанное подтверждение 221
Таймер повторной передачи 221
Временные отметки 224
Быстрая повторная передача 226
Контроль над перегрузками 228
Медленный старт 229
Механизм TCP Path MTU 234

Глава 6. Трансляция адресов и настройка очередей 238
Трансляция адресов 238
Настройка очередей на маршрутизаторе 246
Очередь типа FIFO 246
Качество обслуживания в IP-сетях 249
Очередь RED 252
Настройка очередей RED и WRED 255
Очереди с приоритетами 256

Настройка очередей с приоритетами для маршрутизаторов Cisco 258
Настраиваемые очереди 261
Очереди на основе классов 267
Настройка взвешенной справедливой очереди WFQ 267
Рекомендации по выбору стратегии очередей 271

Глава 7. Настройка протоколов маршрутизации 273

Алгоритмы маршрутизации 273
Протокол маршрутизации RIP 276
Счет до бесконечности 289
Расщепление горизонта 290
Мгновенное обновление 291
Пример подключения объектов к сети Интернет 306

Глава 8. Технология Frame Relay в построении распределенной сети 320

Общие сведения о технологии Frame Relay 320
Настройка Frame Relay на маршрутизаторах Cisco Systems 326
Управление трафиком в сети Frame Relay 333

Глава 9. Настройка базовой безопасности 344

Списки управления доступом Cisco IOS 344
Защита от атак address spoofing 358
Безопасность удаленного доступа 365
Практическая настройка технологии Cisco AAA 369

Глава 10. Построение виртуальных частных сетей 374

Введение 374
Оборудование удаленных объектов 380
Технологии виртуальных частных сетей 382
Практика подключения объекта 384
Основные принципы работы протокола PPTP 389
Методы шифрования информации 401
Примеры настройки защитного экрана 408
Анализ трафика PPTP на сетевом уровне 414
Процедура настройки туннеля 427

Приложение А. Протоколы 442

Алфавитный указатель 447

От автора

О чем эта книга? Прежде всего, это попытка поделиться практическим опытом настройки компонентов компьютерной сети. В книге рассмотрены только те устройства и технологии, которые являются базовыми и основополагающими: маршрутизаторы компании Cisco Systems и их настройка, технология организации очередей, трансляции адресов, построения защитного экрана, настройка протоколов маршрутизации, построение защищенных виртуальных сетей и т. п. Я сознательно старался избежать обширных теоретических рассуждений, переходя сразу к практической работе с этими технологиями. Читатель держит в руках второе издание книги, отличающееся двумя новыми главами, которые, уверен, будут так же интересны, как и остальной материал.

С уважением
Максим Кульгин.
mk@alternativa.spb.ru

Введение

У меня, как и у многих специалистов в области сетевых компьютерных технологий, сложилось вполне определенное мнение о представленной сегодня на книжном рынке литературе по этой тематике. Очень много изданий по программному обеспечению. Гораздо меньше книг по аппаратному сетевому обеспечению. И очень редко появляются книги, в которых даются какие-либо практические рекомендации по внедрению и эксплуатации программного и аппаратного комплексов. Учитывая дефицитность подобной литературы, мне захотелось поделиться со своими коллегами по «компьютерному цеху» теми практическими знаниями, которые я приобрел в процессе создания распределенной компьютерной сети коммерческой фирмы, в которой работаю. Почему я посчитал необходимым представить материал этой книги на широкое обсуждение? Считаю, что он будет интересен и полезен в практической деятельности всех специалистов, работающих с распределенными сетями. Свое утверждение строю на том факте, что подавляющее большинство компьютерных сетей активно работающих коммерческих фирм имеют практически одинаковую конфигурацию и на них возложены схожие задачи. И что очень важно, начиная работу по созданию корпоративной сети или модернизируя старую структуру, хочется или не хочется, специалисту придется пройти примерно тот же путь, что уже одолел я. А зачем «наступать» на те же «грабли», если можно этого избежать?! Тем, у кого сеть уже работает, также не будет лишним попробовать у себя те технологии, которые работают в сети моей компании.

При создании любой сети возникает масса больших и малых проблем, о способах решениях которых ничего не говорится в теоретических фолиантах. Ведь многие проблемы удается решить, применяя совокупность нескольких технологий или даже внедряя некоторые технологии туда, где они в соответствии со своей целевой функцией не могут быть применены. Описание способов решения некоторых проблем можно найти в этой книге. При этом упор сделан на решение технических вопросов, которые возникали при построении и ежедневном обслуживании сети. Особое внимание уделено вопросам организации связи. За несколько лет практической работы набралось много таких решений, что и позволило создать эту книгу.

Я не ставил своей целью описать как можно больше технологий. Думаю, что все сетевые специалисты согласятся с малой ценностью книг, в которых пусть и коротко, но описано все, что хоть в малой степени можно причислить к сетевым технологиям. В нашей стране уже встал на ноги и заявил о себе большой отряд сетевых специалистов: менеджеров IT, сетевых администраторов и анали-

тиков. И для них такого рода пособия — уже давно пройденный этап. А при подготовке новых специалистов эти книги приносят больше вреда, чем пользы. Молодой специалист, начитавшись такой литературы, думает, что он уже изучил все тонкости своей профессии, и готов на практике воплотить в жизнь приобретенные знания. И очень сильным бывает разочарование, когда он сталкивается с реальными проблемами, разрешение которых требует от него глубоких знаний. Кроме того, можно с уверенностью говорить о том, что в нашей стране еще не сформировалась школа подготовки специалистов в этой области. По этой причине считаю, что издание данной книги (а точнее, ее второй редакции с двумя новыми главами) своевременно и должно принести большую пользу как начинающим, так и зрелым специалистам.

Организация книги

Книга содержит десять глав. Первая и третья главы посвящены вопросам настройки маршрутизаторов. Для наглядности рассматриваются устройства одного производителя — компании Cisco Systems. Выбор именно этой компании связан с тем, что ее маршрутизаторы занимают лидирующее положение на мировом рынке. Необходимость включения этого материала в книгу продиктована тем, что, к сожалению, русскоязычную литературу по маршрутизаторам найти достаточно сложно. Материал первых глав поможет специалисту в начальный период работы с маршрутизатором. Возможно, читатель заметит некоторую странность в изложении материала, а именно искусственный разрыв в описании процесса настройки и включение в этот разрыв второй главы, посвященной настройке протокола IP. Это сделано не случайно и связано с использованием в описании дальнейшей настройки маршрутизаторов некоторых параметров протокола IP. Во второй главе кроме практики настройки адресного пространства организации дано изложение практических методов настройки протоколов IP и ARP на маршрутизаторе. Представлено описание атаки на протокол ARP и методов защиты от нее.

Глава 4 полностью посвящена рассмотрению работы протокола ICMP. Несмотря на то что протокол скорее относится к разряду вспомогательных, с его помощью можно получить много интересной информации о сети. Причем зачастую именно этот протокол используется для сбора информации людьми, которые в дальнейшем могут попытаться осуществить проникновение в сеть. Я постарался сделать материал более понятным, приведя в качестве примеров сетевые перехваты наиболее интересных пакетов. Если администратор запустит в своей сети анализатор трафика, настроенный на перехват пакетов этого протокола, он, скорее всего, будет удивлен их количеством (даже не говоря о таких известных командах, как ping или tracert). Например, серверы Microsoft Exchange 2000 используют это протокол для проверки работоспособности контроллеров домена Active Directory.

В пятой главе дано расширенное описание протокола TCP. Некорректная работа почтовой системы Microsoft Exchange в распределенной сети с «облаком»

Frame Relay стала первопричиной для автора обратить самое пристальное внимание на работу этого протокола в сетях Microsoft. Последующий детальный анализ сбоев показал, что все дело вовсе не в почтовой системе, а в громоздком механизме передачи информации. В частности, большинство попыток скопировать Проводником большой объем информации (несколько сотен мегабайтов) с удаленного сервера по сети Frame Relay (CIR = 32 Кбит/с) заканчивались ошибками при передаче. Эти ошибки возникали из-за слишком больших накладных расходов на служебную информацию. Убежденность автора в необходимости подробного рассмотрения протокола TCP усилил факт существования атаки TCP SYN flooding.

Шестая глава содержит описание технологий трансляции адресов (NAT) и настройки адресов на маршрутизаторах, а также обозрение различных технологий организации очередей и практики их применения на маршрутизаторах.

Седьмая глава посвящена вопросам маршрутизации в IP-сетях. Здесь очень подробно рассмотрен протокол RIP (Routing Information Protocol). Освещена практика его конфигурирования на маршрутизаторах. Выбор этого протокола связан с его большой популярностью среди отечественных сетевых специалистов.

Восьмая глава содержит сведения о технологии Frame Relay, спрос на которую в настоящее время все более возрастает. Технология Frame Relay позволяет экономить до 50% средств по сравнению с затратами на аренду выделенных каналов связи. При правильной настройке качество соединения практически не отличается от качества связи по выделенным каналам. По этой причине технология Frame Relay все чаще используется для коммуникации между удаленными объектами, построения корпоративных сетей, доступа в Интернет. В данной главе описаны практические методы настройки этой технологии на граничных маршрутизаторах.

Девятая глава посвящена вопросам базовой безопасности сети. В ней рассмотрены несколько технологий. Большое внимание уделено практическим вопросам создания списков доступа Cisco ACL, которые являются ключевой возможностью операционной системы Cisco IOS. Основное предназначение списков доступа заключается в создании фильтра безопасности, направленного на блокирование трафика, входящего в сеть или покидающего сегмент сети. Однако по мере развития операционной системы IOS компания Cisco Systems расширила синтаксис команд формирования списков доступа для решения и других задач.

Любая крупная сеть предоставляет пользователям возможность задействовать удаленный доступ, в связи с чем очень важно создать систему безопасности удаленного доступа. Этому вопросу в девятой главе также уделено внимание.

Десятая, последняя глава в книге, рассматривает тонкости построения защищенных виртуальных частных сетей (VPN) на основе протокола PPTP и решений компании Microsoft. Данная глава, как и четвертая, появились во втором издании книги, которую читатель сейчас держит в руках. Сегодня наблюдается все больший интерес к построению подобных сетей, так как у многих организаций появляются филиалы, подключение которых традиционными способами представляется либо невозможным, либо непрактичным с точки зрения затрат. В такой ситуации администратор может воспользоваться уже проверенными решениями, позволяющими создать защищенный «туннель» к удаленному филиалу,

проходящий через сеть Интернет. Материал этой главы акцентирован на практическом применении протокола PPTP, поддержка которого входит в операционные системы компании Microsoft. Читатель, помимо помощи по настройке туннеля, найдет также детальный анализ процессов, происходящих на сетевом уровне, что несомненно поможет сделать рассматриваемую технологию более прозрачной для понимания.

В дополнение книга включает в себя приложение, в котором приведены номера портов наиболее популярных программ. Эта информация будет полезной при настройке граничных или внутренних маршрутизаторов (защитных экранов) на блокирование трафика отдельных протоколов.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу электронной почты comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

Подробную информацию о наших книгах вы найдете на web-сайте издательства http://www.piter.com.

1 Начальная настройка маршрутизатора Cisco Systems

Общие сведения

Очень часто в компьютерной литературе можно встретить следующее определение маршрутизатора (английское название — router): «маршрутизатор — это устройство сетевого уровня эталонной модели OSI, использующее одну или более метрик для определения оптимального пути передачи сетевого трафика на основании информации сетевого уровня». Из этого следует, что маршрутизатор прежде всего необходим для выбора дальнейшего пути данных, которые отсылаются в распределенную сеть.

Пользователи для отправки своих данных в сеть указывают лишь адрес абонента. Эти данные уходят в сеть и в точках с разветвлением маршрутов поступают на маршрутизаторы, которые как раз и служат для выбора дальнейшего пути. При этом маршрутизатор выбирает оптимальный путь. Оптимальность пути определяется количественными характеристиками, которые называются *метриками*. Лучший путь — это путь с метрикой, которая в данном конкретном случае является наиболее подходящей. В метрике могут учитываться несколько показателей, например длина пути, время прохождения и т. д.

Рисунок 1.1 показывает обобщенную схему использования маршрутизатора в сети (рисунок позаимствован с сайта компании Cisco Systems, как и некоторые другие рисунке в этой книге). Здесь видно, что для связи маршрутизатора с Интернетом применяется специальное устройство — модем, который с одной стороны подключается к маршрутизатору, а с другой — к выбранной среде передачи информации (это может быть обычный асинхронный модем, подключаемый к городской телефонной сети).

По критерию производительности маршрутизаторы делят на устройства высшего, среднего и низшего классов. Высокопроизводительные маршрутизаторы являются устройствами высшего класса и служат для объединения сетей предприятия. Они поддерживают множество протоколов и интерфейсов, причем не только стандартных. Устройства такого класса могут иметь до 50 портов локальных или глобальных сетей.

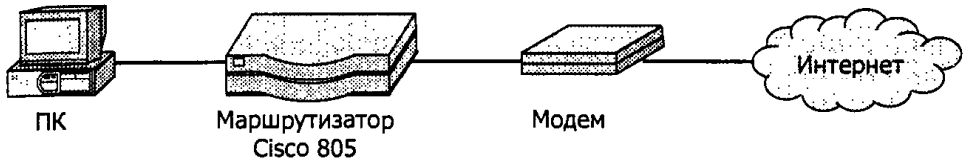


Рис. 1.1. Общая схема подключения маршрутизатора к сети

С помощью маршрутизаторов среднего, промежуточного, класса формируются менее крупные сетевые объединения масштаба предприятия. Стандартная конфигурация включает два-три порта локальных сетей и от четырех до восьми портов глобальной сети. Такие маршрутизаторы поддерживают наиболее распространенные протоколы маршрутизации и транспортные протоколы.

На рис. 1.2 показан общий вариант подключения маршрутизатора к глобальной сети через модем.

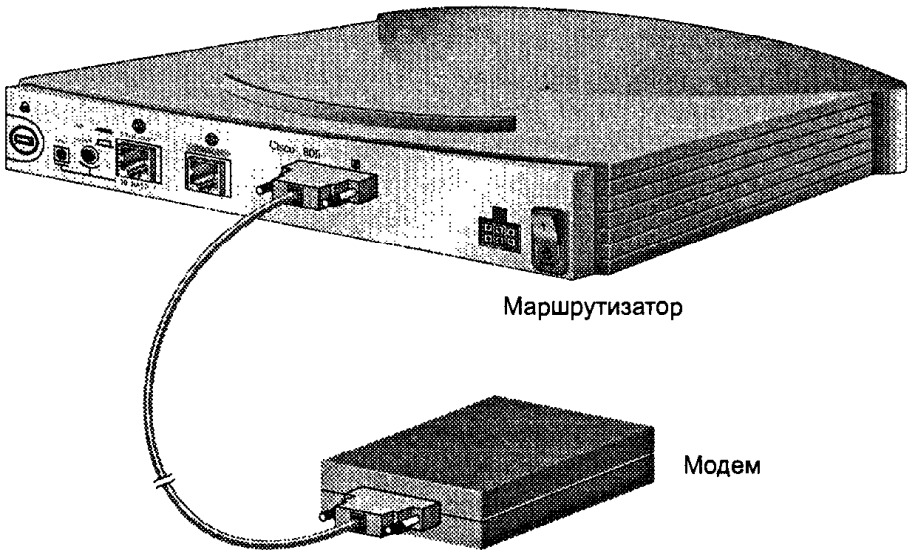


Рис. 1.2. Подключение маршрутизатора к модему

В рассматриваемом примере модем подключается к маршрутизатору Cisco 805 при помощи специального кабеля, который следует заказывать отдельно. И хотя на рисунке модем подключен непосредственно к маршрутизатору, часто вместе с модемом производитель поставляет свой кабель, оканчивающийся разъемом интерфейса V.35. В таком случае администратору сети при оформлении заказа на маршрутизатор потребуется заказать кабель с аналогичным разъемом (тут потребуется уточнить тип разъема — male/female, или «папа»/«мама»).

Кроме того, при подключении модема к маршрутизатору следует помнить о максимально возможной длине кабеля, которая зависит от скорости передачи информации и используемого типа подключения. Так, при работе на скорости 2 Мбит/с рекомендуемая длина кабеля не должна превышать 15 метров (при использовании интерфейса V.35).

ПРИМЕЧАНИЕ

Интерфейс V.35 предназначен для последовательной передачи данных в синхронном режиме. Этот режим основывается на постоянной активности канала связи: даже если нет данных для передачи, канал заполнен сигналами синхронизации. Синхронный режим более адаптирован к работе с протоколами верхних уровней. Этому способствует структурированность кадра с данными (начало и конец каждого кадра отмечены синхросимволами). Первоначально интерфейс V.35 был ориентирован на соединение оконечного оборудования (Data Terminal Equipment — DTE) и коммуникационной аппаратуры (Data Communications Equipment — DCE) при скорости до 100 Кбит/с. При этом теоретическая дальность связи составляла 1200 метров. Однако интерфейс способен работать и на более высоких скоростях — до 45 Мбит/с. Рекомендуемый режим — 64 Кбит/с на дальности 600 метров.

Интерфейс основывается на дифференциальном приеме и передаче данных и сигналов синхронизации и несимметричной передаче управляющих сигналов. Физически интерфейс V.35 реализован на 34-контактном разъеме с фиксаторами двух типов. Встречается реализация на 25-контактном разъеме.

Маршрутизаторы низшего класса предназначены для локальных сетей подразделений; они связывают небольшие офисы с сетью предприятия. Типичная конфигурация: один порт локальной сети (обычно Ethernet) и один или два порта глобальной сети, рассчитанных на низкоскоростные выделенные или коммутируемые соединения. Тем не менее подобные маршрутизаторы пользуются большим спросом у администраторов, которым необходимо расширить имеющиеся межсетевые объединения.

Маршрутизаторы для базовых сетей и удаленных офисов имеют разную архитектуру, поскольку к ним предъявляются разные функциональные требования. Маршрутизаторы базовых сетей обязательно должны быть расширяемыми. Маршрутизаторы локальных сетей подразделения, для которых, как правило, заранее устанавливается фиксированная конфигурация портов, содержат только один процессор, управляющий работой трех или четырех интерфейсов.

По определению, основное назначение маршрутизаторов — это выбор оптимального пути следования трафика сети. Процесс маршрутизации можно разделить на два иерархически связанных уровня.

1. Уровень маршрутизации. На этом уровне происходит работа с таблицей маршрутизации. Таблица маршрутизации служит для определения адреса сетевого уровня следующего маршрутизатора или непосредственно получателя. После определения адреса выбирается интерфейс маршрутизатора, через который будут передаваться пакеты. Этот процесс называется *определением маршрута*. Управление таблицей маршрутизации выполняется протоколами маршрутизации.
2. Уровень передачи пакетов. Перед тем как передать пакет, необходимо: проверить контрольную сумму заголовка пакета, определить адрес (канального уровня) получателя пакета и произвести непосредственно отправку пакета с учетом очередности, фрагментации, фильтрации и т. д. Эти действия выполняются на основании команд, поступающих с уровня маршрутизации.

По мнению автора, основные принципы работы маршрутизаторов различных производителей практически совпадают друг с другом. По этой причине автор выбрал для иллюстрации материала маршрутизаторы одного производителя. Знание основополагающих принципов работы этих устройств позволяет управ-

лять фактически любым маршрутизатором. В книге рассматривается оборудование компании Cisco Systems и по возможности приводятся примеры работы с другим оборудованием.

ПРИМЕЧАНИЕ

Продукция Cisco (часто название фирмы на русском языке произносят как «циско» или «киско») распространяется в 90 странах мира. Приобрести ее можно через дистрибьюторов фирмы, реселлеров с добавленной стоимостью и системных интеграторов Cisco. Центральная штаб-квартира компании расположена в Сан-Хосе, штат Калифорния.

Также крупные отделения Cisco Systems находятся в Research Triangle Park, штат Северная Каролина, и Челмсфорде, штат Массачусетс. Кроме того, Cisco принадлежит более 200 отделений по продажам и поддержке в 54 странах. Компания имеет Европейскую и Азиатскую штаб-квартиры, а также штаб-квартиру в Северной и Южной Америке. За счет деловых союзов, заключаемых Cisco Systems с фирмами-партнерами, и приобретения новых компаний наблюдается активная экспансия фирмы на мировом рынке. Из 15 сегментов сетевого компьютерного рынка, на которых выступает компания, Cisco лидирует в 12.

Общая численность сотрудников компании составляет около 12 100 человек. Примерно 7000 из них работают в Бэй-Ариа (Сан-Франциско). Российское представительство компании Cisco Systems насчитывает на настоящий момент более 30 сотрудников. (Источник: http://www.cisco.com/global/RU/win/about/general/corporate_structure.shtml.)

Маршрутизаторы компании Cisco Systems занимают лидирующее положение на рынке (компания поставляет более 80% маршрутизаторов, являющихся основой Интернета), однако, к сожалению, литературу по ним найти достаточно сложно. По указанной причине администраторам сети, впервые сталкивающимся с необходимостью настроить маршрутизатор, приходится либо пользоваться англоязычными источниками, либо прибегать к сторонней помощи. Материал, изложенный в этой главе, поможет специалисту в начальный период работы с маршрутизатором.

Аппаратная архитектура маршрутизаторов Cisco Systems

Маршрутизатор можно рассматривать как специализированный компьютер, который предназначен для выполнения вполне определенных задач. И как всякий компьютер, маршрутизатор имеет собственный центральный процессор (Central Processing Unit — CPU), тип которого может различаться в зависимости от класса маршрутизатора, фирмы-изготовителя, серии маршрутизатора внутри класса (например, это может быть Motorola 68030 или Orion/R4600). Основная задача процессора маршрутизатора (наряду со многими второстепенными) заключается в обработке входящих пакетов для принятия решений об их дальнейшем перенаправлении. При этом скорость, с которой маршрутизатор способен обрабатывать поступающие пакеты, напрямую зависит от типа используемого процессора.

Другой важной частью маршрутизатора, помимо процессора, является его память, которая поделена по функциональному принципу. Маршрутизаторы компании Cisco Systems поддерживают четыре основных типа памяти: постоянное

запоминающее устройство (Read-Only Memory — ROM), флэш-память (flash memory), память с произвольным доступом (Random-Access Memory — RAM) и энергонезависимую память (Non-Volatile RAM — NVRAM). Из перечисленных типов памяти только RAM является энергозависимой, то есть ее содержимое стирается после выключения питания маршрутизатора. Поэтому память RAM может использоваться только для хранения временных данных при работе маршрутизатора.

Память ROM применяется для хранения загрузочного программного обеспечения (bootstrap software), которое запускается первым в момент включения маршрутизатора и в дальнейшем отвечает за его загрузку. Некоторые типы маршрутизаторов хранят всю операционную систему Cisco IOS (Internetwork Operating System) в этой памяти на случай возникновения сбойных ситуаций, когда другие источники, хранящие образ операционной системы, могут стать недоступными.

Основное назначение флэш-памяти заключается в хранении образа (image) операционной системы (собственно операционной системы Cisco IOS), которая и обеспечивает работу маршрутизатора (в том случае, если маршрутизатор имеет такую память). Администратор может хранить в этой памяти образы нескольких операционных систем, чтобы иметь возможность выбрать тип операционной системы при загрузке маршрутизатора. Рассматриваемый тип памяти реализуется либо на процессорной плате маршрутизатора, либо на карте PCMCIA.

Основное назначение памяти NVRAM — хранение конфигурации маршрутизатора, которая считывается при его загрузке.

Кроме памяти и процессора все маршрутизаторы имеют интерфейсы (interfaces, часто в технической литературе встречается термин «порт маршрутизатора»), которые обязательно поименованы и пронумерованы. При этом полное имя интерфейса маршрутизатора содержит его тип (например, Serial) и номер, отсчитываемый с нуля. На тех маршрутизаторах, которые имеют предварительно установленное фиксированное количество интерфейсов (например, на модели Cisco 805), нумерация интерфейсов осуществляется в соответствии с их физическим расположением (порядком следования) на корпусе маршрутизатора. Так, например, ссылка на интерфейс Ethernet0 подразумевает ссылку на первый интерфейс локальной сети. На маршрутизаторах, которые позволяют выполнять смену интерфейсов во время его работы (Online Insertion and Removal — OIR), полное имя интерфейса содержит по крайне мере два числа, разделенных символом /, где первое число определяет номер слота, в который устанавливается интерфейсный модуль, а второе является номером порта. Например, имя интерфейса Ethernet5/0 указывает на первый интерфейс Ethernet в пятом слоте маршрутизатора. Напомним, что отсчет интерфейсов начинается с нуля.

Помимо интерфейсов локальных (Ethernet, Token Ring и т. п.) и глобальных (Serial, ISDN) сетей, все маршрутизаторы компании Cisco Systems имеют консольный порт, предоставляющий асинхронное соединение EIA/TIA-232. Такой порт позволяет с помощью подключения через консольный кабель управлять маршрутизатором с компьютера. Для удобства подключения на стороне маршрутизатора порт оборудован разъемом RJ45. Подобный способ подключения очень удобен в том случае, если возможен физический доступ к маршрутизатору,

а также если необходимо провести начальную настройку нового маршрутизатора или маршрутизатора, у которого все предыдущие настройки были сброшены. В комплекте с новыми маршрутизаторами компания Cisco Systems поставляет консольный кабель. Кабель имеет достаточную длину, для того чтобы при подсоединении к компьютеру можно было удобно расположить маршрутизатор, скажем, на соседнем столе. Точнее, поставляется даже не сам кабель, а так называемый *serial cable console kit* — набор, состоящий из кабеля и соединителей (connector) разного типа. Такой набор позволяет подключаться к консольному порту любого маршрутизатора, произведенного компанией Cisco Systems, и другому оборудованию (например, к коммутаторам). В том случае, если кабеля в комплекте нет (например, когда был куплен бывший в употреблении маршрутизатор без фирменной упаковки), то набор можно просто заказать. На рис. 1.3 представлены составляющие кабельного набора.

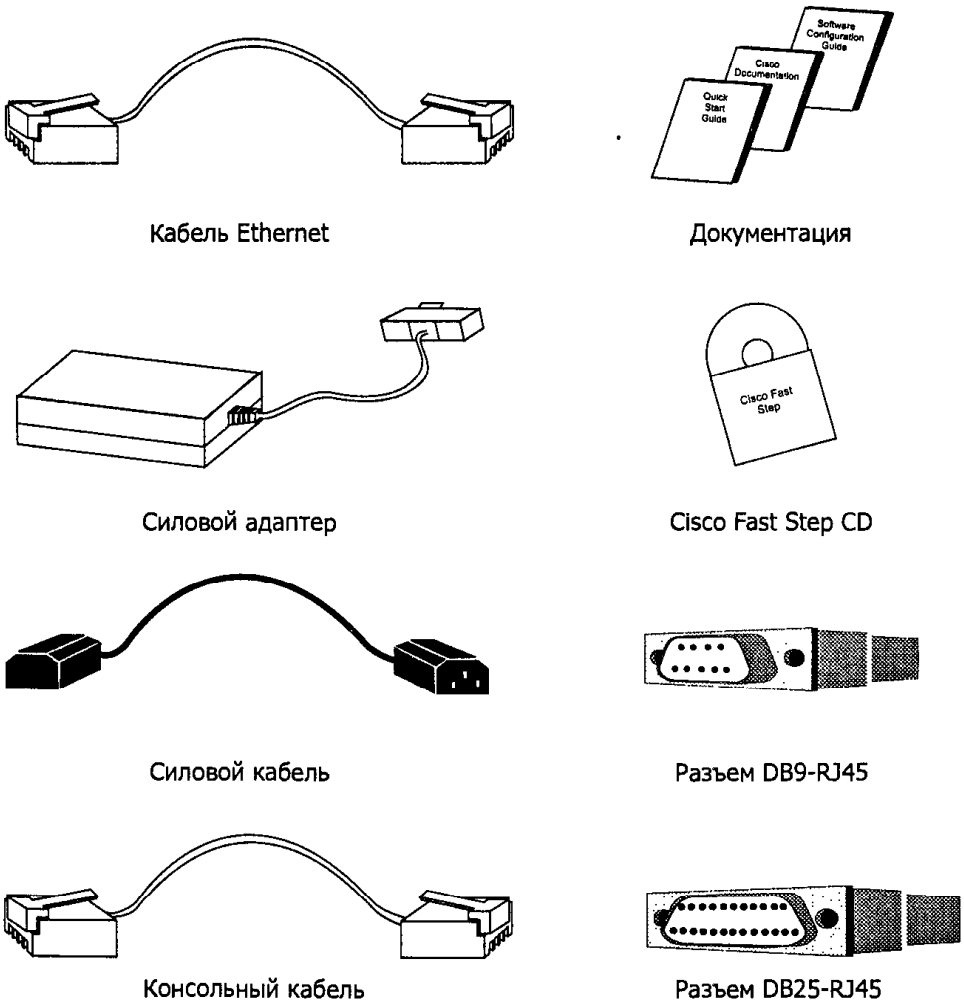


Рис. 1.3. Набор кабелей и разъемов, поставляемый с маршрутизатором

Кроме консольного порта, большинство моделей имеют порт AUX (auxiliary port — вспомогательный порт), который, так же как и консольный, является асин-

хронным соединением EIA/TIA-232 и используется для управления маршрутизатором через обычный модем. Удобство такого управления заключается в том, что в случае возникновения проблем с каналами связи на удаленных объектах всегда существует возможность получения доступа к маршрутизатору для выполнения тех или иных настроек. В качестве другого примера можно привести ситуацию, когда администратор при управлении удаленным маршрутизатором произвел некорректные настройки, результатом которых оказалось то, что связь на сетевом уровне стала недоступной. В этом случае наличие модема, подключенного к порту AUX, поможет исправить последствия допущенных ошибок. Более подробную информацию о портах AUX можно получить в Интернете по адресу http://www.cisco.com/warp/public/701/6.html. Заметим, что новые маршрутизаторы поставляются с инструкцией (на английском языке), в которой расписаны пошаговые действия для подключения модема к AUX-порту.

На рис. 1.4 показана задняя панель маршрутизатора Cisco 805. На этой панели расположены (слева направо) следующие порты: Ethernet, консольный порт и порт Serial. Отметим, что данная модель маршрутизатора не имеет порта AUX.

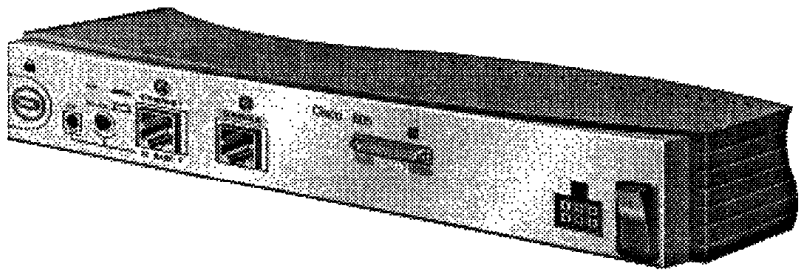


Рис. 1.4. Задняя панель маршрутизатора Cisco 805

Важной составной частью маршрутизатора, помимо его аппаратных компонентов, являются конфигурационные файлы (configuration files). Конфигурационные файлы будут рассмотрены ниже более детально, а пока ограничимся их кратким перечислением. Существует два типа конфигурации операционной системы Cisco IOS: рабочая (running configuration) и загрузочная (startup configuration). Часто конфигурацию первого типа также называют активной (active), так как она располагается в оперативной памяти маршрутизатора (RAM) и определяет его текущие настройки. Когда администратор выполняет команды конфигурирования, на маршрутизаторе изменяется содержимое именно этой конфигурации. В противоположность рабочей (активной), загрузочная конфигурация размещается в памяти NVRAM маршрутизатора и содержит команды операционной системы Cisco IOS, которые выполняются в момент его загрузки.

Рабочая и загрузочная конфигурации в какой-то степени самостоятельны. Обычно администратор сети, выполнив начальные этапы настройки маршрутизатора и проверив его работоспособность, копирует рабочую конфигурацию в память NVRAM, формируя таким образом загрузочную конфигурацию. Очевидно, что основной причиной такой последовательности действий является необходимость сохранения сделанных изменений при перезагрузке маршрутизатора.

ИЗ ОПЫТА

Необходимо отметить одну тонкость, связанную с наличием двух типов конфигурации. При повседневной эксплуатации своей сети автор нередко сталкивался с необходимостью выполнить некоторые сетевые настройки удаленных маршрутизаторов. Бывали случаи, когда последние становились недоступными для управления. А так как далеко не всегда есть возможность самому приехать на удаленный объект, то наиболее действенным способом исправления ситуации является простой звонок по телефону с просьбой к сотруднику этого объекта выключить, а затем снова включить питание маршрутизатора. Учитывая, что все настройки выполняются в рабочей (активной) конфигурации маршрутизатора, а загрузочная при этом не затрагивается (если при этом не копировать принудительно рабочую конфигурацию в загрузочную с помощью команд Cisco IOS), то при повторной загрузке маршрутизатор будет использовать именно загрузочную конфигурацию, в которую некорректные команды не попали.

Введение в Cisco IOS

Компания Cisco Systems, создавшая эту операционную систему, активно работает на рынке сетевого оборудования уже в течение многих лет. Говоря об успехах компании, достаточно сказать, что по имеющимся данным большая часть маршрутизаторов, работающих в Интернете, произведены в компании Cisco, и, по признанию самой компании, своими успехами во многом она обязана именно операционной системе для сетевых устройств IOS. В глоссарии журнала для пользователей Cisco Packet дано следующее определение операционной системы IOS (перевод с английского языка): «Программное обеспечение, работающее в продуктах компании Cisco и представляющее их для пользователей и сетевых администраторов единым целым». IOS предназначена для управления активным сетевым оборудованием и является очень мощной операционной системой. При этом она обеспечивает управление не только через интерфейс командной строки, но и через браузеры Веб.

Достаточно интересно позиционирование компанией Cisco Systems своей операционной системы IOS. На веб-сервере Cisco Systems операционной системе IOS отводится специальный раздел. О распространении IOS в этом разделе говорится как о распространении самостоятельного продукта. При этом своей главной задачей компания видит установление операционной системы IOS в качестве стандарта для других производителей сетевого оборудования. И это несмотря на то, что практическое освоение операционной системы IOS, включающей не одну сотню команд и настроек, дело достаточно трудоемкое. Говорит о сложности этой операционной системы и тот факт, что IOS объединяет все линии продуктов компании, и их совместимость достигается именно благодаря согласованной работе используемых протоколов и технологий.

Большое внимание в системе IOS уделено группе технологий, направленных на защиту данных в сетях. При практической настройке маршрутизаторов часто возникает впечатление, что сам процесс маршрутизации является хоть и необходимым, но далеко не единственным процессом в работе маршрутизатора. Достаточно взглянуть на перечень команд, чтобы понять, насколько трудоемкой может быть настройка механизмов защиты. Для того чтобы настроить процесс маршрутизации, требуется не так уж много команд по сравнению с тем случаем, когда

администратор сети начинает ограничивать доступ с помощью списков (access lists), настраивать технологию трансляции адресов (Network Address Translation — NAT), обеспечивать аутентификацию удаленных пользователей по протоколам RADIUS/TACACS+ и т. д. В книге будут рассмотрены основные моменты практической настройки безопасности, но при первом знакомстве с маршрутизаторами Cisco Systems наиболее важным является понимание базовых команд, без которых невозможна дальнейшая работа.

Начальная настройка маршрутизатора

Для выполнения начальной настройки маршрутизатора абсолютно необходимо предварительно получить полную функциональную схему распределенной сети. Более того, если сеть охватывает больше двух объектов, то без детальной прорисовки сети на бумаге или при помощи специализированных программ настройка маршрутизаторов может оказаться очень обременительным делом. При этом важно не только нарисовать функциональную схему сети, но и разработать адресную схему.

Компания Cisco Systems предлагает очень удобную и функционально богатую программу ConfigMaker, позволяющую графически изобразить планируемую сеть, а затем записать автоматически сгенерированные настройки в память маршрутизаторов. Эта программа доступна без ограничений через Интернет с корпоративного сайта компании Cisco Systems. Если необходимость в использовании программы ConfigMaker отсутствует, то администратор может выполнить пошаговую настройку маршрутизатора средствами операционной системы Cisco IOS.

Для того чтобы приводимое описание команд не было бы чисто теоретическим, автор воспользовался примерами информационных сообщений, полученных с «живых» маршрутизаторов, которые установлены в уже работающей сети одной из петербургских компаний. Сеть сравнительно большая — она связывает более десяти удаленных объектов по выделенным каналам связи и по сети Frame Relay, имеет выход в Интернет и оборудована сервером доступа для обслуживания удаленных пользователей. В сети в качестве аппаратного обеспечения развернуто оборудование компании Cisco Systems — в основном это маршрутизаторы для средних и малых офисов (Cisco 805, Cisco 1005, Cisco 1601 и т. п.). Можно смело сказать, что перечисленные модели маршрутизаторов являются наиболее распространенными, так как наряду с невысокой стоимостью они предлагают широчайший спектр возможностей, часть из которых остается даже невостребованной в повседневной работе. Далее по тексту книги будут демонстрируются фрагменты сети, которые позволят понять работу рассматриваемых технологий. К сожалению, привести всю схему сети не представляется возможным по нескольким причинам, среди которых самой весомой является тот факт, что по мере работы над книгой сеть меняла свои «очертания» — внедрялись новые технологии, менялся подход к обеспечению связи некоторых объектов и т. д.

Первым шагом настройки маршрутизатора является выбор его имени, которое должно быть уникальным.

СОВЕТ

По мнению автора, наиболее удобным является указание в имени маршрутизатора данных о его расположении и модели, что нередко помогает администратору сети, так как позволяет не запоминать используемые пароли доступа к маршрутизатору — из его модели и расположения можно легко определить пароль, если, конечно, администратор выбрал такую схему назначения паролей.

Вообще выбранное имя может быть абсолютно произвольным, однако нужно следовать рекомендациям, приведенным в документе RFC 1035. Имя маршрутизатора не должно занимать больше 63 знакомест. Его следует начинать с символа, а заканчиваться оно может либо цифрой, либо символом. При формировании имени следует учесть, что оно является частью приглашения на ввод команд маршрутизатора, а в приглашении будет использовано не более 29 символов имени.

В работе любой распределенной сети самое активное участие принимает протокол сетевого уровня, причем, как правило, этот протокол является маршрутизируемым. Естественно, на маршрутизаторах необходимо проводить его настройку. Поддержка же того или иного протокола зависит от возможностей операционной системы. Например, если используется операционная система Cisco IP Feature Set, то, как видно из ее названия, маршрутизатор будет поддерживать только протокол IP. Если же используется Cisco Enterprise Feature Set, то будет доступен любой маршрутизируемый протокол.

СОВЕТ

Можно вполне определенно сказать, что поддержки только одного протокола IP вполне достаточно для большинства сетей, администраторы которых стремятся к унификации программной базы. Другой достаточно весомой причиной такого выбора является тот факт, что с новыми маршрутизаторами поставляется именно операционная система IP Feature Set.

В процессе выполнения начальной настройки маршрутизатора следует задать пароли, при помощи которых в дальнейшем будет организован доступ. Все пароли доступа чувствительны к регистру (case sensitive) и могут содержать любую комбинацию прописных или строчных букв, цифр и пробелов (последние не должны быть первыми в пароле). Максимальная длина пароля ограничена 25 символами, хотя на практике такие длинные пароли редко находят применение.

Для того чтобы настроить новый маршрутизатор, необходимо подключить к нему с помощью консольного порта компьютер и настроить программу эмуляции терминала (terminal emulation software). Подключение выполняется достаточно просто. Любой компьютер имеет по крайне мере два последовательных порта: COM1 и COM 2. Обычно физические разъемы этих портов имеют марку либо DB9M (DB8 Male), либо DM25M (DB25 Male). На небольших по размеру маршрутизаторах (например, Cisco 805) для организации консольного порта используется разъем RJ45. Кроме того, для этих моделей поставляется готовый набор для подключения к компьютеру. В набор входит кабель, который имеет на обоих концах штекеры RJ45, и два специальных переходника, имеющих с одной стороны разъем RJ45, а с другой — разъем DB9F (DB9 Female) или DB25F (DB25 Female). Для подключения маршрутизатора выбирается свободный COM-порт на компьютере, а затем к нему подсоединяется кабель с необходимым типом

разъема. Например, если на компьютере свободен порт с разъемом DB9M, то нужно воспользоваться схемой RJ45-DB9F (рис. 1.5).

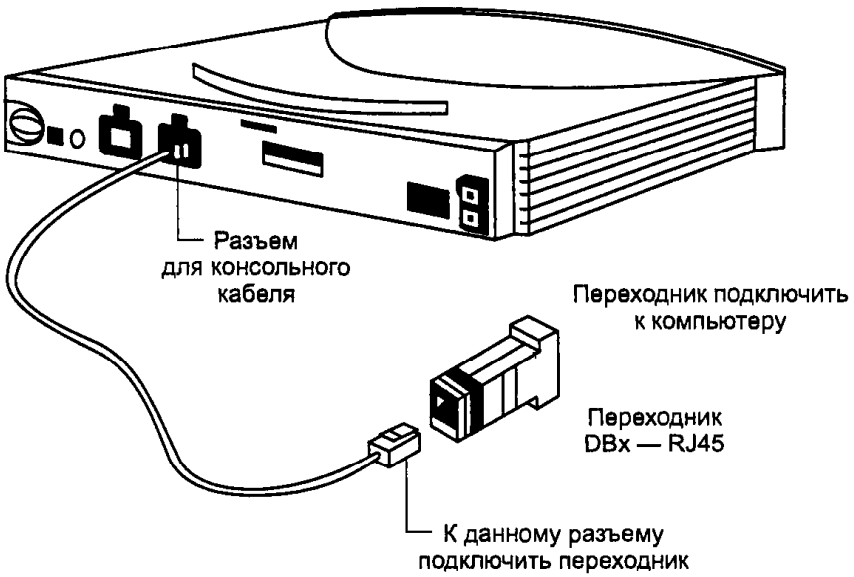


Рис. 1.5. Схема подключения маршрутизатора Cisco 805 к компьютеру

После физического подключения маршрутизатора к компьютеру для проведения процедуры настройки запускается программа эмуляции терминала. Существует множество готовых программных пакетов эмуляции терминала, и часто они поставляются прямо с операционной системой. Например, если на компьютере установлена операционная система Microsoft Windows NT 4.0 или Windows 2000, то можно воспользоваться программой HyperTerminal. Далее необходимо настроить параметры соединения, которые по умолчанию описываются популярной схемой 9600-8N1 (9600 baud, 8 data bits, no parity, 1 stop bits).

Теперь, после подключения маршрутизатора к компьютеру через консольный порт и настройки программы эмуляции терминала можно включать сам маршрутизатор. При этом на маршрутизаторе начинает выполняться загрузочное программное обеспечение (bootstrap software), которое запускает тест самодиагностики (Power-On Self-Test — POST), а затем находит загрузочное устройство (обычно это флэш-память), в котором содержится корректный образ операционной системы Cisco IOS. Если включение маршрутизатора выполнялось при запущенной программе эмуляции терминала, то сообщения о начальной загрузке будут передаваться на консольный порт. Ниже приведен пример выводимых сообщений при загрузке маршрутизатора Cisco 805.

```
TinyROM version 1.2(2)
16:36 08/02/99
Copyright (c) 1998-1999 by cisco Systems, Inc.
All rights reserved.

POST ..... OK.  8MB DRAM, 4MB Flash.
Booting "c805-y6-mw.120-4.XM" ...,
```

```
...
Cisco Internetwork Operating System Software
IOS (tm) C805 Software (C805-Y6-MW), Version 12.0(4)XM, EARLY DEPLOYMENT
   RELEASE SOFTWARE (fc1)
TAC:Home:SW:IOS:Specials for info
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 17-Jun-99 16:51 by linda
Image text-base: 0x0013B000, data-base: 0x00532000

Cisco C805 (MPC850) processor (revision 0) with 46944K bytes of virtual memory
Processor board ID JAD035005G4
CPU part number 33
Bridging software.
1 Ethernet/IEEE 802.3 interface(s)
1 Serial(sync/async) network interface(s)
8M bytes of physical memory (DRAM)
8K bytes of non-volatile configuration memory
4M bytes of flash on board
...
```

Как видно из приведенного примера, был обнаружен образ операционной системы, после чего началась загрузка (Booting "c805-y6-mw.120-4.XM" ...,). Далее из выдаваемой на экран информации можно узнать доступный объем памяти на маршрутизаторе, версию загрузочного программного обеспечения и версию самой операционной системы Cisco IOS (Version 12.0(4)XM).

Пошаговая настройка маршрутизатора

В процессе загрузки операционной системы Cisco IOS на экране отображается достаточно много информации, из которой можно узнать, например, объем оперативной памяти, объем памяти NVRAM и объем флэш-памяти (эту информацию также можно извлечь с помощью команды show version). В это время операционная система производит проверку памяти NVRAM на наличие в ней информации о конфигурации. На новых маршрутизаторах такой информации о конфигурации нет, и операционная система, обнаружив это, запускает программу пошаговой настройки маршрутизатора. В англоязычной технической литературе для обозначения такой программы используется термин System Configuration Dialog. Кратко рассмотрим процесс настройки маршрутизатора в обозначенной ситуации. Нужно отметить простоту использования программы и возможность получения подсказки от нее по каждому возникающему вопросу.

```
...
          --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: yes

At any point you may enter a question mark '?' for help.
```

```
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.
...
```

Работа программы пошаговой настройки маршрутизатора всегда начинается с вывода строк, в которых изложены основные приемы ее использования, после чего запрашиваются различные параметры. Чтобы воспользоваться подсказкой, нужно ввести символ ?. Если администратор сети начал выполнять пошаговую настройку маршрутизатора, а затем решил, что ему нет необходимости продолжать, он вправе нажать комбинацию клавиш Ctrl+C для остановки диалога. Если в процессе выполнения пошаговой настройки были даны ошибочные ответы на вопросы, то возможностей прямого исправления ошибок нет. Для коррекции ошибки есть два варианта действий: либо выключить, а затем вновь включить маршрутизатор и при его загрузке начать пошаговую настройку с выдачей правильных ответов, либо закончить пошаговую настройку, а затем исправить сделанные ошибки вручную, посредством ввода команд операционной системы в командной строке. Программа пошаговой настройки предлагает варианты ответов по умолчанию (которые показываются в квадратных скобках) на большинство задаваемых вопросов. Если ответ по умолчанию подходит для конкретной конфигурации, можно просто нажать клавишу Enter. В процессе пошаговой настройки предлагается изменить имя маршрутизатора. По умолчанию маршрутизатору присваивается имя Router.

```
...
Configuring global parameters:
  Enter host name [Router]: Cisco805_Wharehouse…
```

При вводе имени маршрутизатора различаются строчные и прописные буквы (case sensitive). Напомним, что введенное имя маршрутизатора будет присутствовать в приглашениях на ввод команд в разных режимах. Это позволяет быстро идентифицировать конкретный маршрутизатор, с которым ведется работа. Если распределенная сеть построена на нескольких маршрутизаторах, то имя, предлагаемое по умолчанию, рекомендуется изменять, чтобы не путать маршрутизаторы друг с другом.

)ВЕТ

Как уже отмечалось, можно порекомендовать выбрать схему паролей таким образом, чтобы они были связаны с именем маршрутизатора. Удобство такой схемы в том, что не нужно заучивать наизусть пароли для десятка маршрутизаторов. Достаточно подключиться к маршрутизатору, а затем из его имени определить пароль. Естественный недостаток такой схемы связан с тем, что злоумышленник, поняв таким образом логику формирования паролей, сумеет получить доступ ко всем маршрутизаторам.

После ввода имени будет предложено ввести пароли, которые желательно обдумать еще на этапе планирования настроек маршрутизатора.

```
...
The enable secret is a password used to protect access to
   privileged EXEC and configuration modes. This password, after
   entered, becomes encrypted in the configuration.
   Enter enable secret: ******
```

```
    The enable password is used when you do not specify an
    enable secret password, with some older software versions, and
    some boot images.
    Enter enable password: ******

% Please choose a password that is different from the enable secret
    Enter enable password: ******

    The virtual terminal password is used to protect
    access to the router over a network interface.
    Enter virtual terminal password: ******
...
```

На каждый задаваемый по паролям вопрос необходимо дать ответ, так как для новых маршрутизаторов на эту группу вопросов нет ответов по умолчанию. Для предотвращения случайного просмотра запрашиваемый пароль enable secret password хранится в зашифрованном виде в конфигурации операционной системы Cisco IOS, а пароль enable password показывается в открытую. Оба этих пароля предназначены для выполнения одной задачи — контроля доступа в привилегированном режиме маршрутизатора, особенности которого будут рассмотрены ниже. Следует отметить, что пароль enable secret password имеет преимущество перед паролем enable password, но система предлагает вводить их оба, так как старые версии операционной системы Cisco IOS не поддерживают зашифрованный пароль. Пароль доступа к виртуальному терминальному порту VTY (virtual terminal), как можно догадаться из названия, используется при установлении сеанса Telnet с маршрутизатором после выполнения его начальной настройки.

Далее программа пошаговой настройки дает возможность задать параметры интерфейсов. После этого она представит на обозрение получившуюся конфигурацию маршрутизатора и предложит несколько вариантов дальнейших действий. Это может быть, например, запись конфигурации в память NVRAM, как показано в примере ниже.

```
...
[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

Enter your selection [2]: 0 You can enter the setup, by typing setup at IOS
    command prompt
...
```

Интерфейс командной строки Cisco IOS

После того как администратор выполнил программу пошаговой настройки, может появиться необходимость внести изменения в конфигурацию. Большинство конфигурационных команд выполняются с помощью интерфейса командной строки (Command Line Interface — CLI), который предоставляется операционной

системой Cisco IOS. Поэтому для настройки маршрутизаторов необходимо иметь хорошее представление о том, как им пользоваться. И хотя после приобретения некоторого опыта работа с интерфейсом не будет представлять никаких трудностей, начинающему администратору он вряд ли покажется простым. В этом разделе рассматриваются основные возможности интерфейса командной строки и некоторые задачи настройки маршрутизатора, которые не зависят от сетевых протоколов, используемых в распределенной сети.

Очевидно, что для работы с маршрутизатором к нему нужно предварительно подключиться. Подключение может быть прямым с помощью консольного порта либо через сеть по протоколу Telnet. При начальной настройке маршрутизатора наиболее распространен первый способ подключения (если, конечно, маршрутизатор физически расположен недалеко от компьютера).

После подключения в случае, если на маршрутизаторе не был настроен пароль доступа по консольному порту, можно просто нажать клавишу Enter и, получив доступ, приступить к вводу команд.

```
Press RETURN to get started!
<Enter>
Router>
```

Защита доступа к консольному порту маршрутизатора с помощью пароля является хорошей практикой, но при этом последовательность подключения несколько изменяется — маршрутизатор выдает приглашение к вводу пароля.

```
Press RETURN to get started!
<Enter>
User Access Verification
Password: ******
Router>
```

СОВЕТ

При наличии доступа к консольному порту маршрутизатора становится возможным получить доступ к маршрутизатору, даже не зная пароля. Это можно сделать, запустив процедуру восстановления утерянного пароля. Поэтому рекомендуется помещать маршрутизатор в физически защищенное место для снижения вероятности неконтролируемого доступа.

ИЗ ОПЫТА

Очень удобным способом работы с маршрутизатором через консольный порт может быть подключение последнего к какому-либо серверу и запуск на сервере программы Terminal (в случае операционной системы Windows 2000). Здесь имеется ввиду не программа эмуляции виртуального терминала, а компонент, позволяющий пользователям создавать виртуальные сеансы, как будто они работают за локальным компьютером. При установке операционной системы Windows 2000 администратор может установить компонент Terminal Server, который по умолчанию позволяет работать с виртуальными сеансами только пользователям, включенным в состав административных групп.

Подключив сервер к консольному порту маршрутизатора, администратор получает возможность управления маршрутизатором, находясь при этом сколь угодно далеко от него (естественно, при наличии сетевой связи с сервером). Но несмотря на удобство такого подхода, у него есть и недостаток. Если злоумышленник смог получить доступ к виртуальному сеансу на сервере (что само по себе уже крупная неудача для специалиста, отвечающего за безопасность сети), то он сумеет получить доступ и к консольному порту маршрутизатора.

Доступ предоставляется только в случае введения корректного пароля, при этом даются три попытки. После трех неудачных попыток процедура входа начинается заново. Следует отметить, что при работе с маршрутизатором через консольный порт такая ситуация не создает серьезных проблем, так как, выждав несколько секунд перед вторичным нажатием клавиши Enter, можно начать вводить пароль повторно. Маршрутизатор ждет ввода пароля в течение 30 секунд, и если за это время ничего не ввести, то процедуру регистрации нужно будет проходить еще раз.

Администратор может также подключиться к маршрутизатору с помощью сеанса Telnet и после этого ввести корректный пароль. После того как было выдано приглашение к вводу пароля, дальнейшее поведение маршрутизатора полностью аналогично случаю использования консольного порта. Пароль, который предлагается напечатать при использовании сеанса Telnet, указывается при настройке виртуального терминального порта VTY.

```
User Access Verification

Password:
% Password:  timeout expired!
Password: ******
Router>
```

Следует отметить, что хотя в рассмотренных примерах вместо пароля показаны символы *, на практике при его вводе на экране ничего не отображается. Эти символы приведены для наглядности. Так как все примеры были взяты с реально работающего оборудования, то по понятным причинам пароли не указаны.

Приглашение, выдаваемое операционной системой Cisco IOS (в приведенном выше примере — Router>), изменяется в зависимости от режима настройки. Приведем перечень базовых режимов, в которых выполняется большинство конфигурационных команд.

1. Пользовательский режим (User Mode).
2. Привилегированный режим (Privileged Mode).
3. Глобальный режим (Global Configuration Mode).
4. Режим подконфигурации (Sub-Configuration Modes).
5. Режим ROM Monitoring (ROM Monitoring Mode).

Рассмотрим каждый из этих режимов и их приглашения к вводу команд, которые и помогают идентифицировать текущий режим. Когда происходит подключение к маршрутизатору, запускается пользовательский режим. При этом в строке приглашения выводится настроенное имя маршрутизатора и символ >. В предыдущем примере при подключении к маршрутизатору посредством сеанса Telnet выдаваемое приглашение показывает, что после авторизации (проверки пароля) работа ведется в пользовательском режиме.

```
Router>
```

Возможности, предоставляемые пользовательским режимом, можно охарактеризовать одной фразой: «смотри, но ничего не меняй». Это объясняется тем,

что в данном режиме не предоставляется никаких путей серьезно воздействовать на работу операционной системы Cisco IOS. При этом можно просматривать любую информацию за исключением информации по рабочей (running) и по загрузочной (startup) конфигурации.

Если необходимо выполнить команды, которые потенциально будут влиять на работу маршрутизатора, или просмотреть текущую конфигурацию, то нужно перейти из пользовательского режима в привилегированный, в котором доступны все настройки, предусмотренные в данной операционной системе Cisco IOS. Так, например, можно просмотреть конфигурацию маршрутизатора или перезагрузить его. Для того чтобы перейти в привилегированный режим, следует ввести команду enable, результат которой будет следующий (для подключения через сеанс Telnet):

```
User Access Verification

Password:******
Router>enable
Password:******
Router#
```

Как видно из примера, при вводе команды enable операционная система предлагает ввести пароль для перехода в привилегированный режим работы. При наборе корректного пароля меняется выдаваемое приглашение, что указывает на работу в новом режиме — за именем маршрутизатора следует символ #. Начиная с этого момента администратор получает полный контроль над маршрутизатором. Довольно часто в англоязычной технической литературе привилегированный режим называют режимом разрешения (enable mode), очевидно из-за аналогии с командой enable и в противовес режиму запрета (команда disable).

Цель	Команда
Перейти в привилегированный режим работы	enable <пароль>
Получить список доступных команд	Ввод символа ?
Вернуться в пользовательский режим	disable

При начальной настройке маршрутизатора предлагается задать пароль доступа в привилегированный режим. Пароль доступа обозначается на экране словосочетанием enable secret, где слово secret указывает на то, что при просмотре файлов конфигурации пароль отображается на экране в зашифрованном виде.

Кроме этого пароля предлагается задать так называемый enable password, который при наличии первого зашифрованного пароля не несет на маршрутизаторе никакой нагрузки. Однако в том случае, если из конфигурации маршрутизатора изъять зашифрованный пароль, то доступ к привилегированному режиму будет контролироваться именно паролем enable password.

В рассмотренных выше примерах для возврата в пользовательский режим из привилегированного нужно выполнить команду disable.

```
Router#disable
Router>
```

Конфигурационные режимы предназначены для ввода команд, управляющих работой маршрутизатора. Основным режимом настройки является глобальный режим (global configuration mode), из которого можно попасть в другие режимы. Все команды, которые вводятся в этом режиме, влияют на рабочую конфигурацию маршрутизатора (running configuration) и вступают в силу немедленно после их ввода. В глобальный режим можно попасть только из привилегированного. При этом команды, доступные в пользовательском и привилегированном режимах, в глобальном режиме маршрутизатором не воспринимаются.

В глобальном режиме все изменения сказываются на общей, или «глобальной», работе маршрутизатора. Например, в этом режиме допускается смена имени маршрутизатора, установка некоторых паролей и т. п. Для того чтобы перейти из привилегированного режима в глобальный, нужно выполнить команду configure terminal.

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

Цель	Команда
Перейти в глобальный режим работы	configure terminal
Получить список доступных команд	?
Выйти из глобального режима работы	end, exit или Ctrl+Z

Как видно из примера, приглашение к вводу команд в глобальном режиме состоит из имени маршрутизатора, ключевого слова config, заключенного в скобки, и символа #. В соответствии с рекомендацией, показанной в примере, данный режим можно немедленно покинуть и вернуться в привилегированный режим, если нажать комбинацию клавиш Ctrl+Z. Помимо этого, возврат в привилегированный режим осуществляется также командами end и exit.

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#^Z
Router#
```

Находясь в глобальном режиме, администратор может перейти и в другие режимы конфигурации. Режимы подконфигурации (sub-configuration modes) используются для настройки индивидуальных компонентов, таких, например, как интерфейсы маршрутизатора. При этом сама команда, которая необходима для переключения в тот или иной режим подконфигурации, зависит от настраиваемого компонента. Например, для выполнения настроек интерфейса следует перейти в режим конфигурации интерфейсов (interface configuration mode). Для такого перехода нужно выполнить команду interface, находясь в глобальном режиме. Эта команда требует в качестве параметра полного имени интерфейса и его номера. Задав в качестве параметра знак вопроса, вы увидите список интерфейсов, доступных для настройки.

```
Router(config)#interface ?
  Async             Async interface
  BVI               Bridge-Group Virtual Interface
  Dialer            Dialer interface
  Ethernet          IEEE 802.3
  Group-Async       Async Group interface
  Lex               Lex interface
  Loopback          Loopback interface
  Null              Null interface
  Serial            Serial
  Tunnel            Tunnel interface
  Virtual-Template Virtual Template interface
```

Цель	Команда
Находясь в глобальном режиме, перейти в режим конфигурации интерфейсов	interface <тип интерфейса> <номер интерфейса>
Получить список доступных команд	?

Например, для того чтобы оказаться в режиме конфигурации интерфейсов и настроить интерфейс Serial0, нужно выполнить следующую команду:

```
Router(config)#interface serial0
Router(config-if)#
```

Все режимы подконфигурации имеют приглашение, состоящее из имени маршрутизатора, ключевого слова config и обозначения, указывающего на текущий режим, заключенной в скобки. В приведенном примере обозначение –if свидетельствует о том, что работа идет в режиме конфигурации интерфейсов. В строке этого приглашения можно вводить команды, влияющие на настройку интерфейса Serial0, а также любые команды, относящиеся к глобальному режиму, так как последние принимаются маршрутизатором в любом из режимов конфигурации. Для того чтобы из режима конфигурации интерфейсов вернуться в глобальный, надо ввести команду exit.

В том случае, если администратор, находясь в режиме подконфигурации, введет команду, которая не регламентирует переход в другой режим, система вернется в глобальный режим. Например, в режиме конфигурации интерфейсов можно ввести команду, изменяющую имя маршрутизатора:

```
Router(config)#interface serial0
Router(config-if)#hostname Router
Router(config)#
```

Существуют около двадцати режимов подконфигурации. Наиболее активно используемые будут рассмотрены далее в главах книги, посвященных вопросам безопасности (глава 9), протоколам маршрутизации (глава 7) и технологии Frame Relay (глава 8).

Режим ROM Monitoring (ROMMON) на самом деле не является режимом работы операционной системы Cisco IOS — в этом состоянии маршрутизатор работает, пока операционная система еще не загружена. Если маршрутизатор пы-

тается загрузиться и не находит соответствующего файла загрузки (IOS image), то он переходит в режим ROMMON. Администратор вправе принудить маршрутизатор переключиться в этот режим путем посылки специального сигнала с клавиатуры (обычно следует нажать комбинацию клавиш Ctrl+Break) на консольный порт в течение 60 секунд начиная с момента загрузки. В результате маршрутизатор прервет процесс загрузки и перейдет в режим ROM Monitoring. Как правило, данное переключение выполняется для восстановления утерянного пароля. В этом режиме присутствуют команды, которые позволяют вручную загрузить маршрутизатор с указанием корректного файла загрузки.

Цель	Команда
Перейти в режим ROMMON, находясь в привилегированном режиме (только в нем доступна команда reload, выполняющая принудительную перезагрузку маршрутизатора)	Reload или нажать комбинацию Ctrl+Break в течение 60 секунд в процессе загрузки маршрутизатора
Получить список доступных команд	?

В режиме ROMMON не так много доступных для выполнения команд. Их список можно получить вводом символа ?. Таблица 1.1 содержит перечень команд для навигации по возможным режимам работы маршрутизатора.

Таблица 1.1. Режимы работы маршрутизатора

Режим работы	Метод перехода	Приглашение маршрутизатора	Как покинуть режим
Пользовательский	При вводе пароля на подключение	Router>	Ввод команды logout
Привилегированный	Находясь в пользовательском режиме, ввести команду enable	Router#	Для возврата в пользовательский режим следует ввести команду disable
Глобальный	Находясь в привилегированном режиме, ввести команду configure terminal	Router(config)#	Для выхода в привилегированный режим следует ввести команду exit
Конфигурирование интерфейсов	Находясь в глобальном режиме, ввести команду interface с параметрами в виде имени и номера выбранного интерфейса. interface Serial1	Router(config-if)#	Для выхода в глобальный режим следует ввести команду exit
Конфигурирование подинтерфейсов	Находясь в глобальном режиме, ввести команду interface с параметрами в виде имени и номера выбранного подинтерфейса interface Serial1.1	Router(config-subif)#	Для выхода в глобальный режим следует ввести команду exit

Команды Cisco IOS

После рассмотрения различных режимов, в которых можно настраивать маршрутизатор, разберем формат команд настройки и метод их ввода. Вообще говоря, использование операционной системы Cisco IOS не должно вызывать особых трудностей — требуется только ввести команду и нажать клавишу Enter, которая инициирует исполнение команды. В том случае, если введенная команда некорректна (например, ошибка в синтаксисе), операционная система укажет на допущенную ошибку. Если команда вводится в пользовательском или привилегированном режимах, операционная система проверяет весь список команд, для того чтобы убедиться в правильности введенной команды. В случае, если такая команда не найдена в списке, операционная система рассматривает ее как имя удаленного устройства, с которым нужно установить сеанс Telnet. Например, если, находясь в пользовательском режиме, ввести слово customer, которое не является командой операционной системы, то можно получить следующий результат:

```
Router#customer
Translating "customer"...domain server (192.168.4.253)
% Unknown command or computer name, or unable to find computer address
Router#
```

Из примера видно, что маршрутизатор пытается преобразовать введенное слово в IP-адрес путем опроса настроенного на нем адреса (192.168.4.253) сервера DNS. Поэтому лучше просто ввести IP-адрес и, если это будет возможно, маршрутизатор установит с указанным устройством сеанс Telnet.

```
Router#192.168.2.254
Trying 192.168.2.254 ... Open

User Access Verification
Password:
```

С помощью команды terminal администратор может контролировать способ выдачи информации и изменять другие настройки, доступные в пользовательском режиме. Любые переопределяемые этой командой параметры действительны только в течение текущего сеанса и не восстанавливаются при следующем подключении к маршрутизатору.

Выше были рассмотрены простые команды, которые практически не требовали указания параметров. Однако для углубленной настройки маршрутизатора могут понадобиться и более сложные команды. В связи с этим полезной будет информация о наиболее эффективных способах работы с командной строкой. Кроме того, без них интенсивная настройка с использованием множества команд может показаться обременительной.

Операционная система позволяет редактировать команды во время их ввода, то есть перемещать курсор по командной строке, добавлять или удалять символы. Каждая вводимая команда ограничена по длине 253 символами, а операционная система для удобства запоминает десять последних введенных команд в буфере истории. Самый удобный способ перемещения курсора между символами вводимой команды — это нажатие клавиш с изображениями стрелок. Эти клавиши

можно использовать в ANSI-совместимых терминальных программах, таких как VT100, VT220 и VT320.

Отображение накопленных в буфере истории команд производится клавишами с изображением стрелок, направленных вверх и вниз. Итак, в буфере истории сохраняется десять последних введенных команд, а в распоряжении администратора есть две команды настройки — одна меняет размер буфера, а другая запрещает его использование. Для последнего действия нужно выполнить команду terminal no history, а для изменения количества команд, сохраняемых в буфере, предназначена команда terminal history size с параметром, указывающим на количество команд:

```
Router>terminal history size 15
```

Все вводимые команды и их параметры допускается сокращать до акронима, который однозначно идентифицирует команду. Такое упрощение позволяет значительно сократить время набора и снизить число синтаксических ошибок. В табл. 1.2 перечислены некоторые возможные сокращения.

Таблица 1.2. Возможные сокращения команд

Полное имя команды	Сокращение
Show	sh
Show protocols	sh prot
Terminal length 0	term len 0
Enable	en
Configure terminal	conf t
Ethernet	e
Serial	s
Interface ethernet0	int e0
Quit	q

Следует отметить, что операционная система Cisco IOS обладает очень полезным качеством: она предоставляет обширную справочную информацию. Для того чтобы вызвать необходимую справку, следует вместо команды ввести знак вопроса ? и нажать клавишу Enter. В ответ на это маршрутизатор выведет список команд, которые доступны для выполнения в текущем режиме настройки. В том случае, если знак вопроса записывается в конце набираемой команды, нет необходимости нажимать клавишу Enter, так как операционная система воспримет знак вопроса как конец команды и сразу же выдаст подсказку. Например, можно получить подсказку по командам, допустимым в пользовательском режиме. Отметим, что в данном примере объем выводимой информации искусственно ограничен, в то время как на практике список команд способен простираться на несколько экранов.

```
Router>?
Exec commands:
   <1-99>          Session number to resume
   access-enable   Create a temporary Access-List entry
```

```
  clear          Reset functions
  connect        Open a terminal connection
  disable        Turn off privileged commands
  disconnect     Disconnect an existing network connection
  enable         Turn on privileged commands
  exit           Exit from the EXEC
  help           Description of the interactive help system
  lock           Lock the terminal
  login          Log in as a particular user
  logout         Exit from the EXEC
...
```

В следующем примере показана подсказка по командам, доступным в привилегированном режиме работы операционной системы.

```
Router>enable
Router#?
Exec commands:
  <1-99>          Session number to resume
  access-enable   Create a temporary Access-List entry
  access-template Create a temporary Access-List entry
  bfe             For manual emergency modes setting
  clear           Reset functions
  clock           Manage the system clock
  configure       Enter configuration mode
  connect         Open a terminal connection
  copy            Copy configuration or image data
  debug           Debugging functions (see also 'undebug')
  disable         Turn off privileged commands
  disconnect      Disconnect an existing network connection
  enable          Turn on privileged commands
...
```

Если ввести акроним команды, а за ней без пробела сразу же знак вопроса, то операционная система напечатает список всех доступных команд, которые соответствуют этому сокращению:

```
Router#co?
configure  connect  copy
```

Помимо общего списка доступных команд можно получить рекомендацию по применению конкретной команды, выполняя ее со знаком вопроса в качестве параметра. Следующие примеры иллюстрируют работу справочной системы, помогая усвоить действие одной из команд маршрутизатора — clock, предназначенной для установки даты и времени на маршрутизаторе.

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#clock ?
  summer-time  Configure summer (daylight savings) time
  timezone     Configure time zone
```

```
Router(config)#clock timezone ?
   WORD  name of time zone

Router(config)#clock timezone ZONE ?
   <-23 - 23>  Hours offset from UTC
```

По мере выполнения все большего числа настроек на маршрутизаторе становится поистине бесценной возможность повторения введенных ранее команд с помощью нажатия клавиши ↑. Другой очень полезной функциональностью является использование клавиши табуляции Tab, которая заканчивает вводимую команду после указания нескольких ее первых символов. Другими словами, после того как были набраны несколько первых уникальных символов и нажата клавиша Tab, операционная система Cisco IOS добавит окончание команды самостоятельно. Например, часто приходится вводить команду configure terminal, которая является достаточно длинной, для того чтобы вызвать некоторый дискомфорт для администратора сети при ее частом наборе. Для ускорения набора можно ввести conf t и нажать клавишу Tab.

Следует отметить, что приведенная в примере команда conf t не требует обязательного нажатия клавиши Tab, так как ее первая составляющая (conf) однозначно идентифицирует команду configure, а вторая составная часть (t) так же однозначно идентифицирует ключевое слово terminal. Используя табуляцию и знак вопроса, можно получить минимально необходимое количество информации для выполнения настройки. Например, с комбинации символов co начинаются три команды, поэтому ввод этих двух символов неоднозначно определяет команду configure. Символов con также недостаточно для ввода команды configure, поскольку существует еще команда connect.

```
Router#con?
configure  connect

Router#conf<Tab>
Router#configure t<Tab>
Router#configure terminal<Tab>
```

По мере выполнения настройки маршрутизатора администратор может отменить введенную ранее команду. Для этого следует ввести ключевое слово no в начале конфигурационной команды. Практически все команды операционной системы Cisco IOS могут быть отменены таким способом. В следующем примере показана настройка параметра, отвечающего за скорость, доступную на интерфейсе Serial0, а затем отмена этого действия.

```
Router(config)#interface serial0
Router(config-if)#bandwidth 256
Router(config-if)#no bandwidth 256
```

Наиболее популярная (с точки зрения автора) команда, которая выполняется в пользовательском и привилегированном режимах, — это команда show. Она используется в том случае, когда необходимо получить информацию о работе и настройках операционной системы. Иногда выдаваемые сведения не помещаются

на один экран, и в этом случае операционная система предлагает вручную управлять выдачей дальнейшей информации. Для этого можно воспользоваться клавишей Enter, после нажатия которой выдается следующая строка текста, или клавишей пробела (Space), которая позволяет вывести еще один экран. При этом объем информации, выдаваемой при нажатии клавиши пробела, зависит от того, сколько строк информации требуется отобразить и на выдачу какого количества строк настроена операционная система. Значение по умолчанию составляет 24 строки — число, приемлемое для большинства случаев. Это значение можно поменять посредством команды terminal length, задав в качестве аргумента желаемое число строк. Обычно необходимости в изменении количества отображаемых строк не возникает, но такая возможность потенциально полезна, например, если при использовании терминальной программы необходимо сохранить результат, выдаваемый в ответ на команду show, в файле. При этом можно запретить выдачу информации поэкранно с помощью команды terminal length 0. После выполнения данной команды операционная система будет отображать информацию самостоятельно без нажатия клавиш, инициирующих продолжение ее отображения. Здесь следует отметить, что по умолчанию все терминальные сеансы работы с маршрутизатором будут автоматически закрываться по истечении 10 минут состояния неактивности. При необходимости это значение можно изменить с помощью команды exec-timeout, которая будет рассмотрена в разделе «Работа с интерфейсами и портами маршрутизатора» главы 3.

Информация о работе маршрутизатора и его возможностях является полезным подспорьем для администратора сети. Например, довольно часто нужно быть в курсе следующих вопросов: насколько загружен процессор маршрутизатора, какой объем памяти доступен и сколько ее задействовано, какая версия операционной системы Cisco IOS задействована на маршрутизаторе, какие протоколы настроены и т. п. Такая информация может быть получена с помощью команды show. Эта команда выполняется либо в пользовательском, либо в привилегированном режиме, причем в пользовательском режиме доступна любая информация, за исключением данных о конфигурации маршрутизатора.

Применяемая в пользовательском и привилегированном режимах команда show version возвращает сведения общего характера о маршрутизаторе. Следующий пример показывает результат работы этой команды для модели маршрутизатора Cisco 1601.

```
Cisco1601>show version
IOS (tm) 1600 Software (C1600-Y-L), Version 11.3(3)T, RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1998 by cisco Systems, Inc.
Compiled Mon 20-Apr-98 19:48 by ccai
Image text-base: 0x08022C58, data-base: 0x02005000

ROM: System Bootstrap, Version 11.1(10)AA, EARLY DEPLOYMENT RELEASE SOFTWARE
   (fc1)
ROM: 1600 Software (C1600-BOOT-R), Version 11.1(10)AA, EARLY DEPLOYMENT
   RELEASE
SOFTWARE (fc1)

Cisco1601 uptime is 6 days, 22 hours, 54 minutes
```

```
System restarted by power-on
System image file is "flash:c1600-y-l.113-3.T.bin", booted via flash

cisco 1601 (68360) processor (revision C) with 1536K/512K bytes of memory.
Processor board ID 10510267, with hardware revision 00000000
Bridging software.
X.25 software, Version 3.0.0.
1 Ethernet/IEEE 802.3 interface(s)
2 serial(sync/async) network interface(s)
System/IO memory with parity disabled
2048K bytes of DRAM onboard
System running from FLASH
8K bytes of non-volatile configuration memory.
6144K bytes of processor board PCMCIA flash (Read ONLY)

Configuration register is 0x2102
```

Из результата ее выполнения можно определить, например, версию операционной системы — 11.3(3)T. Далее можно узнать время непрерывной работы маршрутизатора — Cisco1601 uptime is 6 days, 22 hours, 54 minutes, или имя загрузочного файла маршрутизатора — flash:c1600-y-l.113-3.T.bin. Интересные сведения также помещаются в строку, указывающую на причину последней перезагрузки маршрутизатора. Такой причиной бывает выключение питания (как в нашем примере — System restarted by power-on) или произошедший сбой. В этом случае можно наблюдать следующую запись: System restarted by bus error at... Хочется отметить, что автор с последней ситуацией ни разу не сталкивался.

В последней строке выводится содержимое конфигурационного регистра (Configuration register) в шестнадцатеричном формате. Конфигурационный регистр контролирует, например, процесс загрузки маршрутизатора. Следует сказать, что результат работы команды show version различается в зависимости от модели маршрутизатора и, как уже говорилось, с ее помощью можно получить достаточный объем информации о маршрутизаторе и его программном обеспечении.

Проверка содержимого флэш-памяти осуществляется с помощью команды show flash. Ниже приведен пример выполнения этой команды на маршрутизаторе Cisco 805.

```
Cisco805#show flash
Directory of flash:/

  0  ----       49096   May 15 1999 03:55:31  TinyROM-1.2(1)
  1  -r-x     2287176   Jun 18 1999 05:18:51  c805-y6-mw.120-4.XM

4194304 bytes total (1835008 bytes free)
Cisco805#
```

Отсюда видно, что существует некий файл, представляющий собой образ операционной системы Cisco IOS (c805-y6-mw.120-4.XM). При этом флэш-память может содержать несколько файлов — в качестве ограничителя выступает только размер свободного пространства.

Компанией Cisco Systems принята собственная схема именования файлов, в которых размещена операционная система IOS. Эта схема позволяет по имени файла определить версию операционной системы и модель маршрутизатора, для которой она предназначена. Как видно из предыдущего примера, имя файла состоит из нескольких секций, разделенных точкой. При этом первая секция имеет три группы символов, разделяемые знаком -. Первая группа (здесь c805) указывает на платформу маршрутизатора, вторая определяет набор возможностей (feature set), а y6-mw именно указывает на ограниченный набор возможностей по поддержке протокола IP — reduced IP variant. Вторая секция содержит идентификатор версии операционной системы — в нашем примере это 12.0(4). Для более детального ознакомления со схемой именования файлов в операционной системе Cisco IOS необходимо обратиться к документу Cisco IOS Reference Guide, размещенному в Интернете по адресу http://www.cisco.com/warp/public/620/4.html.

Работа с аппаратными интерфейсами маршрутизатора

Для более наглядного представления о расположении аппаратных интерфейсов приведем рисунок, на котором изображена задняя панель маршрутизатора (рис. 1.6)

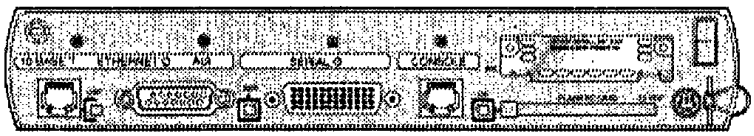


Рис. 1.6. Интерфейсы маршрутизатора

В табл. 1.3 представлено подробное описание каждого интерфейса и выполняемые ими функции.

Таблица 1.3. Описание интерфейсов маршрутизатора

Интерфейс	Задачи
ETHERNET ∅ 10BASE T	Подключение маршрутизатора к локальной сети на скорости 10 Мбит/с
CONSOLE	Подключение маршрутизатора к компьютеру с помощью консольного кабеля. На стороне маршрутизатора консольный разъем имеет тип RJ45
WIC	Свободный слот для дополнительного интерфейсного модуля, с помощью которого, например, можно организовать еще один интерфейс Serial (скорость передачи до 2 Мбит/с)
FLASH PC CARD	Слот для установки флэш-карты, содержащей образ операционной системы Cisco IOS. Новые маршрутизаторы поставляются с предустановленной картой
SERIAL ∅	Для подключения маршрутизатора к выделенным каналам связи (leased lines), сети Frame Relay и т. п.

В модели маршрутизатора Cisco 1601 присутствует свободный слот для установки дополнительного интерфейсного модуля (WIC). Если после покупки маршрутизатора появилась необходимость в организации еще одного подключения, то администратор может просто докупить нужный ему модуль, что получается дешевле приобретения нового маршрутизатора (рис. 1.7).

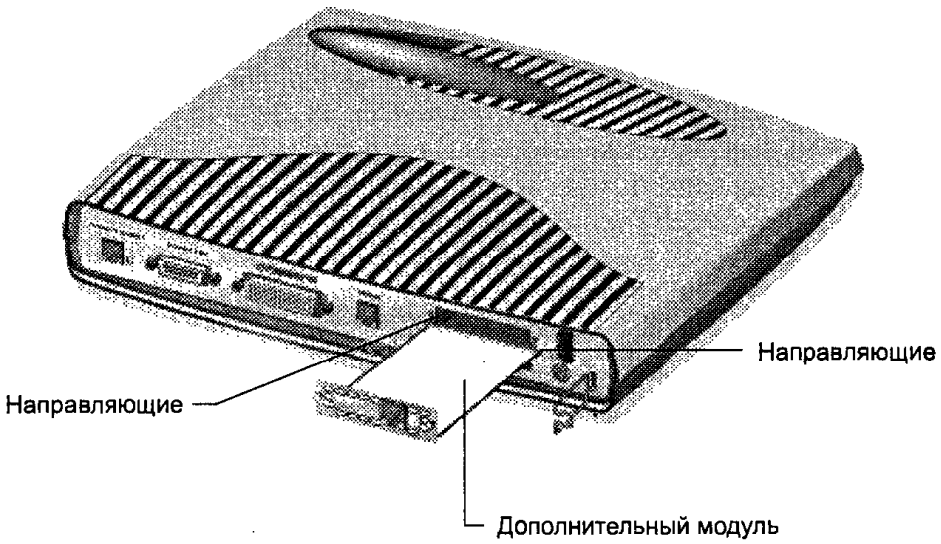


Рис. 1.7. Установка дополнительного модуля в маршрутизатор

Связь между интерфейсами и внутренними компонентами маршрутизатора осуществляется через контроллеры. Контроллеры предназначены для организации взаимодействия между отдельными компонентами маршрутизатора. Существует множество типов контроллеров, и их наличие или отсутствие зависит от модели маршрутизатора и его внутренней конфигурации.

Для того чтобы просмотреть информацию о контроллерах, например узнать тип аппаратного обеспечения, статистику, версию программного кода и т. п., нужно выполнить команду show controllers. Информация, выводимая этой командой, может размещаться на множестве строк, поэтому в следующем примере ограничимся только первыми несколькими строками.

```
Cisco1601#show controllers ethernet
QUICC Ethernet unit 0 using SCC1, Microcode ver 4
Current station address 0010.7bef.4478, default address 0010.7bef.4478
idb at 0x207B9A4, driver data structure at 0x2079490
...
```

В результате работы этой команды администратор получил информацию о контроллере Ethernet, в который на данной модели маршрутизатора встроена микросхема QUICC.

Приведем пример работы этой команды для модели Cisco 2511RJ. Как видно, в ней применяется набор микросхем LANCE. Кроме того, из результатов работы команды можно опередить MAC-адрес (аппаратный адрес устройства), присвоенный интерфейсу (station address).

```
Cisco2511RJ#show controllers ethernet
LANCE unit 0, idb 0x8E3A4, ds 0x8FE30, regaddr = 0x2130000, reset_mask 0x2
IB at 0x206DAC: mode=0x0000, mcfilter 0000/0000/0100/0000
station address 0010.7b3a.b6de default station address 0010.7b3a.b6de
buffer size 1524
...
```

Теперь применим команду **show controllers** для интерфейса Serial маршрутизатора. По результату работы можно определить тип кабеля, подключенного к данному интерфейсу, — V.35 DTE (Data Terminal Equipment cable). Тут следует отметить, что подразумевается наличие кабеля, свободный конец которого после подключения к маршрутизатору имеет разъем интерфейса V.35 (тип male). В нашем случае к данному разъему подключен модем для физических линий компании RAD, который представляет собой оконечное оборудование (DCE, Data Circuit-terminating Equipment) стороны подключения (тип female).

```
Cisco2511RJ#show controllers serial
HD unit 0, idb = 0x94364, driver structure at 0x98010
buffer size 1524  HD unit 0, V.35 DTE cable
cpb = 0x21, eda = 0x4878, cda = 0x488C
RX ring with 16 entries at 0x214800
...
```

В настоящее время предлагаются шесть типов кабелей для интерфейса Serial, которые в англоязычной литературе могут называться serial adapter cables или serial transition cables: EIA/TIA-232, EIA/TIA-449, V.35, X.21, EIA/TIA-530 и EIA/TIA-530A. Все эти кабели имеют универсальный разъем на конце, подключаемом к маршрутизатору, в то время как другие концы кабелей (так называемые сетевые) отличаются типом разъема. Например, кабель EIA/TIA-232 на сетевом конце завершается популярным разъемом DB-25. Все кабели, за исключением EIA-530, доступны в формах DCE и DTE. Само подключение кабеля выполняется очень просто и требует элементарной аккуратности, так как при подсоединении кабеля к маршрутизатору можно погнуть штырьки на разъеме, а с учетом большой плотности их расположения и немалого количества процесс починки может затянуться надолго (рис. 1.8).

:ОВЕТ

Более подробную информацию о кабелях и их распайке можно найти, например, по следующему адресу: http://www.cisco.com/en/US/products/hw/routers/ps233/products_installation_and_configuration_guide_chapter09186a008007e1cd.html.

Одним из наиболее популярных типов кабелей, используемых для подключения оборудования к интерфейсу маршрутизатора Serial, является 32-штырьковый кабель интерфейса V.35, хотя в последнее время автор все чаще сталкивался с оборудованием, имеющим разъем DB-25 (например, модемы серии xDSL WATSON). Если администратору нужен именно кабель V.35, то, как уже отмечалось, при его выборе следует обращать внимание на тип разъема оборудования, которое планируется подключить к маршрутизатору. Выбираемый кабель

может заканчиваться разъемом типа female (DCE) или male (DTE). Чаще всего подключаемое оборудование имеет разъем female, что и определяет тип заказываемого кабеля (рис. 1.9).

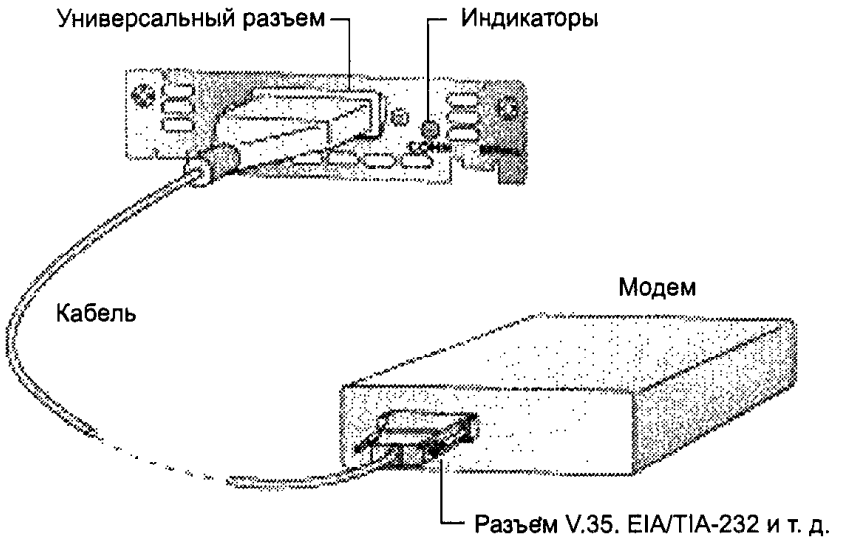


Рис. 1.8. Подключение оборудования к интерфейсу Serial маршрутизатора

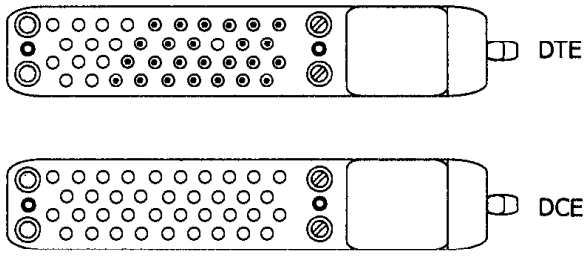


Рис. 1.9. Разъемы интерфейса V.35 (DTE и DCE)

Помимо просмотра информации об аппаратной части интерфейса на маршрутизаторе часто возникает крайняя необходимость проверить статус интерфейса и получить статистику его работы. Причем после выполнения всех необходимых шагов настройки маршрутизатора и подключения его к действующей сети проверяется именно статус интерфейса. Он состоит из двух компонентов, которые соответствуют требованиям эталонной модели OSI, — статус для физического и канального уровней. Первый компонент, статус физического уровня, указывает на то, что интерфейс прошел диагностику и получил необходимые сигналы от подключенного оборудования.

Следующий компонент статуса, соответствующий канальному уровню, указывает на получение интерфейсом так называемых сообщений *keepalive* («пока жив»), которые представляют собой небольшие сообщения канального уровня, рассылаемые сетевыми устройствами для подтверждения своей доступности. Если поддержка этих сообщений не отключена, то по умолчанию они передаются через интерфейс каждые десять секунд. Интерфейсы, подключенные к глобальной

сети, переводятся в рабочее состояние (в информационных сообщениях на маршрутизаторе это обозначается как Up) в том случае, если они получают такие сообщения. На интерфейсах локальной сети маршрутизатор посылает эти сообщения сам себе, и при их получении интерфейс также переходит в рабочее состояние. В табл. 1.4 перечислены комбинации статусов двух компонентов интерфейса с расшифровкой возможных причин их состояний.

Таблица 1.4. Комбинации статусов и причины их установки

Статус физического уровня	Статус канального уровня	Возможные причины
Up	Up	Интерфейс находится в рабочем состоянии
Up	Down	Физическое подключение интерфейса работает, однако сообщения keepalive не получены
Down	Down	Не работает физическое подключение интерфейса
Administratively Down	Down	Интерфейс был вручную отключен администратором сети (shutdown)
Up	Up (looped)	Интерфейс глобальной сети получает свои собственные сообщения keepalive

Интерфейс, находящийся в рабочем состоянии, обычно обозначается как UP/UP. Это говорит о том, что оба компонента интерфейса работают. В том случае, если используемый протокол канального уровня находится в рабочем состоянии, а интерфейс не получает сообщения keepalive в течение трех временных интервалов, компонент канального уровня изменяет свое состояние на нерабочее (down). Для того чтобы проверить состояние этих двух компонентов интерфейса и просмотреть статистику его работы, можно воспользоваться командой show interface. Выполнение этой команды без указания параметров позволит получить информацию обо всех доступных интерфейсах на маршрутизаторе. Если необходимо проверить какой-либо определенный интерфейс, нужно выполнить рассматриваемую команду с именем интересующего интерфейса в качестве параметра. Из приведенного ниже примера видно, что оба компонента интерфейса маршрутизатора Cisco 1005, подключенного к сети Frame Relay, находятся в рабочем состоянии.

```
Cisco1005#sh in s0
Serial0 is up, line protocol is up
   Hardware is QUICC Serial
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
   LMI enq sent  18350, LMI stat recvd 18350, LMI upd recvd 0, DTE LMI up
   LMI enq recvd 0, LMI stat sent  0, LMI upd sent  0
   LMI DLCI 0  LMI type is ANSI Annex D  frame relay DTE
...
```

Вообще говоря, из результата работы команды keepalive администратор сети может извлечь очень много полезной информации. В рассматриваемом примере

интерфейс Serial0 подключен к сети Frame Relay, и на нем настроен подинтерфейс (subinterface). Как видно, статус самого интерфейса выдается в первой строке результата, далее выводится статистика работы протокола LMI (настройка маршрутизаторов для использования в сети Frame Relay будет рассмотрена в главе 8), используемый тип инкапсуляции, значение минимального передаваемого блока (MTU, Maximum Transfer Unit) в байтах, пропускная способность BW (bandwidth), задержка DLY (delay) и т. п.

Следует отметить, что установка значения пропускной способности канала связи на интерфейсе не имеет ничего общего со скоростью самого интерфейса. Данная установка значима для протоколов и приложений верхних уровней. Например, протокол маршрутизации EIGRP привлекает установленное значение пропускной способности для расчета метрики маршрута, а приложение управления сетью по протоколу SNMP может использовать это значение для расчета степени загруженности канала. Пропускная способность устанавливается в килобитах в секунду (Кбит/с). При этом значение по умолчанию составляет 1544 Кбит/с. Для того чтобы изменить значение, заданное по умолчанию, следует воспользоваться командой bandwidth, выполняемой в режиме конфигурирования интерфейсов с новым значением пропускной способности в качестве параметра.

MTU определяет максимальный размер пакета, который может быть получен или передан через данный интерфейс (по умолчанию это значение составляет 1500 байт). Подробнее рассмотрим вопрос оценки оптимального MTU. При выборе MTU необходимо следовать простому правилу: назначать меньшее значение из всех возможных допустимых значений в каналах по пути прохождения пакета. Следствием выполнения данного правила будет повышение пропускной способности канала за счет того, что маршрутизатору не придется разбивать один пакет на несколько, если размер пакета сетевого уровня в следующем канале должен быть меньше, чем в предыдущем. Следует помнить, что назначение заниженного значения MTU также ограничит пропускную способность между двумя конечными станциями или между двумя сетями одного и того же типа.

Из сказанного становится понятно, что присвоение оптимального значения MTU является очень важным этапом настройки сети. Правда, некоторые протоколы сами выбирают значения MTU, которые подходят для всех известных типов сетей (обычно 576 байтов), но опять же за счет сокращения пропускной способности. Иные протоколы обязывают выбирать максимальный размер пакета при конфигурировании драйвера локальной сети на каждом компьютере в сети. Существуют протоколы, производящие анализ пути между двумя узлами во время установления соединения. Они позволяют определить наибольшее значение MTU, при котором не потребуется производить фрагментацию кадров.

Задача усложняется, если путь проходит по каналам глобальной сети. Это относится и к варианту соединения каналов «точка-точка». Конфигурация маршрутизатора содержит параметры MTU для каждого интерфейса глобальной сети, один из которых необходимо установить. Сложность состоит в том, что требуется передавать по глобальной сети как можно большее число пакетов без фрагментации и с минимальным временем задержки за счет сокращения длины пакетов. Необходимо назначать MTU так, чтобы пропускная способность была максимальна, а фрагментация минимальна.

Рассмотрим два сценария передачи файла размером в миллион байтов. В первом сценарии файл отправляется с устройства А посредством 4000-байтовых пакетов; во втором — с устройства В посредством 500-байтовых пакетов. Так как маршрутизатор обрабатывает пакеты по мере поступления и каждый пакет приводит к небольшой, но не нулевой задержке в интерфейсе глобальной сети, то устройство А будет иметь преимущество при передаче от маршрутизатора к маршрутизатору на пути следования просто потому, что его пакеты длиннее. С другой стороны, если размер пакета в глобальной сети ограничен значением 500 байт, то устройство В получит преимущество вследствие дополнительных затрат времени на фрагментацию и сборку пакетов устройством А.

Обобщая сказанное, можно прийти к выводу, что наилучший способ выбора оптимального MTU для интерфейсов маршрутизаторов и компьютеров в сети состоит в определении часто используемых маршрутов передачи данных и настройке значений MTU на выбранных компьютерах и интерфейсов маршрутизаторов до тех пор, пока не будет получен желаемый результат.

Операционная система Cisco IOS поддерживает механизм обнаружения MTU (Path MTU), как это определено в документе RFC 1191. Этот механизм позволяет маршрутизатору автоматически обнаруживать и учитывать различия между максимально допустимыми значениями MTU на различных переходах в распределенной сети. Иногда маршрутизатор не может переслать дейтаграмму протокола IP (главу 2), так как требуется фрагментация. А дейтаграмма имеет больший размер, чем установленное администратором значение MTU для интерфейса. Но в заголовке этой дейтаграммы установлен бит, указывающий на то, что фрагментацию выполнять нельзя (бит Don't Fragment — DF). В таком случае маршрутизатор посылает сообщение отправителю дейтаграммы, в котором говорится о необходимости выполнения фрагментации, с тем чтобы фрагменты подходили к наименьшему значению MTU в пути следования. Механизм обнаружения MTU полезен также в случае, когда один из каналов в распределенной сети выходит из строя. Тогда может быть, например, задействован резервный канал, но с другим значением MTU.

Рассмотрим простой пример, когда дейтаграмма проходит через сеть, где MTU на первом маршрутизаторе равен 1500 байт, а на втором — 512 байт. Если отправитель установил в дейтаграмме бит, запрещающий выполнять фрагментацию, то все дейтаграммы, превышающие по размеру 512 байт, вторым маршрутизатором будут отбрасываться. Этот маршрутизатор пошлет отправителю сообщение протокола ICMP (Destination Unreachable) с указанием необходимости проведения фрагментации. При поддержке механизма PMTU в неиспользуемых полях сообщения ICMP будет содержаться MTU на следующем переходе. Описываемый механизм полезен и в случаях, когда соединение устанавливается в первый раз и отправитель не имеет информации обо всех переходах в сети. Всегда желательно назначать как можно большее значение MTU, так как это позволит повысить производительность. В этом случае потребуется отправлять меньшее количество пакетов и, следовательно, накладные расходы сократятся.

Маршрутизаторы серии Cisco 7000 и Cisco 4000 поддерживают передачу дейтаграмм IP между интерфейсами Ethernet и FDDI. При этом в случае передачи дейтаграмм из сети FDDI в сеть Ethernet без использования механизма PMTU пакеты FDDI с объемом данных более 1500 байт будут фрагментированы

в несколько пакетов для сети Ethernet, что приведет к снижению производительности. Если в сети превалирует объем трафика из сети FDDI, то можно либо пойти на уменьшение размера MTU на устройствах в сети FDDI, либо включить на них поддержку механизма PMTU.

Пропускная способность и задержка используются некоторыми процессами в маршрутизаторе, которые требуют фиксированных значений этих параметров. Следует отметить, что данные параметры не влияют на реальную скорость интерфейса. Дополнительно в результате работы команды show interface можно узнать значение параметра надежности rely (reliability), показывающее, насколько стабильна и надежна работа интерфейса — значение 255 указывает на 100% надежность.

Если для интерфейса был указан адрес протокола IP, то информация о нем также выдается на экран (в предыдущем примере для интерфейса Serial0 адрес назначен не был — в данном случае адрес был назначен для подинтерфейса). Кстати, если посмотреть на команду, которая инициирует выдачу информации, то можно увидеть, что используются только первые символы отдельных команд (sh in s0). Это значительно упрощает работу администратора сети за счет экономии времени.

Точно так же команду show interface действует для интерфейса локальной сети Ethernet, и если посмотреть на результат, то можно узнать используемый тип инкапсуляции — в следующем примере это ARPA (Advanced Research Projects Agency). Это значит, что задействован формат кадра Ethernet II технологии Ethernet. Далее видно, что поддержка сообщений keepalive на интерфейсе Ethernet0 маршрутизатора Cisco 1005 не установлена, в то время как операционная система Cisco IOS ожидает каждые 10 секунд получения этих сообщений по интерфейсу Serial0.

```
Cisco1005#sh int eth0
Ethernet0 is up, line protocol is up
   Hardware is QUICC Ethernet, address is 00e0.1e80.5ea0 (bia 00e0.1e80.5ea0)
   Description: connected to Moskovsky 207 LAN
   Internet address is 192.168.6.254/24
   MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
   Encapsulation ARPA, loopback not set, keepalive not set
   ARP type: ARPA, ARP Timeout 04:00:00
   Last input 00:00:34, output 00:00:56, output hang never
   Last clearing of "show interface" counters never
   Queueing strategy: fifo
   Output queue 0/40, 0 drops; input queue 0/75, 0 drops
   5 minute input rate 0 bits/sec, 0 packets/sec
   5 minute output rate 0 bits/sec, 0 packets/sec
      49748 packets input, 9071864 bytes, 0 no buffer
      Received 24409 broadcasts, 0 runts, 0 giants, 0 throttles
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
      0 input packets with dribble condition detected
      104322 packets output, 48947868 bytes, 0 underruns
      43456 output errors, 0 collisions, 2 interface resets
      0 babbles, 0 late collision, 5 deferred
      43456 lost carrier, 0 no carrier
      0 output buffer failures, 0 output buffers swapped out
```

Из результатов работы команды видно, что счетчики статистики интерфейса (counters) никогда не сбрасывались принудительно (**Last clearing of "show interface" counters never**). В следующих строках отображаются собственно сами счетчики данного интерфейса. Основываясь на полученной статистике работы интерфейса, можно определить количество полученных и переданных пакетов, произошедших ошибок при передаче и почерпнуть другую информацию, например о числе коллизий в сети Ethernet. Небольшое число коллизий объясняется тем, что локальная сеть, подключенная к маршрутизатору, состоит всего из нескольких компьютеров, которые большую часть времени находятся в неактивном состоянии.

И хотя сведения об интерфейсах локальной сети могут быть полезными администратору, чаще всего интерес вызывает информация об интерфейсах глобальной сети. Например, если распределенная сеть включает в себя локальные сети сторонних организаций (например, партнеров компании), то администратор, опираясь на статистику работы интерфейса глобальной сети, может собирать информацию для учета и т. п. Остановимся на информации об интерфейсе, через который проходит наибольший объем трафика (подключенном к Интернету на скорости 512 Кбит/с).

```
Cisco2511RJ#show interfaces serial 0
Serial0 is up, line protocol is up
   Hardware is HD64570
   Internet address is 193.125.189.158/30
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation HDLC, loopback not set, keepalive set (10 sec)
   Last input 00:00:07, output 00:00:05, output hang never
   Last clearing of "show interface" counters never
   Input queue: 0/75/14 (size/max/drops); Total output drops: 0
   Queueing strategy: weighted fair
   Output queue: 0/64/0 (size/threshold/drops)
      Conversations  0/15 (active/max active)
      Reserved Conversations 0/0 (allocated/max allocated)
   5 minute input rate 0 bits/sec, 0 packets/sec
   5 minute output rate 0 bits/sec, 0 packets/sec
      3454178 packets input, 1999455824 bytes, 14 no buffer
      Received 91212 broadcasts, 0 runts, 0 giants
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
      4962814 packets output, 395288471 bytes, 0 underruns
      0 output errors, 0 collisions, 1 interface resets
      0 output buffer failures, 0 output buffers swapped out
      0 carrier transitions
      DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Интерфейс имеет свой собственный адрес сетевого уровня, так как он обслуживает выделенный канал связи и к нему подключен модем для физических линий (модель RAD ASMi-50). Из полученной информации можно узнать используемый тип инкапсуляции — HDLC (High-level Data Link Control), что отличается от инкапсуляции, используемой для интерфейса локальной сети (Encapsulation ARPA) и для интерфейсов, подключенных к сети Frame Relay (Encapsulation

FRAME-RELAY). Исследуемый интерфейс не имеет своего собственного MAC-адреса, а пропускная способность интерфейса составляет 1 544 Кбит/с (или 1,5 Мбит/с), что соответствует пропускной способности канала T-1.

Инкапсуляция HDLC принята по умолчанию для всех синхронных интерфейсов маршрутизаторов компании Cisco Systems. Но администратор волен выбирать тип инкапсуляции для выделенных каналов связи на свое усмотрение. Например, HDLC можно заместить протоколом PPP, хотя если выполняется подключение к Интернету, то данный вопрос следует согласовать с интернет-провайдером.

ИЗ ОПЫТА

Автор на опыте столкнулся с ситуацией, когда самым простым техническим решением была смена инкапсуляции с PPP на HDLC. Маршрутизатор Cisco 2511RJ обслуживал внешних пользователей, попадающих в сеть через модем. В качестве протокола канального уровня работал PPP. Для авторизации на маршрутизаторе был настроен протокол TACACS+ в сочетании с внешним сервером с базой данных пользовательских имен и паролей. Однако после настройки маршрутизатора на работу с протоколом авторизации синхронный интерфейс, подключенный к Интернету, переходил в состояние Down. Это было вызвано тем, что операционная система также пыталась провести авторизацию маршрутизатора провайдера, рассматривая его как одного из удаленных пользователей, работающего по протоколу PPP.

Из информации об интерфейсе, которая выводится командой show interfaces, видно также, что включена поддержка сообщений keepalive. Это означает, что рассматриваемый маршрутизатор будет ожидать получения таких сообщений от маршрутизатора провайдера каждые 10 секунд. Команда show interface выдает достаточно большой объем информации, и если нужны лишь ключевые данные о работе интерфейса, можно выполнить команду show ip interface с параметром brief. В результате получим статус всех интерфейсов в виде нескольких строк.

Рассмотрим пример, в котором эта команда была выполнена на маршрутизаторе Cisco 1601, подключенном к локальной сети, к сети Frame Relay и к выделенному каналу связи. Маршрутизатор имеет один интерфейс локальной сети (Ethernet0) и два интерфейса глобальной сети (Serial0 и Serial1). Интерфейсы Serial1.x являются подинтерфейсами, настроенными на интерфейсе Serial1.

```
Cisco1601#show ip interface brief
Interface    IP-Address      OK?  Method      Status       Protocol
Ethernet0    192.168.4.254   YES  NVRAM    up           up
Serial0      192.168.4.254   YES  unset    up           up
Serial1      unassigned      YES  unset    up           up
Serial1.1    192.168.4.254   YES  unset    up           up
Serial1.2    192.168.4.254   YES  unset    up           up
Serial1.3    192.168.4.254   YES  unset    up           up
```

ИЗ ОПЫТА

Пожалуй, наиболее часто обращаются именно к статистике трафика через интерфейс глобальной сети, в частности через интерфейс, подключенный к Интернету. При использовании выделенного канала связи с провайдером последний чаще всего взимает дань за полученную из Интернета информацию (скажем, за каждый входящий гигабайт). Опираясь на полученную статистику, администратор приобретает в свое распоряжение механизм, позволяющий соотносить объем информации с ее денежным эквивалентом.

Терминальные порты

Компоненты маршрутизатора (как логические, так и физические), которые позволяют осуществить доступ к интерфейсу командной строки маршрутизатора и вводить команды, обобщенно можно назвать *терминальными портами* (terminal lines). Выделяют четыре типа терминальных портов, часть из которых уже была рассмотрена: консольный порт (CTY/CON), порт AUX, асинхронные (TTY) и виртуальные (VTY) порты.

Если консольный порт предполагает наличие физического доступа к маршрутизатору, то в случае, когда маршрутизатор уже настроен и работает на удаленном объекте распределенной сети, можно выполнять необходимые настройки (добавляя новые или изменяя старые) с помощью установления сеанса Telnet. Все, что для этого требуется, — это указать любой IP-адрес маршрутизатора. После этого маршрутизатор «попросит» ввести имя и пароль или немедленно войдет в пользовательский режим.

Следует отметить, что по умолчанию в целях повышения безопасности операционная система Cisco IOS не предпринимает попытки подключения к маршрутизатору в сеансе Telnet. Для включения поддержки входящих сеансов Telnet нужно подключиться к консольному порту маршрутизатора и соответствующим образом настроить виртуальный (логический) порт (VTY). Все маршрутизаторы компании Cisco Systems поддерживают пять подобных портов, хотя не исключается и большее их количество, если требуется поддерживать более пяти одновременных сеансов Telnet с маршрутизатором.

Самый простой способ проверки состояния терминальных портов заключается в выполнении команды show line в пользовательском или привилегированном режиме.

```
Cisco1601#show line
   Tty Typ  Tx/Rx A Modem  Roty     AccO     AccI  Uses     Noise Overruns
   0 CTY    -     -        -        -        -     0        0     0/0
*  3 VTY    -     -        -        -        -     11206    0     0/0
   4 VTY    -     -        -        -        -     11220    0     0/0
   5 VTY    -     -        -        -        -     29       0     0/0
   6 VTY    -     -        -        -        -     0        0     0/0
   7 VTY    -     -        -        -        -     0        0     0/0
```

Результат работы команды представляется в виде таблицы. Первые два столбца — Tty и Typ — содержат номера портов (аналогично номеру интерфейса), отсчитываемые с нуля, и типы портов. В приведенном примере показаны только два типа терминальных портов. Если бы работали асинхронные порты, то они были бы обозначены и повлияли бы на нумерацию, как это показано в следующем примере для маршрутизатора Cisco 2511RJ, имеющего 16 асинхронных портов для подключения модемов. По результатам работы команды можно определить задействованные порты, которые отмечаются символом *. Видно, что на связи присутствуют два удаленных пользователя, подключившихся по модему, и один пользователь, работающий с маршрутизатором через сеанс Telnet (порт 18 VTY). Администратору могут быть интересны данные в столбце Noise, показывающие количество ошибок при передаче информации через порт.

```
Cisco2511RJ#show line
Tty Typ     Tx/Rx            A  Modem  Roty  AccO  AccI  Uses  Noise  Overruns
0 CTY                        -  -      -     -     -     0     3      0/0
I 1 TTY     115200/115200    -  inout  -     -     -     115   39     5/0
I 2 TTY     115200/115200    -  inout  -     -     -     99    11     3/0
...
*15 TTY     115200/115200    -  inout  -     -     -     0     0      0/0
*16 TTY     115200/115200    -  inout  -     -     -     0     0      0/0
 17 AUX     9600/9600        -  -      -     -     -     0     0      0/0
*18 VTY                      -  -      -     -     -     11165 0      0/0
 19 VTY                      -  -      -     -     -     82    0      0/0
 20 VTY                      -  -      -     -     -     70    0/0
 21 VTY                      -  -      -     -     -     00    0/0
 22 VTY                      -  -      -     -     -     00    0/0
```

Подробную информацию о каком-либо конкретном порте можно получить при помощи команды **show line**, указав номер порта в качестве параметра:

```
Cisco2511RJ#show line 2
 Tty Typ  Tx/Rx          A  Modem  Roty  AccO  AccI  Uses  Noise  Overruns
I  2 TTY  115200/115200  -  inout  -     -     -     99    11     3/0
Line 2, Location: "connected to Internet Users - 16", Type: ""
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 115200/115200, no parity, 1 stopbits, 8 databits
...
```

В том случае, когда необходимо получить информацию о том, кто именно задействовал тот или иной порт, можно выполнить команду **show user**. В приведенном ниже примере эта команда отработала для маршрутизатора Cisco 2155RJ, и по ее результатам видно, что подключены два пользователя. При этом один пользователь (**xuser1**) подключен к асинхронному порту, а второй (**kulgin**) через сеанс Telnet:

```
Cisco2511RJ_ONE#show users
Line User       Host(s)              Idle Location
7  tty 7 xuser1 Async interface      0 connected to Internet Users - 16
9  tty 9        Modem Autoconfigure  0 connected to Internet Users - 16
10 tty 10       Modem Autoconfigure  0 connected to Internet Users - 16
11 tty 11       Modem Autoconfigure  0 connected to Internet Users - 16
12 tty 12       Modem Autoconfigure  0 connected to Internet Users - 16
13 tty 13       Modem Autoconfigure  0 connected to Internet Users - 16
14 tty 14       Modem Autoconfigure  0 connected to Internet Users - 16
15 tty 15       Modem Autoconfigure  0 connected to Internet Users - 16
16 tty 16       Modem Autoconfigure  0 connected to Internet Users - 16
*8 vty 0 kulgin idle                 0 proxy.alternativa.spb.ru
```

Команда **show users** показывает также (если это возможно) место (location), откуда выполнялось подключение. Для виртуальных портов VTY это может быть отмечено либо IP-адресом, либо именем хоста, с которого был инициирован сеанс Telnet. Как видно из предыдущего примера, пользователь **kulgin** подключился к маршрутизатору с хоста **proxy.alternativa.spb.ru**, хотя на самом деле

это не совсем соответствует действительности. В реальности компьютер, с которого инициировался сеанс Telnet, располагался за прокси-сервером (proxy server), в результате чего работа с маршрутизатором велась с использованием адреса и от имени последнего.

Проверка аппаратных ресурсов маршрутизатора

Рассмотрим команды, которые можно использовать для проверки состояния операционной системы Cisco IOS на маршрутизаторе. Выполняемая в пользовательском и в привилегированном режимах команда **show processes cpu** позволяет понять, насколько занят процессор маршрутизатора и какие именно процессы операционной системы Cisco IOS выполняются в данный момент.

```
Cisco1005#show processes cpu
CPU utilization for five seconds: 12%/10%; one minute: 24%; five minutes: 25%
PIDRuntime(ms)  Invoked   uSecs    5Sec     1Min     5Min   TTY Process
1  41144        1170530   35       0.00%    0.00%    0.00%  0   Load Meter
2  272          51        5333     1.55%    0.21%    0.04%  1   Virtual Exec
3  119196568    1052518   113249   0.00%    2.42%    2.12%  0   Check heaps
...
```

В первых строках показана средняя загруженность процессора за три последних интервала времени: 5 секунд, 1 минуту и 5 минут. При этом результат, соответствующий пятисекундной загрузке, содержит два числа в процентах, разделенных наклонной чертой. Первое число (12%) обозначает среднюю загрузку процессора в течение этого временного интервала, а второе число (10%) показывает долю времени процессора, затраченную на обслуживание прерываний, например прерываний на обработку поступающих пакетов. В следующих строках отображена информация обо всех активных в настоящее время процессах. При этом выводится идентификатор процесса (Process ID — PID), его имя и некоторая статистика. В приведенном примере результаты работы команды представлены в сокращенном виде, так как на маршрутизаторе может быть запущено множество процессов одновременно.

Память в маршрутизаторе, является, пожалуй, наиболее ценным ресурсом. Операционная система предоставляет в распоряжение администратора две команды для просмотра состояния памяти, которые можно выполнять в пользовательском или привилегированном режиме. Следующий пример показывает работу первой из этих двух команд — **show process memory**.

```
Cisco1005#show processes memory
Total: 1550152, Used: 816588, Free: 733564
PIDTTY   Allocated   Freed    Holding   Getbufs   Retbufs   Process
0  0     154356      1236     718924    0         0         *Init*
0  0     236         38352    236       0         0         *Sched*
0  0     869276      175864   464       277180    55040     *Dead*
...
```

Из первой строки листинга можно получить очень важную информацию о памяти маршрутизатора. А именно наличный объем памяти на маршрутизаторе — Total: 1550152, сколько ее задействовано — Used: 816588 и сколько свободно — Free: 733564. Тот объем памяти, который занимают работающие процессы, можно узнать из столбца Holding. Если с течением времени наблюдается постепенное снижение объема доступной памяти, то можно проверить, какой именно процесс виновен в этой тенденции. Обычно подобные проблемы решаются при обновлении версии операционной системы Cisco IOS. Для получения детальной статистики по использованию памяти предусмотрена команда show memory, которая имеет множество параметров, конкретизирующих результаты ее выполнения:

```
Cisco2511RJ#show memory ?
   <0-4294967294>      Dump memory starting at <address>
   allocating-process  Show allocating process name
   dead                Memory owned by dead processes
   failures            Memory failures
   fast                Fast memory stats
   free                Free memory stats
...
```

Фрагмент результирующего вывода при выполнении команды без параметров приведен ниже. Видно, что результат содержит информацию о распределении памяти для процессора и памяти для процессов ввода/вывода (I/O). Вообще говоря, информация об этих компонентах может быть достаточно подробна, а результат выполнения данной команды может включать в себя сотни строк.

```
Cisco2511RJ#show memory
          Head    Total(b)  Used(b)   Free(b)   Lowest(b)  Largest(b)
Processor 489AC1  795668    1246792   548876    487972     521200
I/O       200000  2097152   458832    1638320   1511048    1638128
...
```

Таблицы, буферы и журналы маршрутизатора

Операционная система Cisco IOS имеет множество типов буферов (buffers). Наибольший интерес для администратора сети могут представить буферы сетевых пакетов (network packet buffers). Существуют шесть типов буферов сетевых пакетов, каждый из которых имеет свой собственный размер. Анализ данных буферов может дать ключевую информацию для решения некоторых проблем, например проблем, связанных с потерей или отбрасыванием пакетов. Для проверки буферов служит команда show buffers, выполняемая как в пользовательском, так и в привилегированном режиме.

Каждый буфер сетевых пакетов содержит пакеты, которые обрабатываются операционной системой. Когда пакет поступает на интерфейс маршрутизатора, из подходящего пула (pool) буферов выделяется буфер, и в него помещается пакет. Выбор пула буферов осуществляется на основе размера поступившего пакета,

при этом действует следующее правило: пакет помещается в наименьший подходящий для него буфер. Сами буферы классифицируются по размеру: небольшой (small), средний (middle), большой (big), очень большой (very big), огромный (large) и гигантский (huge). Например, при получении маршрутизатором пакета размером 500 байт операционная система выделит для этого пакета средний буфер (600 байт).

```
Cisco2511RJ>show buffers
Buffer elements:
      499 in free list (500 max allowed)
      10097407 hits, 0 misses, 0 created
Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50):
      50 in free list (20 min, 150 max allowed)
      125533 hits, 0 misses, 0 trims, 0 created
      0 failures (0 no memory)
Middle buffers, 600 bytes (total 25, permanent 25):
      23 in free list (10 min, 150 max allowed)
      41771 hits, 0 misses, 0 trims, 0 created
      0 failures (0 no memory)
Big buffers, 1524 bytes (total 74, permanent 50):
      73 in free list (5 min, 150 max allowed)
      4794623 hits, 464 misses, 890 trims, 914 created
      17 failures (0 no memory)
VeryBig buffers, 4520 bytes (total 10, permanent 10):
      10 in free list (0 min, 100 max allowed)
      4550 hits, 0 misses, 0 trims, 0 created
      0 failures (0 no memory)
Large buffers, 5024 bytes (total 0, permanent 0):
      0 in free list (0 min, 10 max allowed)
```

Как видно из примера выполнения команды show buffers на маршрутизаторе Cisco 2511RJ, первая строка для каждого пула буферов описывает их размер, количество буферов в пуле на данный момент и количество постоянных буферов в пуле. Последние никогда не удаляются из пула и выделяются при загрузке маршрутизатора. Рассмотрим более подробно большие буферы (big buffers), анализируя результат выполнения команды show buffers.

Число, предшествующее словам free list, указывает на количество буферов, которые в настоящее время не используются, или, иными словами, которые не содержат пакетов (73 in free list). Пул содержит 74 больших буфера, из которых доступно 73. Из этого следует, что для обработки пакетов выделен только один большой буфер. Дополнительно выдается информация о минимально и максимально возможном количестве больших буферов (5 min, 150 max allowed). Таким образом, если число больших буферов окажется меньше пяти, операционная система создаст дополнительные буферы. Максимальное количество указывает на то, что поддерживается до 150 больших буферов, которые могут потребоваться для обработки большого объема сетевого трафика. Следующие строки для больших буферов содержат различные счетчики, например счетчики количества попаданий в буфер, пропусков и т. п.

Интересен счетчик, фиксирующий количество попаданий пакетов в буфер (4794623 hits), который также указывает на успешные попытки выделить буфер при необходимости разместить в нем пакет. Нужно отметить, что значение этого счетчика у данного буфера является наибольшим среди всех остальных буферов. На основании этого можно сделать вывод о распределении размеров пакетов, поступающих на маршрутизатор (в среднем 1500 байт). В свою очередь счетчик пропусков (464 misses) указывает на неуспешные попытки выделить буфер. Счетчик обновляется при нехватке доступных буферов в списке свободных. Обычно при этом создаются новые буферы, до тех пор пока не будет достигнуто максимально возможное их количество. Счетчик удаления (890 trims) показывает количество уничтоженных буферов как неиспользуемых (то есть операционная система выделила буферы под поступающие пакеты, а затем удалила их, когда количество свободных буферов (in free list) достигло количества максимально разрешенных (max allowed)). Счетчик отказов (17 failures) указывает на количество неуспешных попыток выделения буферов из-за их занятости.

Выполняемая в пользовательском режиме команда show stacks дает информацию, которая может быть полезна для выявления причины сбоя маршрутизатора. Подробно на ней останавливаться не стоит, так как в реальной работе команда используется очень редко.

Очевидно, что основная задача маршрутизатора заключается в принятии решения о дальнейшей обработке поступившего пакета. При этом задействуется таблица маршрутизации (routing table), которая хранится в памяти маршрутизатора и поддерживается операционной системой Cisco IOS. Подобная таблица создается и поддерживается для каждого маршрутизируемого протокола, настроенного на маршрутизаторе. Для того чтобы просмотреть таблицу маршрутизации, следует воспользоваться командой show <имя протокола> route, которая требует указания имени протокола (это может быть ip, ipx, appletalk или decnet). Далее приведен результат выполнения данной команды для протокола IP.

```
Cisco805#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
   D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
   N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
   E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
   i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
   U - per-user static route, o - ODR, P - periodic downloaded static route
   T - traffic engineered route

Gateway of last resort is 192.168.101.1 to network 0.0.0.0

R  192.168.4.0/24 [120/1] via 192.168.4.254, 5d00h, Serial0
R  192.168.5.0/24 [120/2] via 192.168.4.254, 5d00h, Serial0
R  192.168.6.0/24 [120/2] via 192.168.4.254, 09:35:44, Serial0
R  192.168.1.0/24 [120/2] via 192.168.101.253, 01:10:23, Ethernet0
R  192.168.2.0/24 [120/2] via 192.168.4.254, 5d00h, Serial0
S  192.168.3.0/24 [1/0] via 192.168.101.1
C  192.168.101.0/24 is directly connected, Ethernet0
S* 0.0.0.0/0 [1/0] via 192.168.101.1
```

Таблица маршрутизации содержит все известные маршруты в распределенной сети. Как видно из результатов работы команды **show ip route**, перечисляются также сокращения (строка **Codes**) с соответствующей им расшифровкой. При помощи сокращений отмечается источник получения информации о маршруте. Заметим, что для запроса обобщающей информации о маршрутизации для протокола IP можно воспользоваться командой **show ip route summary**, которая дает количественную информацию о таблице маршрутизации с указанием объема занимаемой памяти, общем количестве известных сетей и т. п.

```
Cisco805#show ip route summary
Route Source  Networks    Subnets     Overhead  Memory (bytes)
Connected     1           0           56        144
Static        2           0           112       288
Rip           5           0           280       720
Total         8           0           448       1152
```

Самым простым способом просмотреть поддерживаемые и настроенные маршрутизируемые протоколы является выполнение в пользовательском режиме команды **show protocols**, которая к тому же вернет статус всех настроенных на маршрутизаторе интерфейсов.

```
Cisco805#show protocols
Global values:
   Internet Protocol routing is enabled
Ethernet0 is up, line protocol is up
   Internet address is 192.168.101.254/24
Serial0 is up, line protocol is up
   Interface is unnumbered. Using address of Ethernet0 (192.168.101.254)
```

Результат работы команды можно разделить на две основные части: глобальную (**Global values**) и интерфейсную (Ethernet0, Serial0 и т. п.). Глобальная часть показывает маршрутизируемые протоколы, и ее название обусловлено тем, что эти протоколы настраиваются в глобальном режиме маршрутизатора. Маршрутизатор, на котором была выполнена эта команда, работает только с протоколом IP (**Internet Protocol routing is enabled**).

Интерфейсная часть содержит информацию о статусе всех доступных интерфейсов и данные по настроенным для них маршрутизируемым протоколам. С помощью команды **show <имя протокола> interface** можно получить информацию о протоколе на определенном интерфейсе, указав в качестве параметров конкретный протокол (**ip, ipx, appletalk, decnet**) и интерфейс. Например, можно просмотреть информацию для интерфейса Serial0, поддерживающего выделенный канал связи (в примере приведена только часть выдаваемого результата).

```
Cisco805#show ip interface serial 0
Serial0 is up, line protocol is up
   Interface is unnumbered. Using address of Ethernet0 (192.168.101.254)
   Broadcast address is 255.255.255.255
   MTU is 1500 bytes
   Helper address is not set
   Directed broadcast forwarding is disabled
   Multicast reserved groups joined: 224.0.0.9
...
```

Ведение журнала (logging) позволяет информировать администратора сети о происходящих системных событиях с помощью отправки сообщений в предопределенное место. По умолчанию сообщения посылаются на консольный порт маршрутизатора. В операционной системе предусмотрен специальный буфер памяти, в котором сохраняются сообщения. Однако этот буфер имеет небольшой объем, и по мере его наполнения сообщения перезаписываются, начиная с самых старых (то есть можно сказать, что буфер организован циклически). Существуют восемь уровней ведения журнала, которые нумеруются от 0 до 7. Нумерация отражает важность, или приоритет, сообщений. Таблица 1.5 содержит краткое описание уровней сообщений.

Таблица 1.5. Уровни сообщений и их описание

Номер	Имя уровня	Описание
0	Emergencies	Система в нерабочем состоянии
1	Alerts	Требуется немедленное вмешательство
2	Critical	Критические условия работы
3	Errors	Ошибки
4	Warnings	Предупреждения
5	Notifications	Извещения
6	Informational	Информационные сообщения
7	Debugging	Отладочные сообщения

Сообщения, которые соответствуют уровням с нулевого по шестой, всегда начинаются с символа % и содержат специальный код. Этим кодом идентифицируется соответствующий процесс. Сообщение само по себе является текстовой строкой, которая описывает происходящие и происшедшие события. Администратор может использовать временные привязки для сообщений, чтобы обеспечить их соответствие точному системному времени. Для того чтобы просмотреть содержимое буфера сообщений, предусмотрена команда show logging. Ниже приведен результат выполнения этой команды на маршрутизаторе Cisco 2511RJ, по которому можно выяснить статистику смены состояний асинхронных портов маршрутизатора по мере подключения и отключения удаленных пользователей.

```
Cisco2511RJ>show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
      Console logging: level debugging, 6689 messages logged
      Monitor logging: level debugging, 0 messages logged
      Trap logging: level informational, 6693 message lines logged
      Buffer logging: level debugging, 6689 messages logged

Log Buffer (4096 bytes):
1w5d: %LINK-3-UPDOWN: Interface Async2, changed state to down
1w5d: %LINEPROTD-5-UPDOWN: Line protocol on Interface Async3, changed state
   to down
```

```
1w5d: %LINK-5-CHANGED: Interface Async3, changed state to reset
1w5d: %LINK-3-UPDOWN: Interface Async3, changed state to down
1w5d: %LINK-3-UPDOWN: Interface Async1, changed state to up
1w5d: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async1, changed state
   to up
...
```

Существует возможность настроить получателей сообщений от маршрутизатора. Такими получателями могут быть: сервер Syslog (сервер, который способен принимать сообщения, посылаемые маршрутизатором по сети), консольный порт, терминал, буфер, станция управления, поддерживающая протокол SNMP.

)ВЕТ

С точки зрения автора, наиболее привлекательным решением будет использование сервера Syslog, который может быть организован на компьютере, работающем под управлением операционной системы UNIX (процесс syslogd) или Microsoft Windows NT. При этом для получения сообщений необходимо предварительно настроить на маршрутизаторе IP-адрес этого компьютера.

Компания Cisco Systems предлагает программный комплекс CiscoWorks 2000, который включает в себя компонент Resource Manager Essentials, способный принимать сообщения Syslog для их последующего анализа. После того как администратор настроил на всех маршрутизаторах IP-адрес сервера с этим программным обеспечением, он имеет возможность получать не просто текстовые строки с информацией о событиях, а уже обработанную статистику в виде формируемых отчетов.

инхронизация по времени

Каждый маршрутизатор, произведенный компанией Cisco Systems, оснащен внутренними часами. Некоторые модели старшего класса также имеют встроенный системный календарь, питаемый от батареи. Если маршрутизатор не имеет календаря, то его часы начинают отсчет от следующей даты: 1 марта 1993 00:00. На работающем маршрутизаторе доступно несколько способов установки системных часов. Например, они могут быть установлены вручную с помощью команды, выполняемой в привилегированном режиме, или автоматически с помощью протокола NTP (Network Time Protocol) или SNTP (Simple NTP) в зависимости от модели маршрутизатора. Для проверки системного времени предназначена команда show clock, выполняемая в пользовательском режиме. Приведем пример начальной установки времени на маршрутизаторе Cisco 805. На нем нет встроенного системного календаря, поэтому первоначально при его загрузке была установлена приведенная выше дата. Символ * указывает на то, что операционная система не рассматривает установленное время как соответствующее действительности.

```
Cisco805>show clock
*02:36:23.743 UTC Mon Mar 1 1993
```

Если в распределенной сети существует необходимость поддерживать строгую синхронизацию времени на всех маршрутизаторах, то администратор может настроить протоколы NTP/SNTP на синхронизацию показаний часов с внешним источником. Процедура настройки этих протоколов выходит за рамки материала данной книги. Подробную информацию можно получить на сайте компании Cisco Systems, например, выполнив поиск по ключевым словам Time, NTP, SNTP. После указания внешнего источника времени (обычно это просто IP-адрес соответствующих серверов в Интернете) информацию о синхронизации можно получить при помощи команды show sntp.

```
Cisco805#show sntp
SNTP server     Stratum  Version      Last Receive

192.5.41.209    1        1            1d22h
192.5.41.40     1        1            00:00:01 Synced
```

Файлы конфигурации маршрутизатора

Важной задачей администратора сети помимо настройки маршрутизатора является управление файлами конфигурации. Как уже упоминалось, рабочая конфигурация (running configuration) содержит настройки, которые являются активными в процессе работы маршрутизатора: когда на маршрутизаторе вводятся команды, они помещаются в файл рабочей конфигурации. Так как эта конфигурация содержится в оперативной памяти маршрутизатора, она удаляется при выключении последнего. В противоположность рабочей, загрузочная конфигурация (startup configuration) хранится в памяти NVRAM, и именно она считывается маршрутизатором при его загрузке. После того как маршрутизатор загрузился, все вводимые команды меняют рабочую конфигурацию, поэтому, чтобы исключить несовпадение конфигураций из-за настроек, сделанных после перезагрузки маршрутизатора, обе конфигурации следует привести к одинаковому виду. Загрузочной конфигурации может и не быть (например, на новых маршрутизаторах). В этом случае при первом включении маршрутизатор перейдет в диалоговый режим настройки (system configuration dialog). Все действия по управлению файлами конфигурации выполняются в привилегированном режиме. При этом они не отличаются от обычных операций над файлами в любой другой операционной системе — это просмотр, обновление, создание резервных копий, замена и т. п.

Выше были рассмотрены несколько вариантов использования команды show, которая показывает те или иные настройки маршрутизатора. Но в некоторых ситуациях наиболее простым способом выяснить текущую настройку маршрутизатора является просмотр файлов конфигурации. Команда show доступна в пользовательском режиме, но из-за того что конфигурационные файлы содержат много важной информации, например пароли и настройки протокола SNMP (Simple Network Management Protocol), просмотр файлов конфигурации, как

уже отмечено, возможен только в привилегированном режиме. Для просмотра рабочей конфигурации следует выполнить команду show running-config.

```
Cisco805#show running-config
Building configuration...

Current configuration:
!
! Last configuration change at 16:58:51 GMT Thu Feb 8 2001
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname Cisco805
!
enable secret 5 $1$mlwA$zZERJF6HloTce3LgEfy.J.
...
```

В первой строке результата выводится сообщение Building configuration..., которое говорит о том, что операционная система занята сбором информации о своих настройках и подготовкой ее для представления в удобном виде. Для облегчения восприятия секции, на которые поделены выводимые данные, разделяются символом !. Листинг содержит всю информацию, необходимую для понимания будущего поведения маршрутизатора. Например, видно, что установлено имя маршрутизатора Cisco805, которое затем «отражается» как часть приглашения. Далее выводится пароль доступа в привилегированный режим (enable secret password), который устанавливается при начальной настройке маршрутизатора. Для снижения риска неавторизованного доступа пароль представляется в зашифрованном виде.

Загрузочная конфигурация является текстовым файлом, хранимым в памяти NVRAM. В процессе загрузки маршрутизатора операционная система просматривает память NVRAM на предмет наличия этого файла, а затем копирует его в оперативную память. Для того чтобы просмотреть загрузочную конфигурацию маршрутизатора, нужно выполнить в привилегированном режиме команду show startup-config.

```
Cisco1005#show startup-config
Using 879 out of 7506 bytes
!
version 11.2
service timestamps debug uptime
service timestamps log uptime
service password-encryption
...
```

Так как загрузочная конфигурация хранится в виде текстового файла, готового для отображения на экране, то при выполнении данной команды нет за-

держки, которая присуща команде show running-config. Первая строка результата работы команды show startup-config содержит размер текстового файла и объем памяти NVRAM (это значение можно проверить с помощью команды show version). Если начальная настройка на новом маршрутизаторе еще не выполнялась, то память NVRAM будет пустой. В таком случае в качестве реакции на команду show startup-config администратор получит от маршрутизатора следующее сообщение:

```
%% Non-volatile configuration memory has not been set up or has bad checksum.
```

Изменения, которые администратор сети вносит в настройку маршрутизатора, фиксируются в рабочей конфигурации. Если после внесения этих изменений перезагрузить маршрутизатор, то в его память будет считана загрузочная конфигурация, что приведет к утрате всех выполненных администратором настроек. Для того чтобы не допустить потери настроек, следует скопировать рабочую конфигурацию в загрузочную с помощью команды copy running-config startup-config. В приведенном ниже примере сначала изменяется имя маршрутизатора (в рабочей конфигурации), а затем текущая рабочая конфигурация записывается как загрузочная. Это позволяет не допустить потери изменений в имени маршрутизатора.

```
Cisco1005(config)#hostname Cisco1005_Nevsk
Cisco1005_Nevsk#copy running-config startup-config
Building configuration...
[OK]
```

Существует только один способ заменить рабочую конфигурацию полностью, который заключается в перезагрузке маршрутизатора. Однако администратор может обновлять рабочую конфигурацию маршрутизатора вводом новых команд. Как уже не раз говорилось, эти команды вводятся как простой текст, причем для их ввода существует несколько источников, помимо использования консольного порта: терминальные порты, память NVRAM, сервер TFTP, флэш-память и сервер rcp. Хотя источником назван сервер TFTP, реально при передаче команд работает одноименный протокол TFTP (Trivial File Transfer Protocol), позволяющий передавать файлы от сервера к клиенту без какой-либо авторизации. В основе этого протокола лежит протокол UDP, который не обеспечивает надежности ни в доставке данных, ни в средствах управления потоком трафика. Чтобы воспользоваться механизмом обновления конфигурации маршрутизатора, необходимо установить в сети сервер, поддерживающий этот протокол, а маршрутизатор будет играть роль клиента, посылающего серверу запросы на файлы.

Использование сервера rcp также позволяет осуществлять передачу файлов от сервера к клиенту, однако его отличие от протокола TFTP заключается в том, что в данном случае в качестве транспортного протокола работает протокол TCP, обеспечивающий надежную доставку. Кроме того, при передаче требуется осуществлять аутентификацию, что повышает безопасность. Использование протокола TCP позволяет применять данный метод обновления конфигурации маршрутизатора через сети, в которых существует вероятность возникновения перегрузок и, как следствие, удаления пакетов. Например, это может наблюдаться в сети Frame Relay при ее существенной загрузке.

Для ввода команд, воздействующих на конфигурацию маршрутизатора, следует перейти в глобальный режим посредством команды configure terminal, которая уже рассматривалась ранее в деталях. В дополнение к простому вводу команд с клавиатуры администратор сети может воспользоваться возможностями буфера обмена. Если существует текстовый файл с командами конфигурации, сохраненный на локальном диске компьютера, с которого осуществляется управление маршрутизатором, то можно просто передать его на маршрутизатор с помощью программы эмуляции терминала. Команды будут выполняться в порядке их расположения в файле.

Как уже упоминалось, память NVRAM содержит загрузочный файл конфигурации, представляющий собой текстовый файл с командами, формирующими рабочую конфигурацию. Администратор сети имеет право с помощью команды copy startup-config running-config обновить рабочую конфигурацию маршрутизатора. Данная команда была документирована в версии 11.0 операционной системы Cisco IOS, хотя и была доступна в более ранних версиях. Оригинальная команда для обновления рабочей конфигурации маршрутизатора из памяти NVRAM — configure memory.

На сервере TFTP могут находиться текстовые файлы, команды маршрутизатора или резервная копия рабочей конфигурации. Для обновления рабочей конфигурации с помощью протокола TFTP предусмотрена команда copy tftp running-config, которая предписывает маршрутизатору брать строки (команды) из текстового файла на сервере TFTP и помещать их в рабочую конфигурацию в порядке чтения.

```
Router#copy tftp running-config
Host or network configuration file [host]? host
IP address of remote host [255.255.255.255]? 192.168.3.20
Name of configuration file [Router-config]? commands.txt
Configure using commands.txt from 192.168.3.20? [confirm] y
Loading commands.txt from 192.168.3.20 (via Ethernet0): !!!!!
[OK - 2035/32723 bytes]
Router#
```

В процессе пошагового выполнения данной команды, как видно из примера, маршрутизатор запрашивает тип конфигурационного файла (host or network configuration file). Конфигурационный файл называется host, если он содержит команды, которые специфичны для определенного маршрутизатора. В том случае, если файл включает в себя команды, общие для множества маршрутизаторов, он называется network. Далее по умолчанию предполагается, что имя файла состоит из имени маршрутизатора и следующего за ним слова -config. Затем маршрутизатор запрашивает адрес IP-сервера, на котором работает программное обеспечение, поддерживающее функции сервера TFTP. Вместо указания адреса администратор может ввести имя устройства. Тогда маршрутизатор воспользуется механизмом разрешения имен для получения адреса из имени. После подтверждения обновления маршрутизатор загружает указанный файл с сервера, отмечая успешно скопированные блоки символами !. Для того чтобы избежать ошибок в конце копирования, необходимо в конец текстового файла с команда-

ми поместить команду end, которая необходима для выхода из конфигурационного режима при окончании копирования. Команда copy tftp running-config была документирована в версии 11.0 операционной системы, а оригинальная команда обновления рабочей конфигурации с сервера TFTP носит имя configure network.

Для того чтобы обновить рабочую конфигурацию при помощи сервера rcp, администратор сети может воспользоваться командой copy rcp running-config, которая указывает маршрутизатору считывать строки текстового файла на сервере и помещать их в рабочую конфигурацию, так, как если бы они вводились вручную. Аналогично случаю с сервером TFTP, последней командой в текстовом файле должна быть команда end. Диалог, который ведет маршрутизатор с администратором сети, подобен диалогу в случае с сервером TFTP. Но поскольку протокол rcp требует проведения аутентификации для выполнения копирования, потребуется еще несколько шагов, чтобы маршрутизатор получил доступ к файлам на сервере. Так, на маршрутизаторе необходимо задать имя пользователя, которое будет указываться при доступе, с помощью команды ip rcmd remote-username. Далее на сервере rpc нужно создать пользователя с аналогичным именем и задать ему домашний каталог на сервере, в котором должны размещаться текстовые файлы конфигурации. После этого на сервере в домашнем каталоге пользователя следует создать файл (.rhost), содержащий строки для каждого устройства и учетной записи, которым разрешается доступ к этому каталогу. Но, вообще говоря, рекомендуется обратиться к справочной системе программного обеспечения сервера rcp, так как рассмотрение деталей его настройки выходит за рамки книги.

Текстовые файлы, которые были сохранены во флэш-памяти, также подходят для обновления рабочей конфигурации. При этом источником может являться внутренняя флэш-память или память, организованная на карте PCMCIA, вставленной в одноименный слот маршрутизатора. При копировании текстового файла следует указать в команде копирования устройство и имя файла. Формат команды копирования для маршрутизаторов серии Cisco 1600 и Cisco 3600 следующий:

copy device:<номер раздела:> <имя файла> running-config.

Ниже приведен пример копирования файла с именем ios-upgrade-1 (filename) из четвертого раздела (partition-number) флэш-памяти в рабочую конфигурацию маршрутизатора серии Cisco 1600.

```
Router#copy flash:4:ios-upgrade-1 running-config
Copy 'ios-upgrade-1' from flash device
  as 'startup-config' ? [yes/no] yes
[OK]
```

Администратор сети должен рассмотреть вопрос о создании резервных копий конфигурационных файлов маршрутизатора. При этом в его распоряжении имеется несколько получателей, которым можно направить резервную копию. Получатели соответствуют источникам, которые были рассмотрены выше, за исключением локального диска. При создании резервной копии маршрутизатор строит конфигурацию и форматирует ее как обычный текстовый файл, который затем копируется указанному получателю. При этом все операции аналогичны операциям, связанным с обновлением конфигурации маршрутизатора.

Для того чтобы создать резервную копию рабочей конфигурации в памяти NVRAM, следует воспользоваться командой copy running-config startup-config, выполнение которой приводит к замене загрузочной конфигурации маршрутизатора его рабочей конфигурацией. Данная команда должна выполняться каждый раз после изменения настроек маршрутизатора, чтобы при следующей его загрузке сделанные изменения остались в силе. Эта команда была впервые документирована в версии 11.0 операционной системы Cisco IOS. Оригинальная команда для копирования рабочей конфигурации в память NVRAM — write memory.

Для того чтобы скопировать рабочую конфигурацию в указанный файл на сервер TFTP, в распоряжении администратора есть команда copy running-config tftp. Ниже приведен пример работы команды. Эта команда была впервые документирована в версии 11.0 операционной системы Cisco IOS. Оригинальная команда для создания резервной копии рабочей конфигурации на сервере TFTP — команда write network.

```
Router#copy running-config tftp
Remote host [192.168.3.20]?
Name of configuration file to write [router-confg]?
Write file router-confg on host 192.168.3.20? [confirm] y
#
Writing router-confg!!! [OK]
```

В качестве сервера TFTP может служить программное обеспечение компании Cisco Systems, которое доступно через Интернет на странице загрузки http://www.cisco.com/pcgi-bin/tablebuild.pl/tftp. Программа практически не требует никакой настройки и может быть запущена сразу после установки. Описание процедуры установки также можно найти по приведенному адресу. В процессе передачи данных между сервером и маршрутизатором в окне программы выводится информация о процессе передачи и итоговое заключение об успешности результата (рис. 1.10)

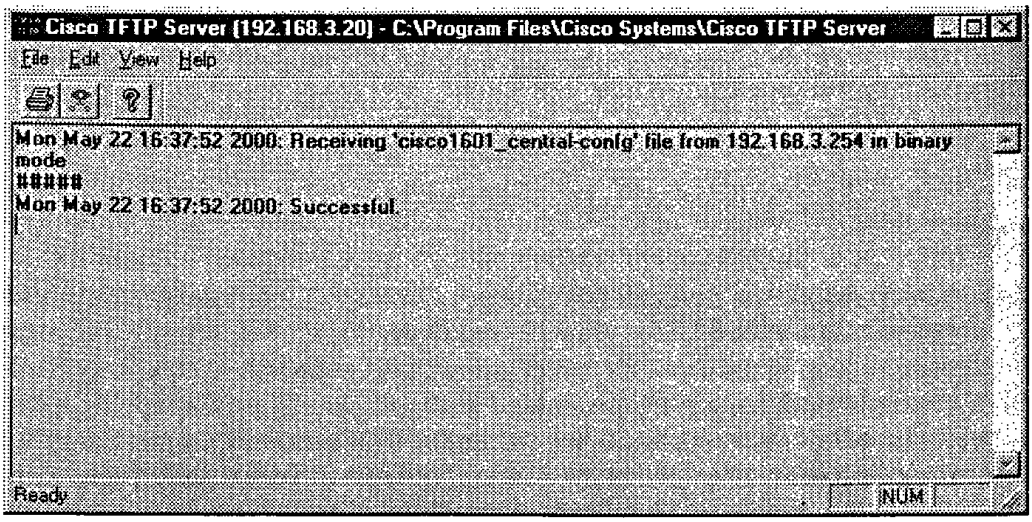


Рис. 1.10. Отображение работы протокола TFTP на сервере

Создание резервной копии на сервере rcp выполняется по аналогии с предыдущим примером, за исключением того, что следует пользоваться командой copy running-config rcp, создающей текстовый файл на указанном сервере rcp. Чтобы создать копию рабочей конфигурации во флэш-памяти, для маршрутизаторов серии Cisco 1600 администратор может прибегнуть к команде copy running-config device:<номер раздела:> <имя файла>. Следует отметить, что маршрутизатор не в состоянии записать информацию во флэш-память, если последняя имеет статус «только для чтения» (read-only). Просмотр статуса памяти производится командами show version или show flash.

```
Cisco1005#show flash

PCMCIA flash directory:
File  Length   Name/status
  1   1589513  c1005-y-mz.113-7.bin
[1589580 bytes used. 507572 available. 2097152 total]
2048K bytes of processor board PCMCIA flash (Read/Write)

Cisco1601#show flash

PCMCIA flash directory:
File  Length   Name/status
  1   4061104  c1600-y-l.113-3.T.bin
[4061168 bytes used. 2230288 available. 6291456 total]
6144K bytes of processor board PCMCIA flash (Read ONLY)
```

Команды для замены загрузочной конфигурации аналогичны тем, которые используются для рабочей конфигурации. При этом необходимо заменить ключевое слово running-config на startup-config. Однако команды, которые работают с загрузочной конфигурацией, имеют одно отличие, состоящее в том, что они заменяют указанным файлом содержимое памяти NVRAM, в то время как команды, воздействующие на рабочую конфигурацию, заменяют команды в активной конфигурации маршрутизатора.

Наиболее часто администратору сети приходится заменять загрузочную конфигурацию рабочей при проведении в последней любых изменений. Для этого имеется команда, которая уже рассматривалась подробно ранее, а именно copy running-confug startup-config. Напомним, что эта команда заменяет содержимое памяти NVRAM файлом рабочей конфигурации при условии, что в этой памяти достаточно свободного места. Напомним также, что оригинальная команда операционной системы Cisco IOS для замены загрузочной конфигурации рабочей — write memory.

Чтобы заместить содержимое памяти NVRAM файлом с сервера TFTP, можно воспользоваться командой copy tftp startup-config, выполнение которой аналогично работе с сервером TFTP. Администратору сети перед ее выполнением следует узнать адрес сервера и имя файла для копирования. Операционная система маршрутизатора позволит скопировать в память NVRAM любой файл, даже если он не содержит конфигурационных команд, хотя в последнем случае произойдет сбой при загрузке маршрутизатора. Рассматриваемая команда была впервые документирована в версии 11.0 операционной системы Cisco IOS, а ори-

гинальная команда создания копирования файла в память NVRAM — configure overwrite-network.

Иногда бывает необходимо начать конфигурацию «с нуля». Когда маршрутизатор стартует без загрузочной конфигурации, его операционная система предполагает, что он не был настроен и переходит в программу пошаговой настройки (System Configuration Dialog). То есть, удалив загрузочную конфигурацию маршрутизатора, можно начать заново настраивать маршрутизатор. Чтобы удалить содержимое памяти NVRAM, администратор сети может выполнить команду erase startup-config. Пожалуй, наибольшее применение эта команда находит в лабораторных (тестовых) условиях, когда достаточно часто требуется полностью менять конфигурацию маршрутизатора. Команда была впервые документирована в версии 11.0 операционной системы Cisco IOS. Оригинальная команда для удаления содержимого памяти NVRAM — write erase. Следует отметить, что если маршрутизатор настраивается с помощью программы Cisco ConfigMaker, то при обновлении настроек маршрутизатора его старая конфигурация будет удаляться автоматически.

Имена и заголовки

Вопросы настройки имени маршрутизатора уже рассматривались выше, и были выработаны некоторые общие рекомендации, смысл которых сводился к необходимости следовать правилам именования, указанным в документе RFC 1035. По умолчанию маршрутизатору присваивается имя Router, однако его рекомендуется изменить на другое, уникальное, приемлемое в распределенной сети. Для этого можно воспользоваться командой hostname, выполняемой в глобальном режиме настройки маршрутизатора. Эта команда имеет всего один параметр — новое имя маршрутизатора.

```
Router(config)#hostname NewName
NewName(config)#
```

По умолчанию маршрутизатор будет использовать в приглашении на ввод команд только первые 29 символов своего имени, что позволяет быстро идентифицировать его. Следует отметить, что длинное имя маршрутизатора хотя и является более информативным, но способно привести к определенным проблемам. Это связано с тем, что общая длина приглашения не может превышать 30 символов вместе с ключевым словом, идентифицирующим текущий режим настройки. А операционная система Cisco IOS будет сокращать длину приглашения за счет ключевого слова. В итоге можно увидеть следующее приглашение:

```
NewName(config)#hostname NewNameOfTheRouterCisco05
NewNameOfTheRouterCisco05(co)#
```

Помимо информативного имени маршрутизатора, в распоряжении администратора есть механизм, позволяющий настроить заголовки (или баннеры, banners), которые предоставляют текстовую информацию при подключении к маршрутизатору. Рассмотрим возможные типы заголовков: MOTD (Message of The Day), Login и Exec. Первый выдается сразу при подключении к маршрутизатору через

терминальный порт. Заголовок Login также посылается при подключении к маршрутизатору (если последний настроен), однако он следует сразу же за заголовком MOTD. Заголовок Exec отображается после успешной авторизации пользователя.

Для того чтобы настроить эти заголовки, предусмотрена команда banner с параметрами — названиями заголовков. Команда выполняется в глобальном режиме. После параметра, указывающего тип заголовка, следует собственно текст сообщения, который будет выдаваться маршрутизатором. Текст начинается и заканчивается специальными символами, идентифицирующими начало и конец сообщения и не являющимися его частью. Сообщение может состоять из нескольких строк текста. Чтобы ввести несколько строк, необходимо после указания первого разделительного символа нажать клавишу Enter, а затем вводить текст построчно. Для завершения ввода нужно вновь ввести разделительный символ и нажать Enter.

```
Cisco1601(config)#banner motd %
Enter TEXT message.  End with the character '%'.
This is the banner.
%
Cisco1601(config)#
```

В приведенном примере был настроен заголовок MOTD. Аналогичным образом настраиваются и другие заголовки, которые отображаются в соответствующий момент подключения или авторизации. Если маршрутизатор настраивается программой Cisco ConfigMaker, то можно воспользоваться тем, что она позволяет в диалоговом режиме настроить заголовок MOTD. После этого программа Cisco ConfigMaker добавит необходимые команды (рис. 1.11).

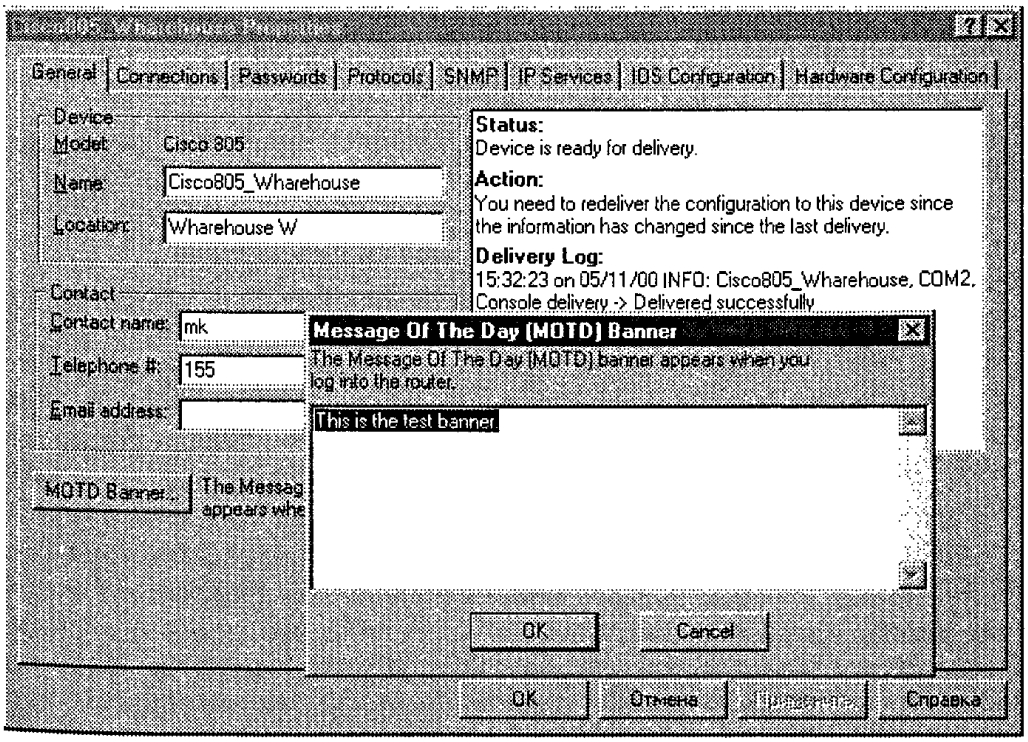


Рис. 1.11. Настройка заголовка с помощью программы ConfigMaker

Обеспечение безопасности доступа к маршрутизатору

Интерфейс командной строки операционной системы Cisco IOS — это основное место, где осуществляется управление и мониторинг маршрутизатора. Без всяких сомнений, администратор сети будет ограничивать доступ к маршрутизатору. И не только для того, чтобы исключить нежелательное влияние на работу маршрутизатора, но и для защиты статистической информации. Для потенциального взломщика статистика и конфигурации, похищенные с маршрутизатора, могут много сказать о сетевой топологии, характере трафика, подключенных системах и т. п. Следовательно, контроль процесса доступа к маршрутизатору — это больше, чем просто защита устройства, это безопасность всей распределенной сети.

ИЗ ОПЫТА

Представьте себе ситуацию, когда доступ в Интернет осуществляется по выделенному каналу связи, который заканчивается маршрутизатором. Причем защита внутренней сети обеспечивается отдельно стоящим защитным экраном. В этом случае маршрутизатор просто выполняет свою главную задачу — передачу и прием трафика Интернета. И если внутренняя сеть защищена, и этот вопрос беспокойства не вызывает, то как быть с маршрутизатором? Что мешает злоумышленникам день за днем пытаться получить доступ к маршрутизатору через Интернет — сначала в пользовательский режим, а затем и в привилегированный? Это может быть простой атакой по словарю, когда некая программа просто перебирает варианты паролей, пытаясь установить сеанс Telnet. В случае, если доступ к маршрутизатору получен, достаточно выполнить несколько команд, например стирающих конфигурации маршрутизатора, или изменить пароли на доступ. В целом безопасности сети это не угрожает, но позволит на некоторое время сделать невозможным доступ в Интернет, что не всегда приемлемо. Для защиты от подобных атак можно, например, воспользоваться списками ограничения доступа.

Операционная система Cisco IOS поддерживает множество типов паролей, которые служат для ее защиты от неавторизованного доступа. Все пароли чувствительны к регистру набираемого текста и не должны содержать более 25 символов, при этом символ пробела не может быть частью пароля.

Доступ в пользовательский режим настраивается на терминальных портах, например на консольном и виртуальном. При настройке пароля для пользовательского режима необходимо перейти в режим конфигурации портов (line configuration mode). Каждый терминальный порт может иметь свой собственный пароль, однако, как правило, для доступа требуется один и тот же пароль. Для того чтобы настроить пароль доступа к терминальному порту маршрутизатора, следует воспользоваться командами login и password. Первая команда указывает маршрутизатору на необходимость выполнения проверки паролей, а вторая непосредственно в своем параметре передает пароль доступа.

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line console 0
Router(config-lin)#password ******
Router(config-lin)#login
```

В этом примере при переходе из глобального режима в режим настройки терминального порта настраивается пароль доступа к консольному порту маршрутизатора (параметр password). Аналогично пароль настраивается на маршрутизаторе и для виртуальных портов VTY. Как правило, для всех портов VTY указывается одинаковый пароль.

СОВЕТ

Если есть вероятность того, что пять активных сеансов Telnet, установленных с маршрутизатором другими людьми, будут препятствовать администратору получить доступ, то можно создать дополнительные порты VTY или, скажем, настроить один пароль для портов с номерами 0–3 и другой пароль для порта 4, который будет задействоваться только в крайних случаях. В последнем варианте команда line vty 4 применима только для настройки порта 4.

При попытке установить с маршрутизатором сеанс Telnet, последний методом случайного выбора выделяет виртуальный порт VTY, вследствие чего невозможно предсказать заранее, к какому порту будет выполнено подключение. В том случае, если все виртуальные порты VTY имеют различные пароли, не исключена ситуация, когда будет неизвестно, какой именно пароль следует указывать.

СОВЕТ

Помимо локальной настройки паролей на маршрутизаторе также применяется внешний сервер авторизации, работающий, например, под управлением программы CiscoSecure ACS. Эта программа, взаимодействуя с маршрутизатором по протоколам TACACS+ или RADIUS, проверяет корректность введенных имен и паролей пользователей. Для решения задачи администратор может также воспользоваться возможностями операционной системы Microsoft Windows 2000, а именно ее компонентом Microsoft IAS (Internet Authentication Service), поддерживающим протокол RADIUS и проводящим авторизацию пользователей, введенных в службу каталогов Active Directory. Такой способ авторизации пользователей очень удобен, например, при создании модемного пула в распределенной сети, где имеются машины под управлением Windows 2000.

Напомним, что, работая в пользовательском режиме, можно только просматривать информацию о работе маршрутизатора, за исключением файлов конфигурации. Полный доступ к маршрутизатору администратор получает лишь при переходе в привилегированный режим. Как уже ранее упоминалось, существуют два пароля, которые предоставляют доступ к привилегированному режиму. Первый из них поддерживается для совместимости со старыми версиями операционной системы Cisco IOS (так называемый enable password). Его особенностью является то, что в файлах конфигурации он хранится в незашифрованном виде. Второй пароль доступа в привилегированный режим (так называемый enable secret password) всегда отображается в зашифрованном виде.

Для настройки этих двух паролей можно обратиться к двум разным командам, которые выполняются в глобальном режиме. Эти команды схожи по названию с паролями, которые они устанавливают: enable password задает незашифрованный пароль, а enable secret — зашифрованный. В обеих командах пароль указывается в виде параметра. Важной особенностью является то, что если указывается зашифрованный пароль, он получает статус приоритетного. Если данный пароль присутствует в рабочей конфигурации маршрутизатора, он и будет основанием для входа в привилегированный режим.

Пароль enable password используется только тогда, когда операционная система не поддерживает более надежный зашифрованный пароль или последний просто не настроен. В случае, если ни один из перечисленных паролей не был задан, после выполнения команды **enable** маршрутизатор не будет предлагать ввести пароль для перехода в привилегированный режим. Сказанное относится к случаю, когда доступ к маршрутизатору осуществляется через консольный порт. Если же работа с маршрутизатором производится через сеанс Telnet, то перейти в привилегированный режим будет невозможно до тех пор, пока не будет присвоен пароль консольному порту.

Очевидно, что два пароля доступа в привилегированный режим следует сделать различными, так как не имеет смысла отображать пароль в зашифрованном виде, если рядом будет существовать незашифрованная версия. При попытке сделать пароли одинаковыми операционная система выдаст предупреждающее сообщение. Но несмотря на предупреждение, одинаковые пароли будут приняты, что можно проверить путем просмотра рабочей конфигурации с помощью команды **show running-config**.

```
Router(config)#enable secret ******
Router(config)#enable password ******
The enable password you have chosen is the same as your enable secret.
This is not recommended. Re-enter the enable password.
Router(config)#
```

Несмотря на то что операционная система Cisco IOS позволяет шифровать все указанные пароли, по умолчанию шифруется только enable secret password. При этом используется однонаправленный (one-way) алгоритм шифрования, что означает невозможность из зашифрованного пароля получить его текстовую версию (выполнить обратное преобразование). Для того чтобы все остальные задаваемые пароли хранились в файлах конфигурации в зашифрованном виде, необходимо выполнить команду, которая указывает операционной системе на необходимость шифрования паролей. Для этого служит команда **service password-encryption**, выполняемая в глобальном режиме. Для шифрования же остальных паролей используется двунаправленный алгоритм (two-way), который позволяет из зашифрованной версии получить исходный текст. Если выполнить команду **show running-config**, можно узнать, какой именно алгоритм шифрования (обозначаемый числом) был применен для тех или иных паролей.

```
Cisco805#show running-config
Building configuration...
...
service password-encryption
...
enable secret 5 $1$mlwO$zZERHd8HloTce7Lcjfy.J.
...
line vty 0 4
 password 7 03665D070F0956424C191G0B1E1R
 login
...
```

Числа, обозначающие алгоритм шифрования, имеют следующий смысл:

- 7 — пароли зашифрованы с помощью двунаправленного алгоритма;
- 5 — пароли зашифрованы с помощью однонаправленного алгоритма;
- 0 — пароли будут отображаться в незашифрованном виде.

Дальнейшая настройка маршрутизатора проводится с активным использованием адресов протокола IP. По этой причине необходимо уделить внимание теории протокола IP (ему посвящена глава 2), а затем продолжить рассмотрение настройки маршрутизатора (в главе 3).

2 Настройка протокола IP

Протокол IP является базовым в корпоративных сетях. На нем, как на фундаменте, выстраивают свою работу практически все остальные сопутствующие технологии. Незнание этого протокола делает бессмысленными все дальнейшие попытки получения знаний в данной области. По означенной причине автор счел необходимым включить в книгу материал по протоколу IP, сделав акцент на практических методах его использования. Кроме того, логика изложения материала книги требует рассмотрения протокола IP именно в этой главе. Для четкого уяснения положения маршрутизаторов в распределенной сети, понимания технологии трансляции адресов, принципов работы протоколов маршрутизации и многих других технологий необходимо твердое знание механизма распределения адресного пространства в протоколе IP.

Протокол IP

Очень кратко рассмотрим основы протокола. Его описание дано в документе RFC 791. IP (Internet Protocol) является базовым протоколом всего стека TCP/IP — он отвечает за передачу информации по сети. Информация передается блоками, которые называются *дейтаграммами*. IP является протоколом сетевого уровня, при этом для каждой среды передачи данных, например Ethernet и ATM, определен свой способ инкапсуляции IP-дейтаграмм. Маршрутизаторы пересылают инкапсулированные дейтаграммы по различным сетям, образуя объединение IP-сетей, в пределах которого каждая рабочая станция может поддерживать связь по протоколу IP с любой другой рабочей станцией.

Услуги, предлагаемые протоколом IP, сводятся к негарантированной доставке дейтаграмм. Протокол IP не исключает потерь и дублирования дейтаграмм, доставки дейтаграмм с ошибками, а также нарушения порядка следования дейтаграмм, заданного при их отправлении.

Протокол IP выполняет фрагментацию и сборку дейтаграмм, если принятый размер кадров в рабочей сети (или участке распределенной сети) отличается от размера исходных дейтаграмм. В протоколе IP отсутствуют механизмы повышения достоверности передачи данных, управления протоколом и синхронизации,

которые обычно предоставляются в протоколах более высокого уровня. Протокол IP получает информацию для передачи от протоколов, расположенных по сравнению с ним на более высоком уровне. К этим протоколам, прежде всего, относятся протоколы TCP и UDP. После получения информации протокол IP передает дейтаграммы через распределенную сеть, используя сервисы локальных сетей. Дейтаграмма состоит из заголовка и поля данных, которое следует сразу за заголовком. Длина поля данных определяется полем «Общая длина» в заголовке. На рис. 2.1 показан формат заголовка IP-дейтаграммы.

<table>
<tr><td>Номер версии (4 бита)</td><td>Длина заголовка (4 бита)</td><td>Тип сервиса (8 битов)</td><td>Общая длина (16 битов)</td></tr>
<tr><td colspan="2">Идентификатор (16 битов)</td><td>Флаги (3 бита)</td><td>Смещение фрагмента (13 битов)</td></tr>
<tr><td colspan="2">Время жизни (8 битов)</td><td>Протокол (8 битов)</td><td>Контрольная сумма заголовка (16 битов)</td></tr>
<tr><td colspan="4">Адрес отправителя (32 бита)</td></tr>
<tr><td colspan="4">Адрес получателя (32 бита)</td></tr>
<tr><td colspan="2">Опции (поле переменной длины)</td><td colspan="2">Выравнивание до 32-разрядной границы</td></tr>
</table>

Рис. 2.1. Формат заголовка дейтаграммы протокола IP

- **Поле «Номер версии» (Version)** указывает на версию используемого протокола IP. В настоящее время распространена версия 4, но планируется переход к версии 6.
- **Поле «Длина заголовка» (Header Lenght)** определяет длину заголовка в 32-разрядных словах. Минимальный размер заголовка — 5 слов (20 байт). Следует отметить, что при увеличении объема служебной информации эта длина может быть увеличена за счет поля «Опции».
- **Поле «Тип сервиса» (Type of Service)** определяет способ обслуживания дейтаграммы. Протокол IP обрабатывает каждую дейтаграмму независимо от ее принадлежности к тому или иному пакету. При этом используются четыре основных механизма: установка типа сервиса, установка времени жизни, установка опций и вычисление контрольной суммы заголовка. Типом сервиса характеризуется набор услуг, которые требуются от маршрутизаторов в распределенной сети. Эти параметры должны использоваться для управления выбором реальных рабочих характеристик при передаче дейтаграмм. В некоторых случаях передача дейтаграммы осуществляется с установкой приоритета, который дает данной дейтаграмме некоторые преимущества при обработке по сравнению с остальными.
- **Поле «Время жизни» (Time to live).** При определенных условиях IP-дейтаграммы могут попасть в замкнутый логический контур (петлю), образованный некоторой группой маршрутизаторов. Иногда такие логические контуры существуют в течение короткого промежутка времени, но иногда они оказываются достаточно долговечными. Чтобы избавить сеть от дейтаграмм, циркулирующих в логических контурах слишком долго, протоколом IP устанавливается предельный срок пребывания дейтаграммы в сети. Он задается в поле

«Время жизни» (TTL — Time to Live). Его содержимое уменьшается на единицу при прохождении дейтаграммы через маршрутизатор; при обнулении поля TTL дейтаграмма отбрасывается.

Первоначально спецификации IP включали еще одно требование: поле TTL должно уменьшаться, по крайней мере, один раз в секунду. Поскольку поле TTL является 8-разрядным, это означает, что теоретически дейтаграмма может находиться в сети не более 4,25 минуты. На практике требование ежесекундного уменьшения поля TTL игнорируется, тем не менее в спецификациях многих протоколов следующих уровней (TCP) по-прежнему предполагается, что максимальное время жизни дейтаграммы в сети составляет лишь несколько минут.

- **Поле «Идентификатор» (Identification)** используется для распознавания дейтаграмм, образованных в результате фрагментации. Все части фрагментированной дейтаграммы должны иметь одинаковое значение этого поля.
- **Поле «Общая длина» (Total Length)** указывает общую длину дейтаграммы (заголовок и поле данных). Максимальный размер дейтаграммы может составлять 65 535 байт. В подавляющем большинстве сетей такой размер дейтаграмм не используется. По стандарту RFC 791 все устройства в сети должны быть готовы принимать дейтаграммы длиной 576 байт. Эти ограничения необходимы для передачи дейтаграмм в физических кадрах. Передача дейтаграммы в кадре называется *инкапсуляцией.* С точки зрения низших уровней дейтаграмма выглядит так же, как и любое другое сообщение в сети. Сетевое оборудование не работает с дейтаграммами, поэтому дейтаграмма является частью области данных кадра.

  Функции фрагментации и сборки также возложены на протокол IP. Фрагментация — это разделение большой дейтаграммы на несколько отдельных частей. В большинстве локальных и глобальных сетей есть ограничения на максимальный размер единицы передаваемой информации. Эту величину называют максимальной единицей передачи (MTU — Maximum Transfer Unit). Например, в сетях Ethernet данная величина составляет 1500 байт, а в сетях FDDI — 4096 байт.

  Когда маршрутизатор переправляет дейтаграмму из одной сети в другую, может оказаться, что размер дейтаграммы окажется недопустимым в новой сети. Спецификация IP предусматривает следующее решение этой проблемы: маршрутизатор может разбить дейтаграмму на мелкие фрагменты, приемлемые для выходной среды, а в пункте назначения эти фрагменты будут объединены в дейтаграмму исходного вида. Формируемые маршрутизатором фрагменты идентифицируются смещением относительно начала исходной дейтаграммы. Дейтаграмма идентифицируется по отправителю, пункту назначения, типу протокола высокого уровня и 16-разрядному полю «Идентификатор». Все это в совокупности должно образовывать уникальную комбинацию.

  Следует подчеркнуть связь между полями «Время жизни» и «Идентификатор». Действительно, во избежание смешивания фрагментов двух разных дейтаграмм отправитель IP-данных обязан исключить ситуацию, когда в один пункт на-

значения по одному и тому же протоколу в течение жизненного цикла дейтаграммы будут отправлены две дейтаграммы с совпадающими идентификаторами. В связи с тем, что идентификатор 16-разрядный, а наибольшее время жизни дейтаграммы исчисляется минутами (будем считать, что оно порядка 2 минут), получаем скорость передачи — 546 дейтаграмм в секунду. При максимальном размере дейтаграммы, равном 64 Кбайт, имеем результирующую скорость около 300 Мбит/с.

Проблема эффективного использования битов идентификатора оказалась практически решенной с появлением метода MTU Discovery, позволяющего определить значения MTU на всем пути к пункту назначения.

Фрагментация и сборка производятся автоматически и не требуют от отправителя специальных действий. Каждая фрагментированная часть имеет тот же формат, что и исходная дейтаграмма. Факт фрагментации повышает вероятность потери исходной дейтаграммы, так как утрата даже одного фрагмента приводит к потере всей дейтаграммы. Сборка дейтаграммы осуществляется на месте назначения. Такой метод позволяет маршрутизировать фрагменты независимо.

- **Поле «Флаги» (Flags)** используется при фрагментации. Нулевое значение первого бита разрешает фрагментацию, а единичное — запрещает. Единичный второй бит указывает на последний фрагмент дейтаграммы.
- **Поле «Смещение фрагмента»** служит для указания смещения данных во фрагменте относительно начала исходной дейтаграммы. Чтобы получить смещение в байтах, надо умножить значение этого поля на 8. Первый фрагмент всегда имеет нулевое смещение. Поле задействуется при сборке фрагментов дейтаграммы после передачи по сетям с различными MTU.

ПРИМЕЧАНИЕ

Здесь стоит упомянуть об атаке TearDrop Fragmentation. Эта атака относится к типу DoS (глава 5) и направлена на выведение компьютера из строя (общий сбой системы или потерю взаимодействия с другими компьютерами) за счет нестандартной фрагментации дейтаграмм протокола IP. Существуют несколько вариантов этой атаки, такие как NewTear, Nestea, SynDrop и Bonk. Так, есть вариант, при котором атакующий посылает специально сформированную пару фрагментов дейтаграмм IP, которые при сборке на атакуемой системе формируют некорректный пакет протокола UDP. Причем второй фрагмент при сборке перезаписывает данные в середине заголовка UDP, содержащегося в первом фрагменте, таким образом, что результирующая дейтаграмма получается как бы незавершенной, что может, например, вызвать останов операционной системы Windows NT с исключением STOP 0x0000000A или 0x00000019. Согласно данным компании ISS (Internet Security Systems), уязвимыми для этой атаки являются такие системы, как Windows NT, Windows 95, Linux. Компания Microsoft выпустила исправление, информацию о котором можно найти по следующему адресу (статья Q179129 «STOP 0x0000000A or 0x00000019 Due to Modified Teardrop Attack»): http://support.microsoft.com/support/kb/articles/q179/1/29.asp.

- **Поле «Протокол» (Protocol)** идентифицирует протокол верхнего уровня, которому принадлежит дейтаграмма. При поступлении дейтаграммы это поле указывает, какому приложению следует ее передать. Таблица 2.1 содержит перечень (неполный) возможных протоколов.

Таблица 2.1. Значения поля «Протокол»

Значение поля	Протокол	Пояснение
0	Зарезервировано	
1	ICMP	Internet Control Message Protocol, протокол управляющих сообщений
2	IGMP	Internet Group Management Protocol, протокол управления группами
4	IP	Инкапсуляция IP в IP
6	TCP	Transmission Control Protocol, протокол управления передачей
8	EGP	Exterior Gateway Protocol, протокол внешнего шлюза
17	UDP	User Datagram Protocol, протокол пользовательских дейтаграмм
88	IGRP	Interior Gateway Routing Protocol, внутренний протокол маршрутизации
89	OSPF	Open Shortest Path First, «первый кратчайший путь»

- **Поле «Контрольная сумма»** рассчитывается по всему заголовку. Так как некоторые поля заголовка меняют свое значение, например «Время жизни», при прохождении дейтаграммы через маршрутизаторы контрольная сумма проверяется и повторно рассчитывается при каждой модификации заголовка. Определение контрольной суммы заголовка обеспечивает безошибочность передачи дейтаграммы через сеть. Перед отправкой дейтаграммы вычисляется контрольная сумма, которая вносится в ее заголовок. При получении дейтаграммы вычисляется ее контрольная сумма, которая сравнивается с соответствующим значением заголовке. При несовпадении дейтаграмма отбрасывается. Контрольная сумма заголовка дейтаграммы применяется и во многих других протоколах, таких как UDP, TCP, ICMP и OSPF.
- **Поля «Адрес отправителя» и «Адрес получателя» (Source Address, Destination Address)** имеют одинаковую длину и структуру. Поля содержат 32-разрядные IP-адреса отправителя и получателя дейтаграммы.
- **Поле «Опции» (Options)** не обязательно и обычно используется при настройке сети. В этом поле может быть указан точный маршрут прохождения дейтаграммы в распределенной сети, данные о безопасности, различные временные отметки и т. д. Поле не имеет фиксированной длины, поэтому для выравнивания заголовка дейтаграммы по 32-разрядной границе предусмотрено следующее поле — поле **«Выравнивание» (Padding)**. При выравнивании поле заполняется нулями.

Классы адресов и их маски

Межсетевая схема адресации протокола IP описана в документах RFC 990 и RFC 997. При разработке протоколов стека TCP/IP рассматривался целый ряд методов идентификации конечных устройств в сети. Окончательным стало решение о присвоении адреса как сети, так и устройствам в этой сети. Основными

доводами в пользу такого подхода явились: возможность задания номеров сетей и устройств в них в широком диапазоне значений и возможность реализации маршрутизации. При этом адреса должны назначаться упорядоченно, для того чтобы сделать маршрутизацию более эффективной.

В сети, построенной на базе протокола TCP/IP, конечные устройства получают уникальные адреса. Эти устройства могут быть персональными компьютерами, коммуникационными серверами, маршрутизаторами и т. д. Некоторые устройства, которые имеют несколько физических интерфейсов, например маршрутизаторы, должны иметь уникальный адрес для каждого из своих интерфейсов. Исходя из схемы адресации и возможности того, что некоторые устройства в сети будут обладать несколькими адресами, напрашивается вывод, что такая схема адресации описывает не само устройство в сети, а определенное соединение этого устройства с сетью. Это приводит к ряду неудобств. Одним из них является необходимость замены адреса устройства при перемещении его в другую сеть. Основной же недостаток в том, что для работы с устройствами, имеющими несколько подключений в распределенной сети, необходимо знать все его адреса, идентифицирующие эти подключения. Незнание хотя бы одного адреса может привести к тому, что эти устройства не получат необходимую информацию при отказе других соединений.

При стандартизации протокола IP в сентябре 1981 года его спецификация требовала, чтобы каждое устройство, подключенное к сети, имело уникальный 32-разрядный адрес. Этот адрес разбивается на две части. Первая часть адреса идентифицирует сеть, в которой располагается устройство. Вторая часть однозначно идентифицирует само устройство. Такая схема создает двухуровневую адресную иерархию (рис. 2.2).

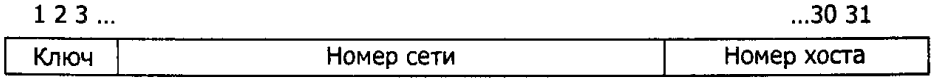


Рис. 2.2. Формат IP-адреса

В последнее время поле номера сети в адресе называется сетевым префиксом. Это связано с тем, что первый квадрант каждого IP-адреса идентифицирует номер сети. Все хосты в определенной сети имеют один и тот же сетевой префикс, но при этом номера хостов обязаны быть уникальны. Аналогично, два любых хоста, расположенные в разных сетях, должны иметь различные сетевые префиксы, но при этом допускаются одинаковые номера хостов.

Для обеспечения гибкости в присвоении адресов компьютерным сетям разработчики определили, что адресное пространство протокола IP должно быть разделено на три основных различных класса — A, B и C. Каждый из этих основных классов фиксирует границу между сетевым префиксом и номером хоста в различных точках 32-разрядного адресного пространства. На рис. 2.3 показаны форматы основных классов.

Одно из основных достоинств использования классов заключается в том, что каждый адрес содержит ключ, который идентифицирует точку, расположенную между сетевым префиксом и номером хоста. Например, если старшие два бита адреса установлены в 1 и 0, то линия раздела пролегает между 15 и 16-м битами.

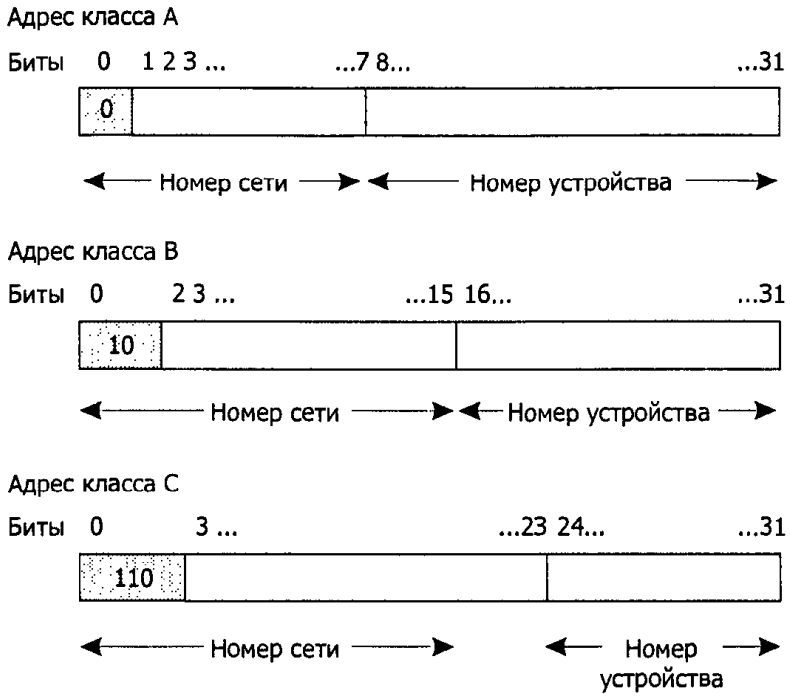


Рис. 2.3. Основные классы IP-адресов

Недостатком этого метода является потребность изменения сетевого адреса, когда в сетях класса C число устройств становится больше 255. В этом случае возникает необходимость замены адресов класса C на адреса класса B. Изменение сетевых адресов может потребовать от администратора сети больших усилий и достаточно много времени для проведения работ по отладке. Ввиду того что существует четкая граница между классами адресов, администраторы сетей не могут заранее спланировать плавный переход изменения адресов. Вместо этого приходится достаточно жестко вмешиваться в работу сети: вводится запрет на использование некоторых сетевых адресов, производится одновременное изменение всех адресов устройств в этой сети. И только тогда сеть вновь включается в работу. Еще одним недостатком классификации адресов является значительное уменьшение числа теоретически возможных индивидуальных адресов. В текущей версии протокола IP (версия 4) общее их число может составлять 2^{32} (4 294 967 296), так как протокол предусматривает только 32 разряда для задания адреса. Использование части битов в служебных целях уменьшает доступное количество индивидуальных адресов.

Адрес класса A предназначен для больших сетей, где количество компьютеров исчисляется тысячами. Каждый адрес класса A имеет 8-разрядный префикс сети, в котором старший бит установлен в 1, а следующие 7 бит задают номер сети. Для указания номера хоста используются оставшиеся 24 бит. В настоящее время все адреса класса A уже выделены. Сети класса A также обозначаются записью /8, так как адреса этого класса имеют 8-разрядный сетевой префикс (7 + 1 = 8).

Максимальное число сетей класса A, которые можно определить, составляет 126 (2^7–2). Каждая сеть этого класса поддерживает максимум 16 777 214 (2^{24} – –2) хостов. Так как адресный блок класса A способен содержать максимум 2^{31}

(2 147 483 648) индивидуальных адресов, а в протоколе IP версии 4 под них отведено максимум 2^{32} (4 294 967 296) адресов, то адресный диапазон класса A занимает 50% предусмотренного адресного пространства.

Адрес класса B используется в сетях среднего размера, например в сетях крупной организации. Каждая сеть класса B имеет 16-разрядный префикс сети, в котором два старших бита установлены в 1 и 0, а следующие 14 бит задают номер сети. Для указания номера хоста используются оставшиеся 16 бит. Сети класса B также обозначаются записью /16.

Максимально доступное число сетей класса B составляет 16 384 (2^{14}). Каждая сеть этого класса поддерживает максимум 65 534 ($2^{16}-2$) хоста. Так как весь адресный блок класса B может содержать максимум 2^{30} (1 073 741 824) индивидуальных адресов, то он занимает 25% предусмотренного адресного пространства.

Адреса класса C используются в сетях с небольшим числом компьютеров. Каждая сеть класса C имеет 24-разрядный префикс сети, в котором три старших бита установлены в 110, а следующие 21 бит используются для задания номера сети. Для задания номера хоста используются оставшиеся 8 бит. Сети класса C также обозначаются записью /24.

Предельно допустимое число сетей класса C составляет 2 097 152 (2^{21}). Каждая сеть этого класса поддерживает максимум 254 (2^8-2) хоста. Так как весь адресный блок класса C может содержать максимум 2^{29} (536 870 912) индивидуальных адресов, он оккупирует 12,5% предусмотренного адресного пространства.

В дополнение к этим трем наиболее популярным классам адресов существуют еще два класса. В классе D старшие четыре бита установлены в 1110. Этот класс используется для поддержки групповой передачи данных. В классе E старшие четыре бита установлены в 1111, и этот класс является зарезервированным для экспериментальных целей.

Для удобства в технической литературе, в прикладных программах и т. д. адреса протокола IP обычно представляются как четыре десятичных числа, разделенные точками. Каждое из этих чисел представляет значение одного из четырех октетов IP-адреса. При этом один октет соответствует 8 бит адреса. Рисунок 2.4 показывает, как адрес класса B может быть представлен в точечно-десятичной нотации.

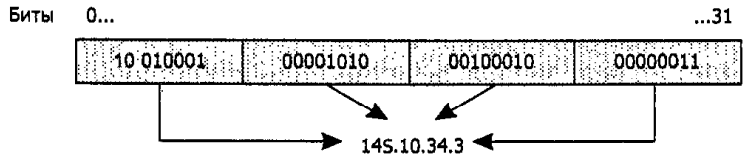


Рис. 2.4. Пример записи IP-адреса в точечно-десятичной нотации

Таблица 2.2 содержит диапазоны десятичных значений трех классов адресов. В этой таблице обозначение XXX представляет собой поле для указания адреса хоста.

Некоторые IP-адреса зарезервированы для определенных целей и не могут присваиваться конечным устройствам в сети. В табл. 2.3 перечислены зарезервированные IP-адреса.

Таблица 2.2. Диапазоны значений адресов трех классов

Класс адреса	Диапазон значений
A	1.XXX.XXX.XXX–126.XXX.XXX.XXX
B	128.0.XXX.XXX–191.255.XXX.XXX
C	192.0.0.XXX–223.255.255.XXX

Таблица 2.3. Зарезервированные IP-адреса

IP-адрес		Примечания
Все биты установлены в 0		Данное устройство
Номер сети	Все биты номера хоста равны 0	Данная IP-сеть
Все биты равны 0	Номер хоста	Устройство в данной IP-сети
Все биты установлены в 1		Все устройства в данной IP-сети
Номер сети	Все биты номера хоста равны 1	Все устройства в указанной сети
127.0.0.1		Адрес обратной связи

В зарезервированных IP-адресах все установленные в ноль биты соответствуют либо конкретному устройству, либо определенной сети. IP-адреса, все биты которых установлены в 1, предназначены для широковещательной передачи информации. Для ссылки на всю IP-сеть в целом используется адрес с номером хоста, у которого все биты установлены в 0. Сетевой адрес класса A 127.0.0.0 зарезервирован для обратной связи и введен для тестирования взаимодействия процессов на одной машине. Когда приложение использует адрес обратной связи, стек TCP/IP возвращает эти данные приложению, ничего не посылая по сети. Следует отметить, что в сетях, построенных на базе протокола IP, запрещается присваивать устройствам IP-адреса, начинающиеся с числа 127.

Все вышесказанное о классах адресов в протоколе IP и их свойствах можно обобщить в одной таблице (табл. 2.4). Обратите внимание на формат записи адреса для соответствующего класса: N обозначает адрес сети, а H — адрес хоста в этой сети. Как уже отмечено, класс D используется для групповой доставки информации, а E — экспериментальный класс.

Помимо возможности направленной передачи информации определенному хосту существует широковещательная передача (broadcasting), при которой сообщения получают все хосты в указанной сети. Положениями стандарта определено, что любой адрес, состоящий из одних единиц, зарезервирован для широковещания. В протоколе IP существуют два типа широковещания: направленное (directed) и ограниченное (limited). Направленное широковещание позволяет хосту удаленной сети передавать одну дейтаграмму, которая будет доставлена всем хостам в адресованной сети. Дейтаграмма с направленным широковещательным адресом может проходить через маршрутизаторы в распределенной сети, при этом исходная дейтаграмма будет доставлена всем хостам только в нужной сети, а не в промежуточных сетях.

Таблица 2.4. Классы адресов и их свойства

Класс адреса	Формат записи	Назначение	Старшие биты	Границы адресов	Количество битов в адресе сети и хоста	Максимальное количество хостов
A	N.H.H.H	Для больших организаций	0	1.0.0.0–126.0.0.0	7/24	16 777 214 (2^{24}–2)
B	N.N.H.H	Организации среднего размера	1, 0	128.1.0.0–191.254.0.0	14/16	65 543 (2^{16}–2)
C	N.N.N.H	Небольшие организации	1, 1, 0	192.0.1.0–223.255.254.0	22/8	254 (2^{8}–2)
D	–	Группы хостов	1, 1, 1, 0	224.0.0.0–239.255.255.255	–	–
E	–	Экспериментальный	1, 1, 1, 1	240.0.0.0–254.255.255.255	–	–

При направленном широковещании адрес получателя содержит корректный номер сети и номер хоста, все биты которого установлены в нули или единицы. Например, адреса 185.100.255.255 и 185.100.0.0 будут рассматриваться как адреса направленного широковещания для сети 185.100.xxx.xxx класса B. Таким образом, направленные широковещательные адреса обеспечивают мощный механизм, позволяющий удаленному устройству посылать одну IP-дейтаграмму, которая будет доставлена в режиме широковещания в указанную сеть. Для получения более подробной информации о направленном широковещании можно обратиться к документу RFC 1812.

СОВЕТ

В рекомендациях компании Cisco Systems по внедрению системы безопасности в распределенной сети предлагается отключать поддержку направленного широковещания на промежуточных маршрутизаторах. В общем случае это отключение не повлияет на работоспособность конечных устройств в сети и на их связь друг с другом. Отключение направленного широковещания может препятствовать попыткам заполнения целевой сети ненужным широковещательным трафиком, который способен помешать нормальной работе. Для отключения этого механизма следует использовать команду no ip directed-broadcast в режиме конфигурирования интерфейсов. Следующие команды отключают направленное широковещание через интерфейс Ethernet0:

```
Router(config)#int eth0
Router(config)#no ip directed-broadcast
```

Главным недостатком направленного широковещания является то, что требуется знание номера целевой сети. Вторая форма широковещания, называемая ограниченной, обеспечивает широковещательную передачу для сети отправителя независимо от указанного IP-адреса. Дейтаграмма с ограниченным широковещательным адресом никогда не сможет пройти через маршрутизаторы — последние не пропустят ее дальше себя в остальные части распределенной сети.

При ограниченном широковещании биты номера сети и номера хоста состоят из одних нулей или единиц. Таким образом, дейтаграмма с адресом получателя 255.255.255.255 или 0.0.0.0 будет рассматриваться как дейтаграмма с ограниченным широковещанием.

Прежде чем использовать в сети (подключенной к Интернету) протоколы стека TCP/IP, необходимо получить один или несколько официальных сетевых адресов. Все адреса присваивает одна организация — Internet Network Information Centre (InterNIC), что обеспечивает их уникальность. До апреля 1993 года назначением IP-адресов занималась организация Network Information Center (NIC). В настоящее время эта организация выполняет запросы только для сетей Defense Data Network (DDN), иначе говоря для военных целей. Организация InterNIC назначает только сетевую часть адреса, или сетевой префикс, оставляя ответственность за определение номеров хостов в этой сети самой организации, запросившей адрес.

Подсети

В 1985 году документом RFC 950 был определен стандартный процесс поддержки формирования подсетей, или разделения единственного номера сети классов A, B и C на составные части. Формирование подсетей было введено для преодоления следующих проблем:

- резкого роста размера и количества таблиц маршрутизации в Интернете;
- появления дефицита номеров сетей при необходимости расширения их количества.

Обе эти проблемы решались путем добавления еще одного уровня иерархии к адресной структуре протокола IP. На рис. 2.5 представлен процесс формирования подсетей, в котором номер хоста делится на две части: номер подсети и номер хоста в этой подсети.

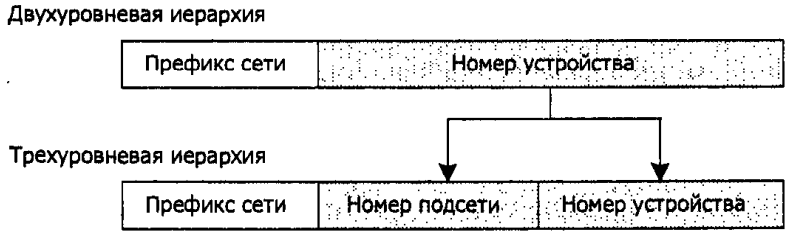


Рис. 2.5. Формирование подсетей

Формирование подсетей решает проблему роста таблиц маршрутизации, так как конфигурация подсетей корпоративной сети никогда не видна за пределами организации. Маршруты из Интернета в любую подсеть данного IP-адреса одинаковы, независимо от того, на какой подсети расположен получатель. Это стало возможным потому, что все подсети данного номера сети используют один и тот же сетевой префикс, но с разными номерами подсетей. Маршрутизаторам в част-

ной сети требуется различать отдельные подсети, а у маршрутизаторов в Интернете все эти подсети определены единственной записью в таблицах маршрутизации. Это позволяет администратору частной сети вносить любые изменения в логическую структуру сети без влияния на размер таблиц маршрутизации у маршрутизаторов в Интернете.

Формирование подсетей также обеспечивает решение второй проблемы, связанной с выделением организации нового сетевого номера или номеров при ее росте. Организации можно выделить один номер сети, после чего администратор получает право произвольно присваивать номера подсетей каждой из своих внутренних сетей. Это позволяет внедрять дополнительные подсети без необходимости получения нового сетевого номера.

На рис. 2.6 показан пример распределенной сети, состоящей из нескольких логических сетей, которые используют концепцию подсетей внутри одного адреса класса B. Граничный маршрутизатор получает весь трафик, адресованный сети 130.5.0.0 из Интернета, и передает его внутренним подсетям, основываясь на информации, содержащейся в третьем октете.

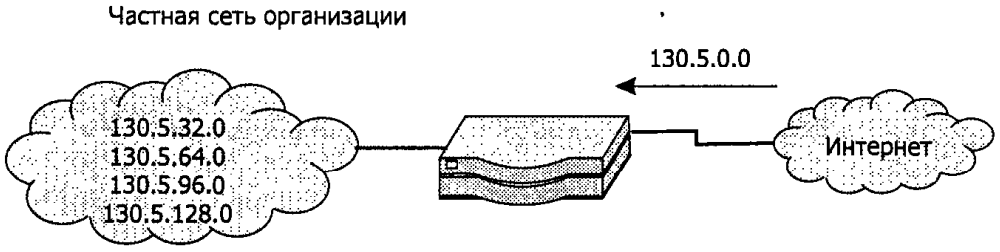


Рис. 2.6. Введение подсетей в организации

Формирование подсетей внутри частной сети организации дает следующие преимущества:

- размер глобальных таблиц маршрутизации в Интернете не растет, так как администратор частной сети не нуждается в получении дополнительной адресной информации;
- администратор получает возможность по своему усмотрению внедрять дополнительные подсети без запроса новых номеров сетей;
- изменение топологии частной сети не влияет на таблицы маршрутизации в Интернете, поскольку маршрутизаторы в Интернете не имеют маршрутов в индивидуальные подсети организации — они могут направить данные только в родительскую сеть.

Маска подсети

Если маршрутизаторам в Интернете достаточно для передачи трафика в окружение подсетей только сетевого префикса адреса получателя, то маршрутизаторы внутри этого окружения для передачи трафика индивидуальным подсетям

используют расширенный сетевой префикс. Расширенный сетевой префикс состоит из префикса сети и номера подсети (рис. 2.7).

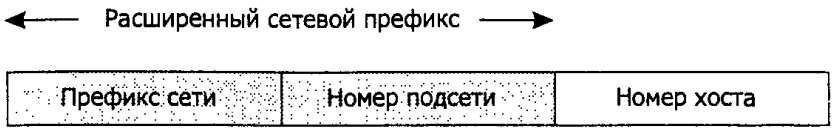


Рис. 2.7. Расширенный сетевой префикс

Структуру расширенного сетевого префикса можно представить в виде маски подсети (subnet mask). Маска подсети — это число, двоичная запись которого содержит единицы в тех разрядах, которые должны интерпретироваться как номер сети. Маска подсети позволяет провести четкую границу между двумя составляющими IP-адреса. Одна часть идентифицирует номер подсети, вторая используется для идентификации хостов в этой подсети.

Хосты и маршрутизаторы используют старшие биты IP-адреса для определения его класса. После того как класс определен, хост может легко найти границу между битами, идентифицирующими номер сети и номер хоста в этой сети. Однако для определения границ битов, указывающих на номер подсети, такая схема не подходит. Для решения этого вопроса используется 32-разрядная маска подсети, которая помогает однозначно определить требуемую границу. Для стандартных классов сетей маски имеют следующие значения:

- 255.0.0.0 — маска для сети класса A;
- 255.255.0.0 — маска для сети класса B;
- 255.255.255.0 — маска для сети класса C.

Например, пусть адрес класса B 130.5.0.0 и сетевой администратор хочет использовать весь третий октет для номера подсети, тогда ему необходимо указать маску подсети 255.255.255.0. Биты в маске подсети задают способ интерпретации битов адреса: если бит в маске подсети равен 1, то система, проверяющая адрес, должна рассматривать соответствующий бит в IP-адресе как часть расширенного сетевого префикса. Другими словами, после определения класса IP-адреса любой бит в части номера хоста, которому соответствует единичный бит в маске подсети, принимает участие в идентификации номера подсети. Оставшаяся часть номера хоста, которой соответствует нулевое значение маски подсети, используется для задания номера хоста. На рис. 2.8 показан пример IP-адреса класса B с соответствующей маской подсети.

		Сетевой префикс		Номер подсети	Номер хоста
IP-адрес	130.5.5.25	10000010	00000101	**00000101**	00011001
Маска подсети	255.255.255.0	11111111	11111111	**11111111**	00000000
		Расширенный сетевой префикс			

Рис. 2.8. Пример использования маски подсети

Все сказанное можно проиллюстрировать рисунком, из которого становится понятным механизм формирования маски подсети — происходит простое преобразование двоичной записи в десятичную (рис. 2.9).

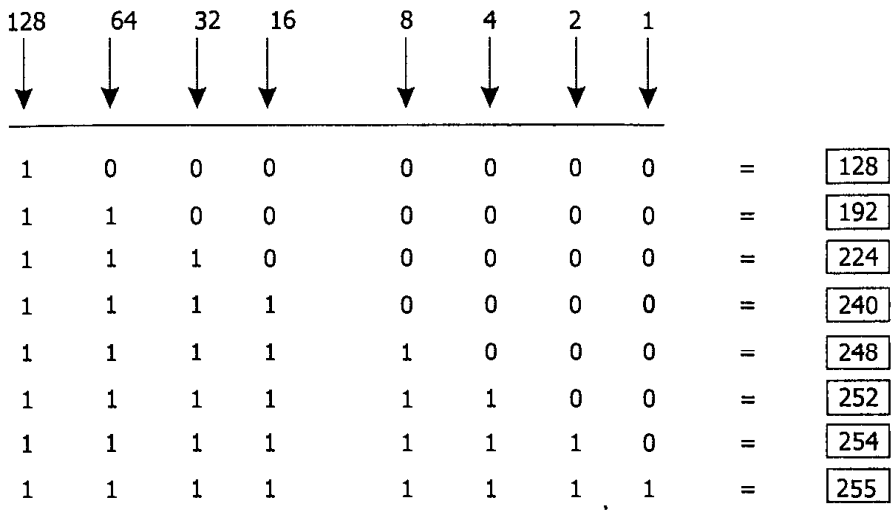


Рис. 2.9. Формирование маски подсети

В стандартах, описывающих современные протоколы маршрутизации, часто делается ссылка на длину расширенного сетевого префикса, а не на маску подсети. Данная длина эквивалентна количеству единичных битов в маске подсети. Это означает, что сетевой адрес 130.5.5.25 с маской подсети 255.255.255.0 может быть также записан как 130.5.5.25/24. Число 24 указывает на то, что в маске подсети 255.255.255.0 количество единичных битов равно 24. Такая запись является более компактной и легкой для понимания, чем запись с оформлением маски подсети в традиционной точечно-десятичной нотации. Рисунок 2.10 иллюстрирует пример записи расширенного сетевого префикса.

	Сетевой префикс		Номер подсети	Номер хоста
130.5.5.25	10000010	00000101	00000101	00011001
255.255.255.0	11111111	11111111	11111111	00000000
Или эквивалентная запись:				
	24-разрядный расширенный сетевой префикс			Номер хоста
130.5.5.25/24	**10000010**	**00000101**	**00000101**	00011001

Рис. 2.10. Пример использования расширенного сетевого префикса

Следует отметить, что большинство современных протоколов маршрутизации переносят маску подсети в своих сообщениях. В то же время не существует

стандартного протокола маршрутизации, который имел бы дополнительное однобайтовое поле в заголовке своих сообщений, определяющее число битов в расширенном сетевом префиксе. Протоколы маршрутизации передают полную четырехоктетную маску подсети.

Перед тем как разрабатывать сеть на базе протокола IP, сетевому администратору необходимо ответить на следующие четыре важных вопроса.

1. Сколько подсетей требуется организации сегодня?
2. Сколько подсетей может потребоваться организации в будущем?
3. Сколько хостов существует в наибольшей подсети организации сегодня?
4. Сколько хостов необходимо будет поддерживать в наибольшей подсети организации в будущем?

Первым шагом в процессе планирования является определение максимального количества требуемых подсетей. Данное значение округляется до ближайшей степени числа 2. Когда выполняется эта оценка, важно учесть будущее увеличение количества подсетей. На втором шаге проверяется факт существования достаточного количества адресов хостов в наибольшей подсети организации. И в заключение следует убедиться в том, что выделенный организации класс адреса предоставляет достаточное количество битов, необходимых для формирования подсетей.

Рассмотрим пример формирования подсетей в организации. Предположим, что организация получила сеть класса С 193.1.1.0 и ей необходимо сформировать шесть подсетей. Наибольшая подсеть должна поддерживать 25 хостов. На первом шаге определяется число битов, требуемых для выделения необходимых шести подсетей. Округляя число 6 до ближайшей степени 2, получаем, что в данном примере администратор должен определить восемь подсетей ($2^3 = 8$), то есть для выделения подсетей будут использованы три бита адреса.

Так как в данном примере выделен адрес класса С с записью расширенного сетевого префикса в виде /24, то полученный после выделения подсетей расширенный сетевой префикс будет выглядеть как /27 (24 + 3 = 27). Этот расширенный сетевой префикс эквивалентен маске подсети 255.255.255.224 (рис. 2.11).

		Сетевой префикс			Байт для задания номеров хостов в данной сети	
		Байты для задания номера сети			Биты для номеров подсетей	Биты для номеров хостов
Адрес	193.1.1.0	11000001	00000001	00000001	000	00000
Маска подсети	255.255.255.0	11111111	11111111	11111111	111	00000
Или эквивалентная запись:						
Адрес	193.1.1.0/27	**11000001.00000001.00000001.000**				00000

Рис. 2.11. Определение маски подсети в организации

Необходимо отметить, что номер подсети не обязательно должен располагаться сразу после сетевого префикса. Администратор вправе устанавливать биты в маске подсети независимо от остальной части адреса. В примере с адресом 193.1.1.0/27 третий байт маски подсети вместо (11100000_2) может быть, например, равен (00011100_2). Однако на практике в большинстве случаев так не поступают.

Используемый 27-разрядный расширенный сетевой префикс оставляет 5 битов для задания номеров хостов в каждой из подсетей. Это означает, что в каждой подсети может быть выделено до 32 ($2^5 = 32$) индивидуальных адресов хостов. Но так как адреса, у которых все биты равны нулю либо единице, являются зарезервированными, то общее число адресов хостов в каждой подсети становится равным 30 ($2^5 - 2 = 30$).

Для определения какой-либо подсети администратор помещает двоичное представление ее номера (в данном примере для восьми подсетей это может быть цифра от 0 до 7) в битовое поле номера подсети. Например, для определения подсети #4 администратор просто заносит двоичное представление числа 4 ($00000\mathbf{100}_2$) в трехбитовое поле номера подсети. Таблица 2.5 представляет все восемь возможных вариантов подсетей рассматриваемого примера.

Таблица 2.5. Возможные варианты подсетей

	Точечно-десятичная нотация	Двоичное представление адреса
Базовая сеть	193.1.1.0/24	11000001.00000001.00000001.00000000
Подсеть #0	193.1.1.0/27	11000001.00000001.00000001.<u>00</u>000000
Подсеть #1	193.1.1.32/27	11000001.00000001.00000001.<u>00</u>100000
Подсеть #2	193.1.1.64/27	11000001.00000001.00000001.<u>01</u>000000
Подсеть #3	193.1.1.96/27	11000001.00000001.00000001.<u>01</u>100000
Подсеть #4	193.1.1.128/27	11000001.00000001.00000001.**10**000000
Подсеть #5	193.1.1.160/27	11000001.00000001.00000001.<u>10</u>100000
Подсеть #6	193.1.1.192/27	11000001.00000001.00000001.<u>11</u>000000
Подсеть #7	193.1.1.224/27	11000001.00000001.00000001.<u>11</u>100000

Самым простым способом проверки корректности определения подсетей является контроль кратности всех десятичных номеров подсетей номеру подсети #1. В данном примере все номера подсетей кратны 32.

ПРИМЕЧАНИЕ

Компания Cisco Systems включила в свой программный продукт ConfigMaker v2.4 инструмент работы с адресами, называемый IP Subnet Calculator. Администратор, введя адрес и маску подсети, получает полную классификацию (рис. 2.12).

Когда схема введения подсетей была опубликована в документе RFC 950, запрещалось использование номеров подсетей, у которых все биты установлены в единицы или нули. Причиной такого ограничения является необходимость устранения возможных проблем при работе тех протоколов маршрутизации, которые не переносят в своих служебных сообщениях ни маску подсети, ни

длину расширенного сетевого префикса. Например, если используется протокол маршрутизации RIP (его первая версия обозначается как RIP-1), который не учитывает маску подсети и длину расширенного сетевого префикса, маршруты в разные подсети с адресами 193.1.1.0/27 и 193.1.1.0/24 будут рассматриваться как идентичные. То есть без указания маски подсети маршрутизатор не делает различий между маршрутом в одну подсеть и маршрутом во всю сеть. Похожая проблема, но только с определением направления широковещания возникает и в случае равенства всех битов единице. Например, один адрес 193.1.1.255 используется как широковещательный для всей сети 193.1.1.0/24 и для подсети 193.1.1.224/27. Рисунок 2.13 иллюстрирует обе рассмотренные ситуации.

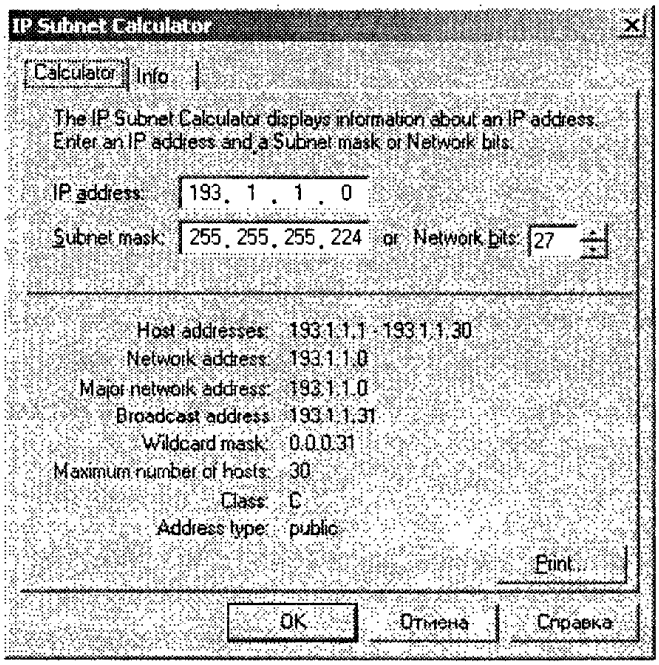


Рис. 2.12. Cisco IP Subnet Calculator

Маршруты в сети		24-разрядный расширенный сетевой префикс			Номер хоста
	193.1.1.0/24	11000001	00000001	00000001	00000000
		27-разрядный расширенный сетевой префикс			
	193.1.1.0/27	11000001	00000001	00000001.000	00000
Широковещательные адреса		24-разрядный расширенный сетевой префикс			
	193.1.1.0/24	11000001	00000001	00000001	11111111
		27-разрядный расширенный сетевой префикс			
	193.1.1.224/27	11000001	00000001	00000001.111	11111

Рис. 2.13. Идентичные маршруты и широковещательные адреса

С разработкой протоколов маршрутизации, переносящих маску подсети (OSPF, IS-IS) с каждым рекламируемым маршрутом, стало возможно вопреки документу RFC 950 использовать подсети, все биты которых установлены в единицу и ноль. По этой причине производители позволяют настраивать подсети с такими номерами на портах своих маршрутизаторов. При этом нужно учитывать следующие два фактора: протоколы маршрутизации внутри корпоративной сети, относящиеся к классу IGP, должны поддерживать маску подсети или расширенный сетевой префикс; необходима поддержка номеров подсетей со всеми единичными и нулевыми битами всеми маршрутизаторами в сети. Кроме того, иногда важно учитывать номер версии программного обеспечения маршрутизатора.

В рассматриваемом примере (табл. 2.6) остается 5 бит для задания адресов хостов в каждой подсети. В результате каждая подсеть может содержать блок из 30 адресов хостов ($2^5 - 2 = 30$), которые нумеруются от 1 до 30. Для определения адреса хоста N в сети администратор помещает двоичное представление числа N в поле номера хоста. Например, для определения адреса, который необходимо присвоить хосту #28 в подсети #2, администратор просто вписывает двоичное представление 28 (11100_2) в пятибитовое поле подсети #2. В табл. 2.6 показаны некоторые возможные комбинации номеров хостов в подсети #2.

Таблица 2.6. Возможные варианты адресации хостов в подсети #2

	Точечно-десятичная нотация	**Двоичное представление**
Подсеть #2	193.1.1.64/27	11000001.00000001.00000001.010**00000**
Хост #1	193.1.1.65/27	11000001.00000001.00000001.010**00001**
Хост #2	193.1.1.66/27	11000001.00000001.00000001.010**00010**
Хост #3	193.1.1.67/27	11000001.00000001.00000001.010**00011**
...		
Хост #28	193.1.1.92/27	11000001.00000001.00000001.010**11100**
Хост #29	193.1.1.93/27	11000001.00000001.00000001.010**11101**
Хост #30	193.1.1.93/27	11000001.00000001.00000001.010**11110**
Широковещательный адрес для подсети #2		
	193.1.1.95	11000001.00000001.00000001.010**11111**

Для того чтобы проверить корректность широковещательного адреса для определенной подсети, можно придерживаться следующего простого правила: во всех случаях широковещательный адрес для подсети #N на единицу меньше, чем базовый адрес для подсети #(N+1). Например, широковещательный адрес для подсети #2 (193.1.1.95) на единицу меньше базового адреса подсети #3 (193.1.1.96).

Перед передачей дейтаграммы хосту необходимо определить слежующее.

1. Располагается ли получатель в той же подсети, что и отправитель?
2. Если существует более чем один маршрутизатор, имеющий маршрут в нужную сеть, какой маршрутизатор должен выбрать отправитель?

До введения подсетей процесс передачи выполнялся путем выделения поля сетевого номера из IP-адреса получателя, содержащегося в заголовке IP-дейтаграммы, и последующего сравнения с собственным IP-адресом хоста. Если сетевые номера адреса отправителя и получателя совпадают, то оба располагаются в одной локальной сети, и дейтаграмма может быть передана напрямую. Если номера сетей разные, то дейтаграмму необходимо посылать получателю через маршрутизатор по умолчанию.

После введения подсетей этот процесс значительно усложнился, так как получатель может располагаться в другой подсети той же самой сети, что и получатель. Процесс передачи в этом случае осуществляется при помощи маски подсети. То есть применяется операция «логическое И» к IP-адресу получателя и маске подсети. Результат сравнивается с результатом выполнения той же операции, но для собственного IP-адреса хоста и той же маски подсети. Если результаты обеих операций идентичны, то отправитель и получатель находятся в одной подсети и дейтаграмма может быть послана напрямую. Если результаты различны, то получатель находится в другой подсети. В этом случае дейтаграмма отправляется маршрутизатору.

В табл. 2.7 и 2.8 представлены некоторые возможные (и наиболее часто используемые) варианты выделения подсетей для сетей классов B и C.

Таблица 2.7. Возможные комбинации выделения подсетей для сети класса B

Маска подсети	Префикс	Биты для подсети	Биты для устройств	Количество подсетей	Количество устройств
255.255.0.0	/16	0	16	0 (1 сеть)	65 534
255.255.192.0	/18	2	14	2	16382
255.255.224.0	/19	3	13	6	8190
255.255.240.0	/20	4	12	14	4094
255.255.248.0	/21	5	11	30	2046
255.255.252.0	/22	6	10	62	1022
255.255.254.0	/23	7	9	126	510
255.255.255.0	/24	8	8	254	254
255.255.255.128	/25	9	7	510	126
255.255.255.192	/26	10	6	1022	62
255.255.255.224	/27	11	5	2046	30
255.255.255.240	/28	12	4	4094	14
255.255.255.248	/29	13	3	8190	6
255.255.255.252	/30	14	2	16 382	2

При рассмотрении примера разделения на подсети сети класса C 193.1.1.0/24 была отмечена рекомендация относительно порядка расположения битов при выделении подсетей. В документе RFC 1219 описано основное правило, которому желательно следовать при присвоении номеров подсетям и хостам. Номера подсетей назначают таким образом, чтобы старшие биты в номере подсети устанавливались первыми. Например, если поле номера подсети состоит из четырех битов,

то первые несколько номеров подсетей должны быть следующими: 8 (1000_2), 4 (0100_2), 12 (1100_2), 2 (0010_2), 6 (0110_2) и т. д. Иными словами, единичные биты номеров подсетей рекомендуется устанавливать, начиная с крайней левой позиции. В то время как единичные биты номеров хостов рекомендуется устанавливать, начиная с крайней правой позиции (табл. 2.9).

Таблица 2.8. Возможные комбинации выделения подсетей для сети класса C

Маска подсети	Префикс	Биты для подсети	Биты для устройств	Количество подсетей	Количество устройств
255.255.255.0	/24	0	8	0 (1 сеть)	254
255.255.255.192	/26	2	6	2	62
255.255.255.224	/27	3	5	6	30
255.255.255.240	/28	4	4	14	14
255.255.255.248	/29	5	3	30	6
255.255.255.252	/30	6	2	62	2

Таблица 2.9. Пример присвоения номеров подсетей и хостов

Сетевой префикс		Номер подсети	Номер хоста
11111111	11111111	3 бит	Остальные 12 бит
Рекомендуемая схема присвоения адресов			
Биты подсетей			**Биты хостов**
128	1000 **0000. 0000** 0001		1
64	0100 **0000. 0000** 0010		2
192	1100 **0000. 0000** 0011		3
32	0010 **0000. 0000** 0100		4
160	1010 **0000. 0000** 0101		5
96	0110 **0000. 0000** 0110		6
224	1110 **0000. 0000** 0111		7

Если следовать этому правилу, то на границе между номером подсети и номером хоста будут существовать нулевые биты. Это позволяет менять маску подсети без изменения IP-адреса, присвоенного хосту. Необходимость в изменении маски подсети может возникнуть при расширении числа хостов в каждой подсети, с учетом того, что планируемое число возможных подсетей обычно больше необходимого в настоящий момент. В таком случае существует возможность «заимствования» под номера подсетей некоторых битов из числа зарезервированных. Достоинством описанного правила является то, что администратору достаточно обновить маску подсети на каждом хосте и не нужно переконфигурировать IP-адреса хостов во всей организации. Изменение адресов потребует больших усилий от администратора, так как в данном случае может потребоваться перенастройка почтовых сервисов, статических таблиц маршрутизации и т. д.

В сети, которая была разбита на подсети, применимы два типа широковещательной передачи информации: направленное широковещание и ограниченное. Направленное широковещание имеет целью передачу дейтаграммы всем хостам в определенной подсети. Для посылки дейтаграммы всем хостам во всех подсетях следует использовать ограниченное широковещание с адресом 255.255.255.255. Однако необходимо учесть, что маршрутизаторы не пропускают дейтаграммы с таким адресом. Существует одно ограничение, накладываемое на направленное широковещание в средах с подсетями. Биты, ответственные за формирование номеров подсетей, обычно являются частью поля номера хоста и не могут быть все сразу установлены в нули или единицы. Например, если есть адрес класса B, в котором третий байт выделен под номера подсетей — 128.1.<Номер подсети>.<Номер хоста>, то в этом случае адрес направленного широковещания не может быть равен 128.1.255.255, 128.1.0.255, 128.1.255.0 и 128.1.0.0.

Существуют теоретические и практические аргументы, призывающие не использовать «верхние» (все биты единичные) и «нижние» подсети (все биты нулевые) в распределенных сетях (встречается также термин «граничные подсети»). Как уже не раз отмечалось, в теории битовое поле содержит два специальных значения: все биты сброшены в ноль, что обычно означает «данный», например «данное устройство» или «данная сеть», и все биты установлены в единицу, что означает «все», например «все устройства» или «все сети». Ранние документы, регламентирующие адресацию в Интернете, рекомендовали оставлять эти значения в неприкосновенности. Это накладывало запрет на использование адреса нижней сети, состоящего только из нулей, и адреса верхней сети, составленного только из единиц. В результате программное обеспечение, обслуживающее протокол IP на сетевых устройствах, убеждалось в факте настройки пользователем адресации в обход этих правил. На практике использование таких подсетей может быть проблематичным, так как не все устройства, особенно выпущенные давно, поддерживают данного рода настройку. Для того чтобы избежать проблем, следует быть хорошо осведомленным о конкретном устройстве и выяснить, какое адресное пространство оно позволяет настроить. Маршрутизаторы фирмы Cisco Systems допускают настройку на использование рассматриваемых подсетей таким образом, что в свое распоряжение можно получить еще одну подсеть. Для того чтобы задействовать нижнюю подсеть, нужно выполнить команду **ip subnet-zero** в глобальном режиме конфигурации.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip subnet-zero
```

В том случае, если администратор сети не вводил эту команду на маршрутизаторе, попытка настроить интерфейс с адресом нижней подсети приведет к появлению сообщения об ошибке:

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no ip subnet-zero
Router(config)#interface serial1
Router(config-if)#ip address 192.168.1.2 255.255.255.224
Bad mask /27 for address 192.168.1.2
```

Практика выделения IP-подсетей

Рассмотрим практические примеры выделения подсетей и назначения адресов устройствам в этих подсетях. Предположим, что организации для ее корпоративной сети предоставлен сетевой адрес 140.25.0.0/16 (или адрес 140.25.0.0 класса B — в существующей технической литературе автор не нашел устоявшегося определения). При этом организация планирует разделить сеть на несколько подсетей, каждая из которых должна поддерживать до 60 устройств.

На первом шаге необходимо определить число битов, требуемых для идентификации 60 устройств в подсети. Адрес конкретного устройства имеет определенное двоичное представление, и верхняя граница адресного пространства для устройств одной подсети выражается степенью числа 2. Это, в частности, означает, что невозможно выделить адресное пространство точно для 60 устройств, так как 60 не является степенью двойки. Ближайшая сверху степень двойки — это $64 = 2^6$. На самом деле, к числу устройств нужно прибавить 2, поскольку адреса, содержащие только нули или только единицы, не используются для адресации отдельных устройств. Выбрав эту степень двойки, мы ввели два запасных адреса: $(60 + 2 = 62) < 64$. Однако, удовлетворяя существующие на сегодня потребности по числу рабочих мест, такой выбор не оставляет адресного пространства для возможного роста подсети. И хотя следующая степень двойки равна 128 (2^7) и число адресов устройств будет равно $2^7 - 2 = 126$, то есть намного больше требуемого в настоящий момент, сетевому администратору следует выбрать именно это адресное пространство и получить 66 (126 – 60) дополнительных адресов для каждой подсети. Такой выбор означает, что поле адреса устройства займет 7 бит.

На втором шаге определяется маска подсети и длина расширенного сетевого префикса. Поскольку для идентификации устройств из 32-разрядного IP-адреса решено выделить 7 бит, получаем расширенный сетевой префикс /25 (32 – 7 = 25). Такой 25-разрядный расширенный сетевой префикс может быть выражен в точечно-десятичном представлении маской подсети 255.255.255.128. Рисунок 2.14 показывает запись маски подсети и расширенного сетевого префикса.

	Сетевой префикс		Номер подсети	Номер устройства
140.25.0.0/16	1001100	00011001	00000000.**0**	0000000
255.255.255.128	11111111	11111111	11111111.**1**	0000000
Или эквивалентная запись:				
	25-битовый расширенный сетевой префикс			Номер устройства
140.25.0.0/25	1001100	00011001	00000000.**0**	0000000

Рис. 2.14. Определение маски подсети и расширенного сетевого префикса

Как видим, 25-разрядный расширенный префикс предполагает выделение 9 бит для идентификации подсетей. Теперь можно вычислить количество идентифицируемых подсетей: $2^9 = 512$, то есть 9 бит позволяют назначить адреса 512 подсетям. Понятно, что администратор сети получает в этом случае некоторую свободу действий при определении соотношения числа идентифицируемых устройств и числа подсетей. Выделяя большее количество битов в поле идентификации устройств, администратор становится в состоянии включать в подсеть больше устройств. С другой стороны, чем меньше битов выделено для идентификации устройств, тем больше подсетей способен создать администратор. Все зависит от текущих требований организации.

Выделенные 512 подсетей пронумеруем от 0 до 511. Если использовать 9 разрядов для двоичного представления десятичных чисел от 0 до 511, то получим: 0 $(000000000)_2$, 1 $(000000001)_2$, 2 $(000000010)_2$, 3 $(000000011)_2$, ..., 511 $(111111111)_2$. Например, для определения подсети номер 3 (#3) сетевой администратор укладывает двоичное представление числа 3 $(000000011)_2$ в 9 битов номера подсети. Номера подсетей для рассматриваемого примера приводятся ниже. В каждом адресе *курсивом* выделен расширенный сетевой префикс полного адреса, в то время как 9-битовое представление поля номера подсети имеет **полужирное** выделение:

Базовая сеть:**10001100.00011001.00000000.**00000000 = 140.25.0.0/16
Подсеть #0:*10001100.00011001.***00000000.**00000000 = 140.25.0.0/25
Подсеть #1:*10001100.00011001.***00000000.1**0000000 = 140.25.0.128/25
Подсеть #2:*10001100.00011001.***00000001.**00000000 = 140.25.1.0/25
Подсеть #3:*10001100.00011001.***00000001.1**0000000 = 140.25.1.128/25
Подсеть #4:*10001100.00011001.***00000010.**00000000 = 140.25.2.0/25
Подсеть #5:*10001100.00011001.***00000010.1**0000000 = 140.25.2.128/25
Подсеть #6:*10001100.00011001.***00000011.**00000000 = 140.25.3.0/25
Подсеть #7:*10001100.00011001.***00000011.1**0000000 = 140.25.3.128/25
Подсеть #8:*10001100.00011001.***00000100.**00000000 = 140.25.4.0/25
Подсеть #9:*10001100.00011001.***00000100.1**0000000 = 140.25.4.128/25
...
Подсеть #510:*10001100.00011001.***11111111.**00000000 = 140.25.255.0/25
Подсеть #511:*10001100.00011001.***11111111.10000000 = 140.25.255.128/25**

Итак, администратор выделил 7 бит для идентификации устройств в каждой подсети. Это означает, что каждая подсеть имеет 126 адресов для идентификации устройств. Устройства в подсети нумеруются от 1 до 126. Приведем перечень адресов устройств для подсети #3. При этом курсивом выделен расширенный сетевой префикс, а выделение жирным относится к 7-разрядному полю номера устройства:

Подсеть #3:*10001100.00011001.00000001.1***0000000** = 140.25.1.128/25
Устройство #1:*10001100.00011001.00000001.1***0000001** = 140.25.1.129/25
Устройство #2:*10001100.00011001.00000001.1***0000010** = 140.25.1.130/25
Устройство #3:*10001100.00011001.00000001.1***0000011** = 140.25.1.131/25
Устройство #4:*10001100.00011001.00000001.1***0000100** = 140.25.1.132/25

Устройство #5:*10001100.00011001.00000001.1***0000101** = 140.25.1.133/25
Устройство #6:*10001100.00011001.00000001.1***0000110** = 140.25.1.134/25
...
Устройство #62:*10001100.00011001.00000001.1***0111110** = 140.25.1.190/25
Устройство #63:*10001100.00011001.00000001.1***0111111** = 140.25.1.191/25
Устройство #64: *10001100.00011001.00000001.1***1000000** = 140.25.1.192/25
Устройство #65: *10001100.00011001.00000001.1***1000001** = 140.25.1.193/25
...
Устройство #123:*10001100.00011001.00000001.1***1111011** = 140.25.1.251/25
Устройство #124:*10001100.00011001.00000001.1***1111100** = 140.25.1.252/25
Устройство #125:*10001100.00011001.00000001.1***1111101** = 140.25.1.253/25
Устройство #126:*10001100.00011001.00000001.1***1111110** = 140.25.1.254/25

Для подсети #3 широковещательным адресом будет адрес, в котором все биты поля номера устройства установлены в единицу:

*10001100.00011001.00000001.1***1111111** = 140.25.1.255.

Следует отметить, что широковещательный адрес для подсети #3 ровно на единицу меньше базового адреса подсети #4 (140.25.2.0).

Рассмотрим другую ситуацию. Пусть организации назначен сетевой адрес 132.45.0.0/16. Администратору поручено сформировать 8 подсетей. Для идентификации такого количества подсетей требуется три бита. В этом случае расширенный сетевой префикс будет равен /19 (маска подсети 255.255.224.0). Приведем адреса этих подсетей в двоичном и десятичном представлениях:

Подсеть #0:*10000100.00101101.***00**000000.00000000 = 132.45.0.0/19
Подсеть #1:*10000100.00101101.***00**100000.00000000 = 132.45.32.0/19
Подсеть #2:*10000100.00101101.***01**000000.00000000 = 132.45.64.0/19
Подсеть #3:*10000100.00101101.***01**100000.00000000 = 132.45.96.0/19
Подсеть #4:*10000100.00101101.***10**000000.00000000 = 132.45.128.0/19
Подсеть #5:*10000100.00101101.***10**100000.00000000 = 132.45.160.0/19
Подсеть #6:*10000100.00101101.***11**000000.00000000 = 132.45.192.0/19
Подсеть #7:*10000100.00101101.***11**100000.00000000 = 132.45.224.0/19

Теперь определим адреса устройств для подсети #3 (132.45.96.0/19, или *10000100.00101101.011*00000.00000000):

Подсеть #3:*10000100.00101101.***01**100000.00000000 = 132.45.96.0/19
Устройство #1:*10000100.00101101.*011**00000.00000001** = 132.45.96.1/19
Устройство #2:*10000100.00101101.*011**00000.00000010** = 132.45.96.2/19
Устройство #3:*10000100.00101101.*011**00000.00000011** = 132.45.96.3/19
...
Устройство #8190: *10000100.00101101.*01**1111111.11111110** = 132.45.127.254/19

Определим широковещательный адрес для подсети #3 (132.45.96.0/19):

*10000100.00101101.*01**1111111.11111111** = 132.45.127.255/19

Проделаем те же операции для сетевого адреса 200.35.1.0/24. Пусть также в каждой подсети необходимо предусмотреть адресное пространство для 20 устройств.

Требуется определить расширенный сетевой префикс. Для идентификации устройств требуется минимум пять битов. Поэтому расширенный сетевой префикс будет равен /27 (32–5=27). При этом максимальное количество устройств в каждой подсети — 30 (25–2 = 32–2=30), а максимальное число подсетей равно 8 (23).

Вот номера получившихся подсетей в двоичном и десятичном представлениях:

Подсеть #0:*11001000.00100011.00000001*.**00**000000 = 200.35.1.0/27
Подсеть #1:*11001000.00100011.00000001*.**00**100000 = 200.35.1.32/27
Подсеть #2:*11001000.00100011.00000001*.**01**000000 = 200.35.1.64/27
Подсеть #3:*11001000.00100011.00000001*.**01**100000 = 200.35.1.96/27
Подсеть #4:*11001000.00100011.00000001*.**10**000000 = 200.35.1.128/27
Подсеть #5:*11001000.00100011.00000001*.**10**100000 = 200.35.1.160/27
Подсеть #6:*11001000.00100011.00000001*.**11**000000 = 200.35.1.192/27
Подсеть #7:*11001000.00100011.00000001*.**11**100000 = 200.35.1.224/27

Приведем список адресов устройств, которые могут быть определены в подсети #6 (200.35.1.192/27):

Подсеть #6:*11001000.00100011.00000001*.11**000000** = 200.35.1.192/27
Устройство #1:*11001000.00100011.00000001*.11**000001** = 200.35.1.193/27
Устройство #2:*11001000.00100011.00000001*.11**000010** = 200.35.1.194/27
Устройство #3:*11001000.00100011.00000001*.11**000011** = 200.35.1.195/27
...
Устройство #29:*11001000.00100011.00000001*.11**011101** = 200.35.1.221/27
Устройство #30:*11001000.00100011.00000001*.11**011110** = 200.35.1.222/27

Широковещательный адрес для подсети 200.35.1.192/27 равен:

11001000.00100011.00000001.11**011111** = 200.35.1.223/27

Маска подсети переменной длины

Документ RFC 1009 от 1987 года сформулировал, каким образом в сетях, состоящих из нескольких подсетей, можно использовать больше одной маски подсети. В ситуации, когда в распределенной IP-сети назначается несколько масок подсетей, она рассматривается как сеть с масками подсетей переменной длины, так как в этом случае расширенные сетевые префиксы в различных подсетях имеют разную длину.

СОВЕТ

Если в сети планируется использовать маску подсети переменной длины (Variable Length Subnet Mask — VLSM), следует включать в работу протоколы маршрутизации OSPF или EIGRP. Протоколы RIP и IGRP относятся к так называемым протоколам маршрутизации на основе классов (classful) и не поддерживают VLSM.

При применении протокола маршрутизации RIP первой версии (RIP-1 IP) в сети поддерживается только одна маска подсети с каждым адресом (а точнее, с каждым номером сети), так как протокол не переносит информацию о масках

подсетей в своих сообщениях об обновлении маршрутизации (сообщения, рассылаемые маршрутизаторами при задействовании протоколов маршрутизации, таких как RIP, IGRP и т. п.). При отсутствии данной информации протокол маршрутизации RIP-1 IP принимает простое решение, выбирая маску подсети, которая соотносится с каждым маршрутом в таблице маршрутизации.

При этом возникает вопрос: каким образом маршрутизатор, работающий по протоколу RIP-1, выбирает, какую маску необходимо сопоставить с новым маршрутом, полученным от соседнего маршрутизатора? Если в полученном маршруте номер адреса сети совпадает с номером адреса сети, присвоенным любому локальному порту маршрутизатора, то будет использоваться маска подсети порта маршрутизатора. Если же номера сетей не совпадают, то за основу будет взята классическая маска подсети.

Покажем это на примере. Предположим, что порту 1 маршрутизатора был присвоен адрес **130.24.**13.1 с маской 255.255.255.0 (расширенный сетевой префикс /24), а порту 2 — адрес 200.14.13.2 с такой же маской подсети (тот же расширенный сетевой префикс). Анализируя первые биты адреса порта 1 и маску подсети, маршрутизатор определяет, что это адрес класса B, а третий байт адреса используется для задания номера подсети. Порту 2 присвоен адрес класса C без выделения подсетей.

Если маршрутизатор получает информацию о маршруте в сеть **130.24.**36.0 от соседнего маршрутизатора, он будет работать с маской подсети 255.255.255.0 (расширенный сетевой префикс /24), так как порту 1 присвоен адрес с тем же номером сети — **130.24.**0.0. Маска подсети просто наследуется. Но если маршрутизатор получит от соседа информацию о маршруте в сеть 131.25.0.0, он станет использовать стандартную маску подсети 255.255.0.0, так как адрес 131.25.0.0 является адресом класса B, а этому классу соответствует маска подсети 255.255.0.0. Актуальной будет именно эта маска, так как маршрутизатор не имеет дополнительной информации о необходимой маске подсети.

Маршрутизатор, поддерживающий протокол RIP-1 IP, будет включать биты, определяющие номера подсетей, в свои сообщения об обновлении маршрутов только в том случае, если порт, через который предполагается посылать сообщения, настроен на подсеть с тем же номером сети. Если порт настроен с другим сетевым номером, то маршрутизатор станет посылать только сетевую часть адреса.

Продемонстрируем сказанное, продолжая рассмотрение предыдущего примера. Предположим, что маршрутизатор получил информацию от соседа о маршруте в сеть **130.24.**36.0. Так как порт 1 настроен на адрес того же класса, то маршрутизатор предполагает, что сеть **130.24.**36.0 имеет маску 255.255.255.0. Поэтому, когда наступает время рекламировать данный маршрут, маршрутизатор информирует о маршруте сеть с адресом **130.24.36.**0 через свой порт 1, а также в сеть с адресом 130.24.0.0 через порт 2, то есть происходит потеря информации, содержащейся в третьем байте адреса (36).

Протокол RIP-1 IP может использовать с каждым номером сети только одну маску подсети. Возможность назначения одному адресу нескольких масок подсетей дает следующие преимущества:

- множество масок подсетей позволяет более эффективно обходиться с выделенным организации адресным пространством;

- множество масок подсетей позволяет объединять маршруты, что значительно уменьшает количество маршрутной информации внутри домена маршрутизации организации.

Применение маски подсети переменной длины позволяет эффективнее использовать адресное пространство протокола IP, выделенное организации. Ранее единственность маски подсети ограничивала возможности организации, так как номер и размер подсети были фиксированными.

Например, предположим, что администратор намеревается настроить выделенную организации сеть класса B 130.5.0.0 на использование расширенного сетевого префикса /22 (табл. 2.10). Для задания номеров подсетей необходимо и достаточно 6 бит.

Таблица 2.10. Распределение адресного пространства для префикса /22

Адрес сети	Сетевой префикс (класс B)		Номер подсети	Номер хоста
130.5.0.0 /22	10000010	00000101	000000	00. 00000000
	Расширенный сетевой префикс /22			

Сеть класса B с расширенным сетевым префиксом /22 позволяет организовать 64 подсети ($2^6 = 64$), каждая из которых поддерживает максимум 1022 ($2^{10} - 2 = 1022$) индивидуальных адреса хостов. Такой вариант может устроить администратора, если организации нужно некоторое число подсетей с большим количеством хостов. Однако если организации не нужны подсети с числом хостов более 30, то имея фиксированную маску подсети, администратору придется эксплуатировать подсети, рассчитанные на большое количество хостов, подключая к ним всего нескольких пользователей. В результате невостребованными могут оказаться около 1000 допустимых адресов хостов в подсетях. Как видно из этих рассуждений, ограничение, полученное по причине единственной маски подсети, отрицательно, причем существенно, влияет на эффективность использования адресного пространства.

Основным решением данной проблемы является работа с маской подсети переменной длины. Рассмотрим другой пример. Предположим, что администратор хочет установить расширенный сетевой префикс /26. Адрес класса B с таким расширенным сетевым префиксом позволит поддерживать до 1024 подсетей ($2^{10} = 1024$), каждая из которых охватывает до 62 ($2^6 - 2 = 62$) адресов хостов (табл. 2.11). Такой расширенный сетевой префикс идеально подходит к небольшим подсетям с числом хостов порядка 60, в то время как префикс /22 ориентирован на большие подсети с тысячами хостов.

Таблица 2.11. Распределение адресного пространства для префикса /26

Адрес сети	Сетевой префикс		Номер подсети	Номер хоста
130.5.0.0/26	10000010	00000101	**00000000. 00**	000000
	Расширенный сетевой префикс /26			

Сочетание различных расширенных сетевых префиксов — /22 и /26 — позволило получить два типа подсетей с резким отличием по числу поддерживаемых хостов. Внедрение маски подсети переменной длины позволяет администратору получать в рамках своей организации подсети с необходимыми характеристиками на нужный период времени. Это происходит следующим образом: сначала сеть делится на подсети, затем некоторые из этих подсетей делятся, в свою очередь, на подсети и т. д., то есть происходит рекурсивное дробление количества подсетей.

На рис. 2.15 показан пример, в котором сеть класса А с адресом 10.0.0.0 сначала делится на подсети с расширенным сетевым префиксом /16 (маска подсети 255.255.0.0). Общее количество получаемых подсетей — 254. В каждой подсети поддерживается до 65 534 (2^{16} – 2 = 65 534) адресов хостов. Подсеть с адресом 10.253.0.0 при рекурсии с расширенным сетевым префиксом /24 поддерживает до 254 подсетей, каждая из которых включает в себя до 254 (2^8 - 2 = 254) адресов хостов. При дальнейшей рекурсии с расширенным сетевым префиксом /27 в состав подсети с адресом 10.253.1.0 войдут 6 подсетей с номерами, кратными 32, каждая из которых будет поддерживать до 30 (2^5 – 2 = 30) номеров хостов.

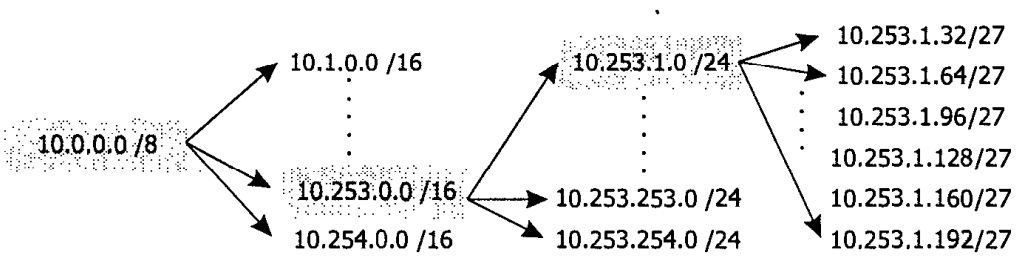


Рис. 2.15. Пример рекурсивного размножения адресов подсетей

Таким образом, рекурсивное разбиение адресного пространства организации может быть выполнено с учетом пожеланий администратора сети. Кроме возможности рекурсии, внедрение маски подсети переменной длины позволяет значительно уменьшить объем таблиц маршрутизации для маршрутизаторов организации. Каждый маршрутизатор имеет возможность объединять свои подсети в одну запись в сообщении об обновлении. Поскольку структура подсетей не видна вне организации, граничный маршрутизатор рекламирует в Интернете маршрут с адресом 10.0.0.0 (рис. 2.16).

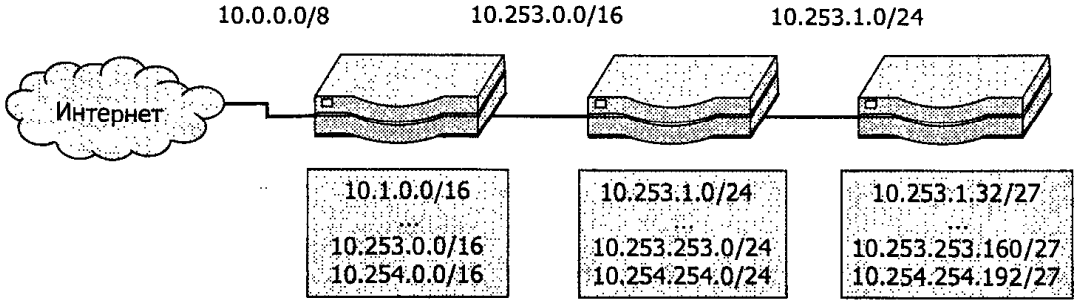


Рис. 2.16. Объединение маршрутов с помощью VLSM

При разработке концепции подсетей, определяемых маской подсети переменной длины администратору необходимо убедиться в том, что в предоставляемом организации IP-адресе свободно достаточное количество битов для формирования требуемого числа подсетей на каждом уровне рекурсии.

Предположим, что сеть организации охватывает несколько удаленных филиалов. Если организация имеет три удаленных сети, то сегодня ей потребуется выделить три бита для формирования подсетей, которых может хватить как сегодня, так и в обозримом будущем ($2^3 = 8$). Но допустим, что администратор хочет сформировать отдельные подсети внутри каждого филиала. Эти подсети будут вторым уровнем в иерархии подсетей. Кроме того, внутри каждого филиала необходимо поддерживать отдельные рабочие группы, для которых также нужно выделить подсети. Следуя приведенной модели иерархии, верхний уровень определяется числом удаленных филиалов, второй — количеством зданий во владении каждого филиала, а третий — максимальным числом подсетей в каждом здании и максимальным числом хостов в каждой из подсетей.

Для успешного внедрения маски подсети переменной длины требуется выполнить три основных условия:

- протокол маршрутизации должен переносить информацию о маске подсети в каждом сообщении;
- все маршрутизаторы обязаны поддерживать алгоритм передачи, основывающийся на технологии наибольшего совпадения (longest match);
- адреса должны присваиваться в соответствии с существующей топологией сети.

Современные протоколы маршрутизации, такие как OSPF и IS-IS, умеют оперировать маской подсети переменной длины. Это достигается с помощью передачи значения маски подсети в каждом сообщении об обновлении маршрутов, что позволяет рекламировать каждую подсеть с ее маской. Если протокол маршрутизации не работает с масками подсети, маршрутизатор либо предполагает, что должна использоваться маска подсети, присвоенная его локальному порту, либо выполняет поиск в статически настроенной таблице, содержащей всю информацию о масках подсетей. Первое решение не в состоянии гарантировать выбор корректной маски подсети, а статическая таблица не имеет возможности масштабирования, кроме того, она сложна в управлении и исправлении ошибок.

Таким образом, если требуется задействовать маски подсети переменной длины в сложной сетевой топологии, наилучшим вариантом является применение протоколов маршрутизации OSPF и IS-IS, а не RIP-1 IP. Однако при этом нужно учитывать, что вторая версия протокола RIP (RIP-2 IP), описанная в документе RFC 1388, расширяет возможности первой версии протокола, в том числе и добавлением функциональности переноса маски подсети.

Применение маски подсети переменной длины

Рассмотрим практические примеры работы с маской подсети переменной длины. Предположим, что организации был выделен сетевой адрес 140.25.0.0/16 (IP-адрес класса B), и администратор планирует принять во внимание маски подсети

переменной длины. На рис. 2.17 изображена вероятная схема выделения адресов для подсетей этой организации.

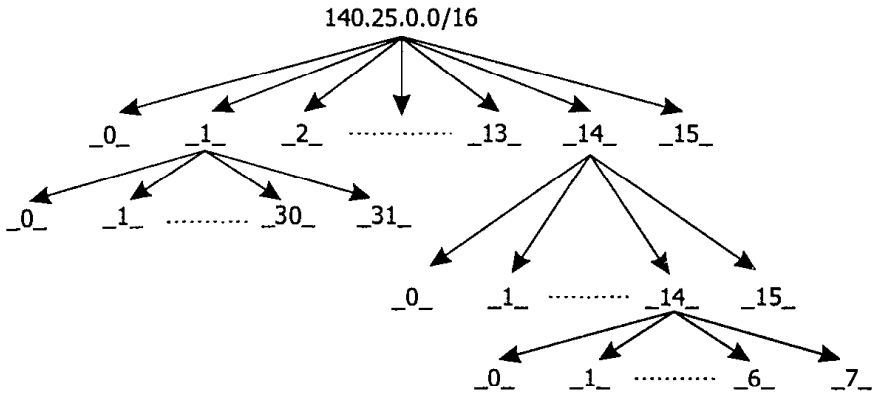


Рис. 2.17. Стратегия выделения подсетей

Первый шаг в процессе выделения подсетей состоит в делении основного сетевого адреса (140.25.0.0/16) на 16 адресных блоков равного размера (рис. 2.18). Затем подсеть #1 разбивается на 32 одинаковых по длине адресных блока, а подсеть #14 также делится на 16 одинаковых адресных блоков (подсетей нижнего уровня). Промежуточные сети #2–13 не разбиваются. Полученная подсеть нижнего уровня, например 14-я подсеть в 14-й подсети (обозначим ее как #14-14), делится на 8 адресных блоков (подсетей) равного размера, которые образуют подсети следующего уровня. Таким образом, в сети организации планируется получить 74 подсети, каждая из которых будет поддерживать определенное количество устройств.

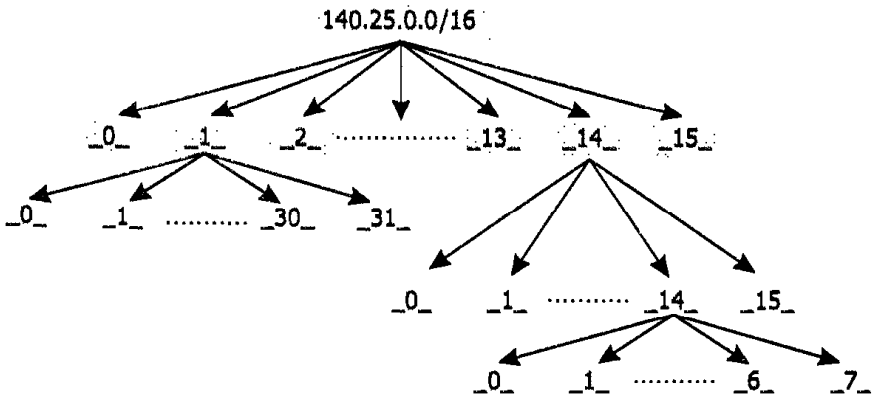


Рис. 2.18. Выделение 16 подсетей для адреса 140.25.0.0/16

Так как $16 = 2^4$, то потребуется 4 бита, чтобы идентифицировать каждую из 16 подсетей. Это означает, что администратор нуждается в 4 битах или в расширенном сетевом префиксе /20, для того чтобы выделить 16 подсетей. Каждая из этих подсетей представляет собой смежный блок из 2^{12} (32 – 20 = 12) адресов устройств. Таким образом, мы получаем $2^{12} - 2 = 4094$ устройства. Ниже описы-

ваются 16 подсетей, выделенные из адресного блока 140.25.0.0/16. Все подсети нумеруются от 0 до 15. Курсивом выделен расширенный сетевой префикс, жирным начертанием — 4-разрядный номер подсети:

Базовая сеть:*10001100.00011001.***0000**0000.00000000 = 140.25.0.0/16
Подсеть #0:*10001100.00011001.***0000**0000.00000000 = 140.25.0.0/20
Подсеть #1:*10001100.00011001.***0001**0000.00000000 = 140.25.16.0/20
Подсеть #2:*10001100.00011001.***0010**0000.00000000 = 140.25.32.0/20
Подсеть #3:*10001100.00011001.***0011**0000.00000000 = 140.25.48.0/20
Подсеть #4:*10001100.00011001.***0100**0000.00000000 = 140.25.64.0/20
...
Подсеть #13:*10001100.00011001.***1101**0000.00000000 = 140.25.208.0/20
Подсеть #14:*10001100.00011001.***1110**0000.00000000 = 140.25.224.0/20
Подсеть #15:*10001100.00011001.***1111**0000.00000000 = 140.25.240.0/20

Определим адреса устройств, которые могут быть назначены в подсети #3 (140.25.48.0/20) (рис. 2.19).

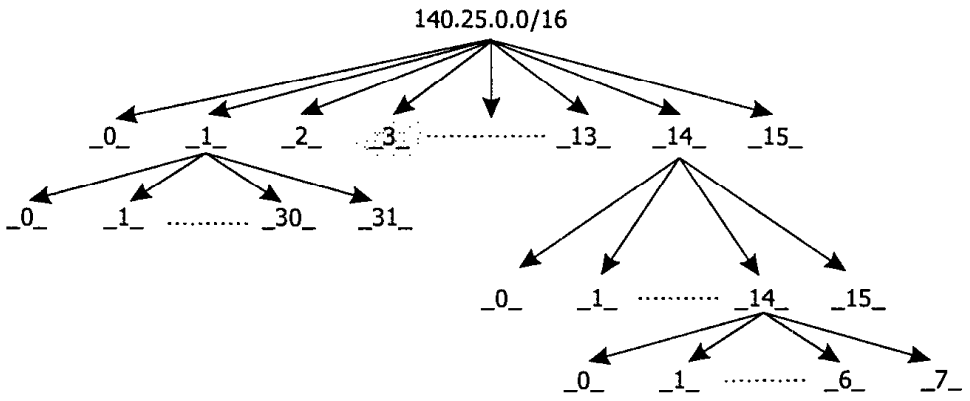


Рис. 2.19. Определение адресов устройств в подсети #3 (140.25.48.0/20)

Так как поле номера устройства в подсети #3 состоит из 12 бит, возможно 4 094 (2^{12} – 2) корректных адреса устройств. Устройства нумеруются от 1 до 4 094. Адреса устройств для подсети #3 приводятся ниже. Курсивом обозначается расширенный сетевой префикс, жирным начертанием — 12-разрядный номер устройства.

Подсеть #3:*10001100.00011001.0011*0000.00000000 = 140.25.48.0/20
Устройство #1:*10001100.00011001.*0011**0000.00000001** = 140.25.48.1/20
Устройство #2:*10001100.00011001.*0011**0000.00000010** = 140.25.48.2/20
Устройство #3:*10001100.00011001.*0011**0000.00000011** = 140.25.48.3/20
...
Устройство #4093: *10001100.00011001.*0011**1111.11111101** = 140.25.63.253/20
Устройство #4094: *10001100.00011001.*0011**1111.11111110** = 140.25.63.254/20

Широковещательный адрес для подсети #3 — это тот адрес, в котором все биты в поле номера устройства установлены в единицу, то есть *10001100.00011001.* 0011**1111.11111111** = 140.25.63.255.

Следует отметить, что широковещательный адрес для подсети #3 ровно на единицу меньше базового адреса для подсети #4 (140.25.64.0).

Определим подсети нижнего уровня (подсети подсетей) для подсети #14 (140.25.224.0/20). После того как основной сетевой адрес разделен на 16 подсетей, подсеть #14 разбивается на 16 адресных блоков равного размера (рис. 2.20).

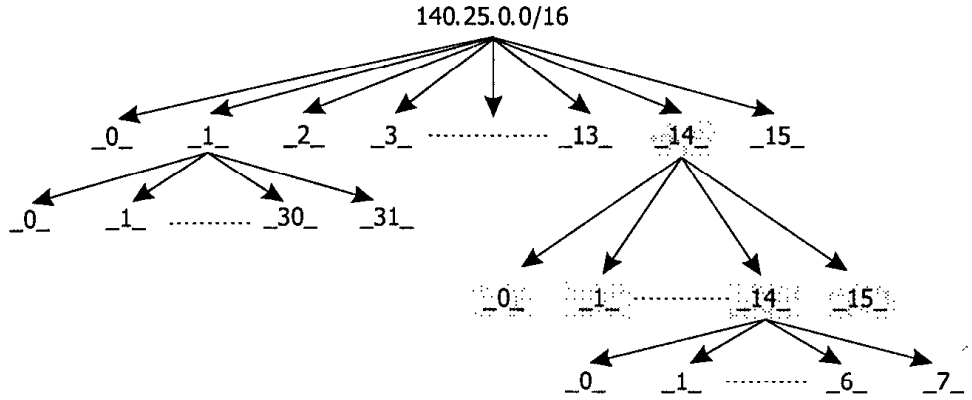


Рис. 2.20. Определение подсетей нижнего уровня для подсети #14 (140.25.224.0/20)

Так как 16 = 2^4, то для того, чтобы идентифицировать каждую из этих 16 подсетей, требуется еще 4 бита. Это означает, что потребуется расширенный сетевой префикс /24. Ниже приводятся 16 подсетей из адресного блока 140.25.224.0/20, которые нумеруются от 0 до 15. Курсивная часть каждого адреса подсети нижнего уровня идентифицирует расширенный сетевой префикс, жирное выделение соответствует 4-разрядному полю подсети нижнего уровня:

Подсеть #14:*10001100.00011001.1110*0000.00000000 = 140.25.224.0/20
Подсеть #14-0:*10001100.00011001.*1110**0000**.00000000 = 140.25.224.0/24
Подсеть #14-1:*10001100.00011001.*1110**0001**.00000000 = 140.25.225.0/24
Подсеть #14-2:*10001100.00011001.*1110**0010**.00000000 = 140.25.226.0/24
Подсеть #14-3:*10001100.00011001.*1110**0011**.00000000 = 140.25.227.0/24
Подсеть #14-4:*10001100.00011001.*1110**0100**.00000000 = 140.25.228.0/24
...
Подсеть #14-14:*10001100.00011001.*1110**1110**.00000000 = 140.25.238.0/24
Подсеть #14-15:*10001100.00011001.*1110**1111**.00000000 = 140.25.239.0/24

Теперь следует определить адреса, которые могут быть назначены устройствам в подсетях нижнего уровня, например #14-3 (140.25.227.0/24) (рис. 2.21).

В подсети нижнего уровня #14-3 можно использовать 8 бит для задания адресов устройств. Это означает, что каждая подсеть нижнего уровня #14-х способна поддерживать блок из 254 адресов устройств (2^8 – 2), которые нумеруются от 1 до 254. Адреса устройств в подсети #14-3 приводятся ниже. Курсивная часть каждого адреса идентифицирует расширенный сетевой префикс, жирным шрифтом выделен 8-разрядный номер устройства:

Подсеть #14-3:*10001100.00011001.11100011*.00000000 = 140.25.227.0/24
Устройство #1:*10001100.00011001.11100011*.**00000001** = 140.25.227.1/24
Устройство #2:*10001100.00011001.11100011*.**00000010** = 140.25.227.2/24

Устройство #3:*10001100.00011001.11100011*.**00000011** = 140.25.227.3/24
Устройство #4:*10001100.00011001.11100011*.**00000100** = 140.25.227.4/24
Устройство #5:*10001100.00011001.11100011*.**00000101** = 140.25.227.5/24
...
Устройство #253:*10001100.00011001.11100011*.**11111101** = 140.25.227.253/24
Устройство #254:*10001100.00011001.11100011*.**11111110** = 140.25.227.254/24

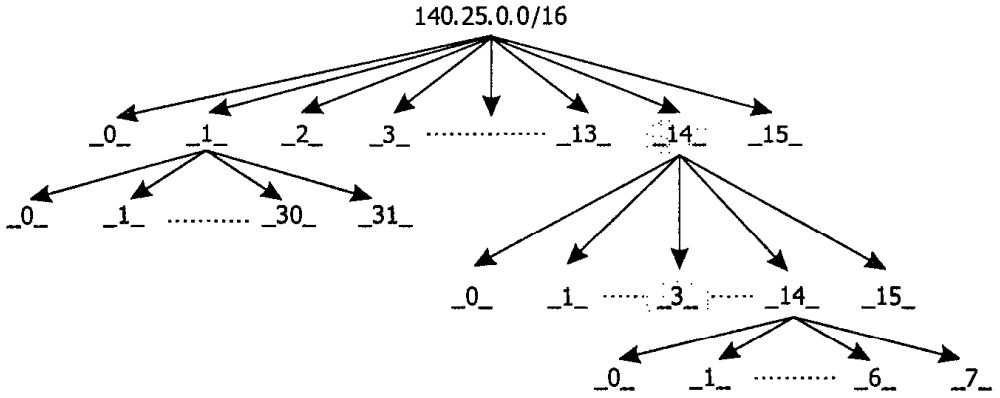


Рис. 2.21. Определение адресов устройств в подсети #14-3 (140.25.227.0/24)

Широковещательный адрес для подсети #14-3 — это тот, в котором все биты в поле номера устройства установлены в единицу: *10001100.00011001.11100011.* **11111111** = 140.25.227.255.

Чтобы лучше разобраться с назначением подсетей нижнего уровня, рассмотрим случай, когда в подсети нижнего уровня (в нашем случае — второго уровня) вводятся подсети.

После того как подсеть #14 была разделена на 16 подсетей нижнего уровня (под-подсетей), в под-подсети #14-14 (140.25.238.0/24) выделяются 8 адресных блоков равного размера (рис. 2.22).

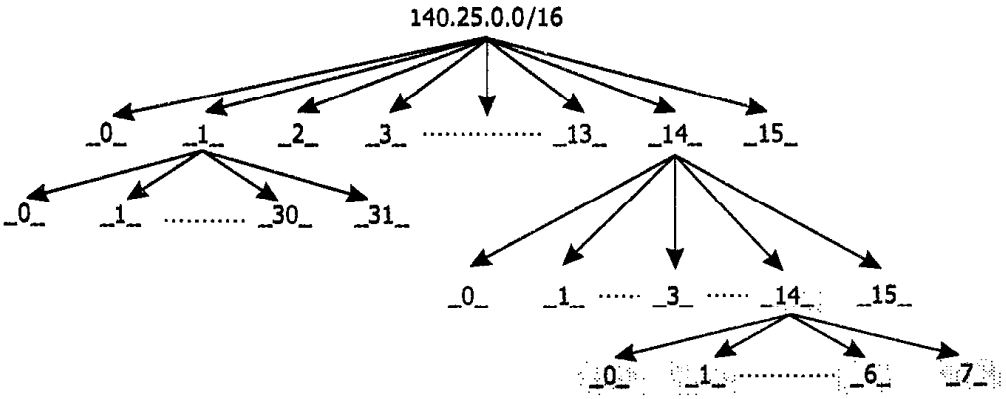


Рис. 2.22. Дальнейшая рекурсия в делении подсетей

Так как $8 = 2^3$, то для того, чтобы идентифицировать каждую из этих 8 подсетей, дополнительно требуется три бита. Это означает, что возникает потребность

в расширенном сетевом префиксе /27. Адреса 8 подсетей из адресного блока 140.25.238.0/24 приведены ниже. Подсети нумеруются от 0 до 7. Курсивная часть каждого адреса под-под-подсети идентифицирует расширенный сетевой префикс, в то время как жирным шрифтом показан 3-разрядный номер под-под-подсети:

Подсеть #14-14:*10001100.00011001.11101110*.00000000 = 140.25.238.0/24
Подсеть #14-14-0:*10001100.00011001.11101110*.**000**00000 = 140.25.238.0/27
Подсеть #14-14-1:*10001100.00011001.11101110*.**001**00000 = 140.25.238.32/27
Подсеть #14-14-2:*10001100.00011001.11101110*.**010**00000 = 140.25.238.64/27
Подсеть #14-14-3:*10001100.00011001.11101110*.**011**00000 = 140.25.238.96/27
Подсеть #14-14-4:*10001100.00011001.11101110*.**100**00000 = 140.25.238.128/27
Подсеть #14-14-5:*10001100.00011001.11101110*.**101**00000 = 140.25.238.160/27
Подсеть #14-14-6:*10001100.00011001.11101110*.**110**00000 = 140.25.238.192/27
Подсеть #14-14-7:*10001100.00011001.11101110*.**111**00000 = 140.25.238.224/27

Теперь можно определить адреса устройств в подсети #14-14-2 (140.25.238.64/27) (2.23).

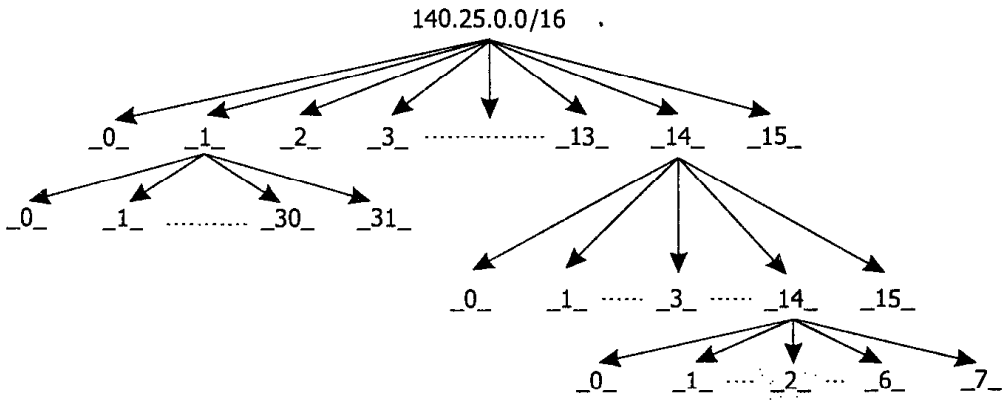


Рис. 2.23. Определение адресов устройств в подсети #14-14-2 (140.25.238.64/27)

Каждая из подсетей третьего уровня в подсети второго уровня #14-14 имеет пять битов для задания адресов устройств. Это означает, что каждая из этих подсетей может поддерживать до 30 адресов устройств ($2^5 - 2$), которые нумеруются от 1 до 30. Адреса устройств для подсети #14-14-2 приводятся ниже (жирным шрифтом выделен 5-разрядный номер устройства):

Подсеть #14-14-2:*10001100.00011001.11101110.010* 00000 = 140.25.238.64/27
Устройство #1:*10001100.00011001.11101110*.010**00001** = 140.25.238.65/27
Устройство #2:*10001100.00011001.11101110*.010**00010** = 140.25.238.66/27
Устройство #3:*10001100.00011001.11101110*.010**00011** = 140.25.238.67/27
Устройство #4:*10001100.00011001.11101110*.010**00100** = 140.25.238.68/27
Устройство #5:*10001100.00011001.11101110*.010**00101** = 140.25.238.69/27
...
Устройство #29:*10001100.00011001.11101110*.010**11101** = 140.25.238.93/27
Устройство #30:*10001100.00011001.11101110*.010**11110** = 140.25.238.94/27

Широковещательный адрес для подсети #1414-2 — тот, в котором все биты в поле номера устройства установлены в единицу: *10001100.00011001.11011100.* 01011111 = 140.25.238.95.

Широковещательный адрес для подсети #14-14-2 ровно на единицу меньше базового адреса подсети #14-14-3 (140.25.238.96).

Немного изменим ситуацию и предположим, что организации был выделен сетевой адрес 140.25.0.0/16, и администратор собирается использовать маски подсети переменной длины. На рис. 2.24 показана планируемая схема выделения подсетей для этой организации.

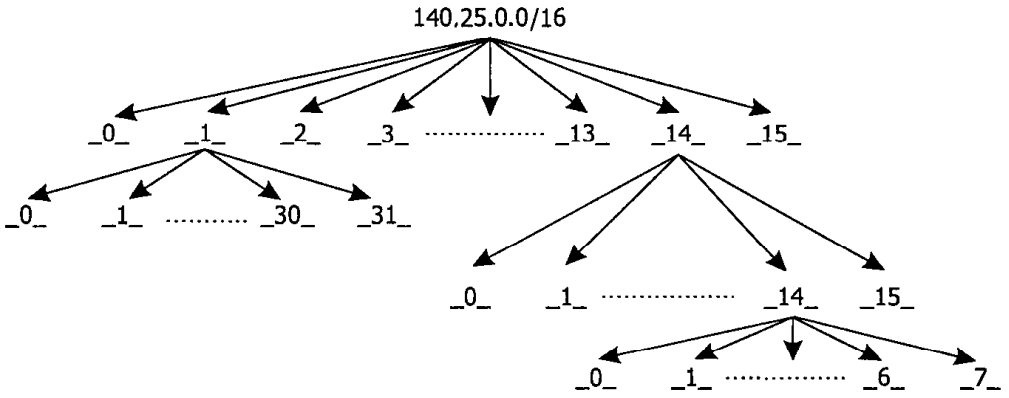


Рис. 2.24. Стратегия выделения подсетей

Первый шаг при организации подсетей состоит в делении основного сетевого адреса на 8 адресных блоков равного размера. Затем подсеть #1 делится на 32 одинаковых по длине адресных блока, а подсеть #6 также разбивается на 16 одинаковых адресных блоков. Далее получившиеся подсети нижнего уровня, например подсеть #6-14, делятся на 8 адресных блоков равного размера. Определим 8 подсетей сети с адресом 140.25.0.0/16:

Базовая сеть:*10001100.00011001*.00000000.00000000 = 140.25.0.0/16
Подсеть #0:*10001100.00011001*.**000**00000.00000000 = 140.25.0.0/19
Подсеть #1:*10001100.00011001*.**001**00000.00000000 = 140.25.32.0/19
Подсеть #2:*10001100.00011001*.**010**00000.00000000 = 140.25.64.0/19
Подсеть #3:*10001100.00011001*.**011**00000.00000000 = 140.25.96.0/19
Подсеть #4:*10001100.00011001*.**100**00000.00000000 = 140.25.128.0/19
Подсеть #5:*10001100.00011001*.**101**00000.00000000 = 140.25.160.0/19
Подсеть #6:*10001100.00011001*.**110**00000.00000000 = 140.25.192.0/19
Подсеть #7:*10001100.00011001*.**111**00000.00000000 = 140.25.224.0/19

Вот список адресов устройств, которые могут использоваться в подсети #3 (140.25.96.0):

Подсеть #3:*10001100.00011001.011*00000.00000000 = 140.25.96.0/19
Устройство #1:*10001100.00011001*.011**00000.00000001** = 140.25.96.1/19
Устройство #2:*10001100.00011001*.011**00000.00000010** = 140.25.96.2/19
Устройство #3:*10001100.00011001*.011**00000.00000011** = 140.25.96.3/19
...

Устройство #8189: *10001100.00011001.*011**11111.11111101** = 140.25.127.253/19
Устройство #8190: *10001100.00011001.*011**11111.11111110** = 140.25.127.254/19

Теперь определим широковещательный адрес для подсети #3 (140.25.96.0):
*10001100.00011001.*011**11111.11111111** = 140.25.127.255
Определим 16 подсетей нижнего уровня в подсети #6 (140.25.192.0/19):

Подсеть #6:*10001100.00011001.110*00000.00000000 = 140.25.192.0/19
Подсеть #6-0:*10001100.00011001.*110**00000**.00000000 = 140.25.192.0/23
Подсеть #6-1:*10001100.00011001.*110**00010**.00000000 = 140.25.194.0/23
Подсеть #6-2:*10001100.00011001.*110**00100**.00000000 = 140.25.196.0/23
Подсеть #6-3:*10001100.00011001.*110**00110**.00000000 = 140.25.198.0/23
Подсеть #6-4:*10001100.00011001.*110**01000**.00000000 = 140.25.200.0/23
...
Подсеть #6-14:*10001100.00011001.*110**11100**.00000000 = 140.25.220.0/23
Подсеть #6-15:*10001100.00011001.*110**11110**.00000000 = 140.25.222.0/23

Перечислим адреса устройств, которые могут использоваться в подсети нижнего уровня #6-3 (140.25.198.0/23):

Подсеть #6-3:*10001100.00011001.1100011*0.00000000 = 140.25.198.0/23
Устройство #1:*10001100.00011001.*1100011**0.00000001** = 140.25.198.1/23
Устройство #2:*10001100.00011001.*1100011**0.00000010** = 140.25.198.2/23
Устройство #3:*10001100.00011001.*1100011**0.00000011** = 140.25.198.3/23
Устройство #4:*10001100.00011001.*1100011**0.00000100** = 140.25.198.4/23
Устройство #5:*10001100.00011001.*1100011**0.00000110** = 140.25.198.5/23
...
Устройство #509:*10001100.00011001.*1100011**1.11111101** = 140.25.199.253/23
Устройство #510:*10001100.00011001.*1100011**1.11111110** = 140.25.199.254/23

Определим широковещательный адрес для подсети нижнего уровня #6-3 (140.25.198.0/23):
*10001100.00011001.*1100011**1.11111111** = 140.25.199.255
Определим 8 подсетей третьего уровня для подсети второго уровня #6-14 (140.25.220.0/23):

Подсеть #6-14:*10001100.00011001.1101110*0.00000000 = 140.25.220.0/23
Подсеть #6-14-0: *10001100.00011001.*1101110**0.00**000000 = 140.25.220.0/26
Подсеть #6-14-1: *10001100.00011001.*1101110**0.01**000000 = 140.25.220.64/26
Подсеть #6-14-2: *10001100.00011001.*1101110**0.10**000000 = 140.25.220.128/26
Подсеть #6-14-3: *10001100.00011001.*1101110**0.11**000000 = 140.25.220.192/26
Подсеть #6-14-4: *10001100.00011001.*1101110**1.00**000000 = 140.25.221.0/26
Подсеть #6-14-5: *10001100.00011001.*1101110**1.01**000000 = 140.25.221.64/26
Подсеть #6-14-6: *10001100.00011001.*1101110**1.10**000000 = 140.25.221.128/26
Подсеть #6-14-7: *10001100.00011001.*1101110**1.11**000000 = 140.25.221.192/26

Приведем список адресов устройств, которые могут использоваться в подсети #6-14-2 (140.25.220.128/26):

Подсеть #6-14-2:*10001100.00011001.11011100.10*000000 = 140.25.220.128/26
Устройство #1:*10001100.00011001.11011100.*10**000001** = 140.25.220.129/26

Устройство #2:*10001100.00011001.11011100.*10**000010** = 140.25.220.130/26
Устройство #3:*10001100.00011001.11011100.*10**000011** = 140.25.220.131/26
Устройство #4:*10001100.00011001.11011100.*10**000100** = 140.25.220.132/26
Устройство #5:*10001100.00011001.11011100.*10**000101** = 140.25.220.133/26
...
Устройство #61:*10001100.00011001.11011100.*10**111101** = 140.25.220.189/26
Устройство #62:*10001100.00011001.11011100.*10**111110** = 140.25.220.190/26

Определим широковещательный адрес для подсети третьего уровня #6-14-2 (140.25.220.128/26):

*10001100.00011001.11011100.*10**111111** = 140.25.220.191

Технология CIDR

Поддержка маршрутизаторами технологии наибольшего совпадения сводится к выполнению следующего правила: маршрут в таблице маршрутизации с наибольшим расширенным сетевым префиксом описывает меньший набор получателей, чем тот же маршрут с коротким расширенным сетевым префиксом. В результате при передаче трафика маршрутизатор должен выбирать маршрут с наибольшим расширенным сетевым префиксом.

Например, если адрес получателя равен 11.1.2.5 и в таблице маршрутизации существуют три маршрута (табл. 2.12), то маршрутизатор выберет маршрут #1, так как его расширенный сетевой префикс охватывает большее число битов в адресе получателя.

Таблица 2.12. Пример работы технологии наибольшего совпадения

Получатель	11.1.2.5	00001011.00000001.00000010.00000101
Маршрут #1	11.1.2.0/24	**00001011.00000001.00000010.**00000000
Маршрут #2	11.1.0.0/16	**00001011.00000001.**00000000.00000000
Маршрут #3	11.0.0.0/8	**00001011.**00000000.00000000.00000000

Необходимо сделать одно важное замечание. Так как адрес получателя совпадает с тремя маршрутами, он должен быть присвоен хосту, который подключен к подсети 11.1.2.0/24. Если адрес 11.1.2.5 присвоен хосту, который связан с другими подсетями, маршрутизатор не будет передавать трафик этому хосту, так как технология наибольшего совпадения предполагает, что последний является частью подсети 11.1.2.0. Наивысшего внимания требует присвоение адресов хостам с учетом особенностей работы технологии наибольшего совпадения.

Иерархическая маршрутизация, предусмотренная протоколом OSPF, требует, чтобы адреса, присвоенные хостам, отражали актуальную сетевую топологию. Это уменьшает количество маршрутной информации, так как набор адресов, назначенных подсетям региона, можно свести в одно сообщение об обновлении. Иерархическая маршрутизация позволяет выполнять это рекурсивно в различных точках внутри топологии маршрутизации. Если адреса не соответствуют

топологии, то нельзя выполнить обобщение адресной информации, и размер таблиц маршрутизации не будет уменьшаться. Этот постулат является основополагающим при рассмотрении технологии бесклассовой маршрутизации (Classless Inter-Domain Routing — CIDR).

Концепция бесклассовой междоменной маршрутизации была официально документирована в сентябре 1993 года (RFC 1517, RFC 1518, RFC 1519 и RFC 1520). Ее появление было инспирировано участившимися кризисами в Интернете. Из-за несовершенства протоколов маршрутизации трафик, вызванный сообщениями об обновлении таблиц маршрутизации, приводил к сбоям магистральных маршрутизаторов. Это было связано с полным задействованием их ресурсов на обработку большого объема служебной информации. Так, в 1994 году таблицы маршрутизации магистральных маршрутизаторов Интернета содержали до 70 000 маршрутов. Внедрение технологии CIDR сократило число записей маршрутов до 30 000. Кроме того, дополнительной предпосылкой для внедрения технологии CIDR явилась реальная опасность нехватки адресного пространства при дальнейшем расширении Интернета. Эта технология позволяет реализовать новые, не поддерживавшиеся ранее возможности.

1. Отход от традиционной концепции разделения адресов протокола IP на классы, благодаря чему можно более эффективно использовать адресное пространство протокола IP версии 4.
2. Объединение маршрутов. Здесь одна запись в таблице маршрутизации может представлять сотни адресов. Это позволяет контролировать количество маршрутной информации в магистральных маршрутизаторах Интернета.

Технология CIDR позволяет заменить традиционное использование классов адресов протокола IP на обобщенный сетевой префикс. Для определения границ между номером сети и номером хоста в IP-адресе маршрутизаторы выделяют сетевой префикс, вместо того чтобы по первым трем битам адреса определять его класс. В результате CIDR поддерживает организацию сетей произвольного размера, а не сетей со стандартными сетевыми номерами, которые ассоциируются с классами адресов.

В технологии CIDR каждая часть маршрутной информации рекламируется маршрутизаторами совместно с сетевым префиксом. Длина сетевого префикса помогает определить число старших битов, соответствующих номеру сети в записи таблицы маршрутизации.

Например, адрес подсети в таблице маршрутизации с номером сети, занимающим 20 бит, и номером хоста размером 12 бит, будет записан с сетевым префиксом длиной 20 бит, что можно представить как /20. Удобство заключается в том, что рекламируемый маршрутизатором IP-адрес подсети с префиксом /20 может быть адресом любого класса (A, B или C). Маршрутизаторы с поддержкой технологии CIDR не проверяют обычными методами класс адреса — вместо этого они полагаются на информацию о сетевом префиксе, поступившую с рекламируемым маршрутом.

Если отказаться от разделения адресов на классы, то сетевой префикс можно рассматривать как непрерывный битовый блок в адресном пространстве протокола IP. Например, рассмотренный выше сетевой префикс /20 предоставляет такое же количество битов для задания адресов хостов, как может быть получено

при разделении адресов на классы, а именно 12 битов. Это обеспечивает поддержку до 4094 ($2^{12} - 2 = 4094$) адресов хостов. Таблица 2.13 показывает пример использования сетевого префикса /20.

Таблица 2.13. Пример использования сетевого префикса /20

Класс A	10.23.64.0/20	**00001010.00010111.0100**0000.00000000
Класс B	130.5.0.0/20	**10000010.00000101.0000**0000.00000000
Класс C	200.7.128.0/20	**11001000.00000111.1000**0000.00000000

Ввиду того что многие хосты учитывают принадлежность адресов к определенному классу, при их настройке требуется задавать маску подсети. Если администратор вместо маски подсети укажет сетевой префикс, хост не будет его воспринимать.

Например, проблема возникает в случае, когда необходим адрес 200.25.16.0 с сетевым префиксом /20 для поддержки 4094 хостов ($2^{12} - 2 = 4094$), так как хосты, не поддерживающие технологию CIDR, будут интерпретировать заданный адрес как адрес класса C с маской 255.255.255.0. При этом битов, оставшихся в поле номера хоста, не хватит для задания требуемого количества адресов хостов. Если хосты «понимают» технологию CIDR, то для данного адреса подойдет любой сетевой префикс.

Отметим, что технология CIDR в настоящее время встроена в магистральные интернет-маршрутизаторы с протоколом BGP4, а обычные хосты в локальных сетях ее не поддерживают.

Технология CIDR позволяет более эффективно использовать адресное пространство протокола IP. Обычно интернет-провайдер выделяет своим клиентам адреса определенных классов, что приводит к некоторой избыточности в одном месте и к дефициту в другом. Обратившись к технологии CIDR, провайдеры получают возможность «нарезать» блоки из выделенного им адресного пространства, которые оптимально подходят под требования каждого клиента, оставляя в то же время возможность его будущего беспроблемного роста.

Предположим, что провайдеру был выделен адрес 206.0.64.0 с сетевым префиксом /18. Тогда для задания индивидуальных адресов остается 14 бит, что позволяет поддерживать до 16 384 (2^{14}) хостов. В случае использования классов адресов достижение той же цели требует от провайдера выделения 64 адресов класса C.

Если клиенту, которого обслуживает данный провайдер, для своей сети требуется 800 адресов хостов, то провайдер может выделить ему адресный блок 206.0.68.0/22, то есть блок из 1024 (2^{10}) адресов хостов. При таком выделении клиент получает в свое распоряжение 224 дополнительных адреса. Если следовать классовой адресной схеме, то клиенту потребовалось бы выделить или один адрес класса B, или четыре адреса класса C. При предоставлении одного адреса класса B клиент получает более 64 000 дополнительных адресов. При выделении четырех адресов класса C клиент имеет приемлемое количество адресов, но заодно увеличивается размер таблиц маршрутизации (добавляются четыре записи). Таблица 2.14 иллюстрирует эти рассуждения.

Таблица 2.14. Пример выделения адресов клиенту

Блок адресов провайдера	206.0.64.0/18	**11001110.00000000.01**000000.00000000
Блок адресов клиента	206.0.68.0/22	**11001110.00000000.010001**00.00000000
Или		
Адрес класса C #0	206.0.68.0/24	11001110.00000000.010001**00**.00000000
Адрес класса C #1	206.0.69.0/24	11001110.00000000.010001**01**.00000000
Адрес класса C #2	206.0.70.0/24	11001110.00000000.010001**10**.00000000
Адрес класса C #3	206.0.71.0/24	11001110.00000000.010001**11**.00000000

Рассмотрим еще один пример. Предположим, что провайдеру был выделен адресный блок 200.25.0.0/16. Нетрудно подсчитать, что этот адресный блок поддерживает до 65 536 (2^{16} = 65 536) индивидуальных адресов хостов. Из данного блока провайдер хочет выделить адресный блок 200.25.16.0/20 в расчете на 4096 (2^{12} = 4096) адресов. Опираясь на классовую схему, провайдер будет вынужден использовать для этой цели 16 адресов класса C (табл. 2.15).

Таблица 2.15. Пример выделения адресного пространства

Сеть #0	200.25.16.0/24	11001000.00011001.0001**0000**.00000000
Сеть #1	200.25.17.0/24	11001000.00011001.0001**0001**.00000000
Сеть #2	200.25.18.0/24	11001000.00011001.0001**0010**.00000000
...		
Сеть #14	200.25.30.0/24	11001000.00011001.000**11101**.00000000
Сеть #15	200.25.31.0/24	11001000.00011001.000**11111**.00000000

Продемонстрировать различие между адресацией на основе классов и на основе технологии CIDR наглядно позволяют круговые диаграммы. В первом случае пример, расписанный в табл. 2.15, можно изобразить в виде круга, разделенного на 16 одинаковых секторов (рис. 2.25). Каждый сектор соответствует одной сети класса C. Изменение в классе адресов в состоянии привести к изменению количества секторов. Секторы в любом случае имеют одинаковые размеры.

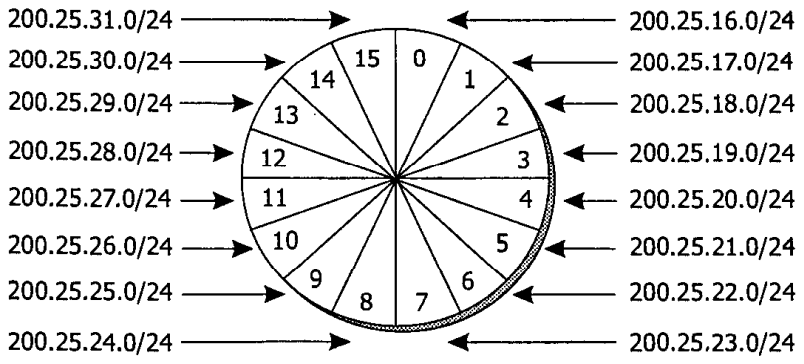


Рис. 2.25. Диаграмма деления адресного пространства

При использовании технологии CIDR провайдер имеет возможность «нарезать» адресное пространство на секторы произвольного размера. Предположим, что провайдер обслуживает четыре организации: A, B, C и D. Организация A претендует на половину всего адресного пространства провайдера. Организации B необходима четверть, а организациям C и D требуется по одной восьмой от всего диапазона.

Процесс выделения адресного пространства провайдер может провести за три этапа. На первом шаге адресный блок провайдера 200.25.16.0/20 разделяется на две равные части. Каждая из частей включает в себя 2048 (2^{11} = 2048) возможных адресов хостов (табл. 2.16).

Таблица 2.16. Выделение адресного пространства для организации A

Блок адресов провайдера	200.25.16.0/20	**11001000.00011001.0001**0000.00000000
Организация A	200.25.16.0/21	**11001000.00011001.00010**001.00000000
Оставшийся резервный блок	200.25.24.0/21	**11001000.00011001.00011**010.00000000

Сетевой префикс /21 получен следующим образом. Провайдер имеет 4096 исходных адресов хостов. При делении адресного пространства пополам организация A получает 2048 адресов. Для поддержания этих адресов требуется 11-разрядное поле номера хоста. В результате сетевой префикс получается равным /21 (32 – 11 = 21).

На втором шаге оставшийся резервный блок разбивается на две равные части. Каждая из этих половинок представляет одну четверть всего адресного пространства провайдера и поддерживает до 1024 (2^{10} = 1024) адресов хостов. Сетевой префикс получается равным /22 (32 – 10 = 22). Выделение адресного пространства организации B показано в табл. 2.17.

Таблица 2.17. Выделение адресного пространства для организации B

Оставшийся после шага 1 резерв	200.25.24.0/21	**11001000.00011001.00011**000.00000000
Организация B	200.25.24.0/22	**11001000.00011001.00011**000.00000000
Оставшийся резервный блок	200.25.28.0/22	**11001000.00011001.00011**100.00000000

На третьем шаге образовавшийся резервный блок также разбивается на две равные части. Каждая из частей занимает одну восьмую от всего адресного пространства провайдера и поддерживает до 512 (2^9 = 512) адресов хостов. Сетевой префикс для адресов организаций C и D получается равным /23 (32 – 9 = 23). Таблица 2.18 показывает распределение адресного пространства для организаций C и D.

Таблица 2.18. Выделение адресного пространства организациям C и D

Оставшийся после шага 2 резерв	200.25.28.0/22	**11001000.00011001.00011**100.00000000
Организация C	200.25.28.0/23	**11001000.00011001.00011**100.00000000
Организация D	200.25.30.0/23	**11001000.00011001.00011**110.00000000

Круговая диаграмма на рис. 2.26 иллюстрирует разделение адресного пространства провайдера между всеми четырьмя организациями.

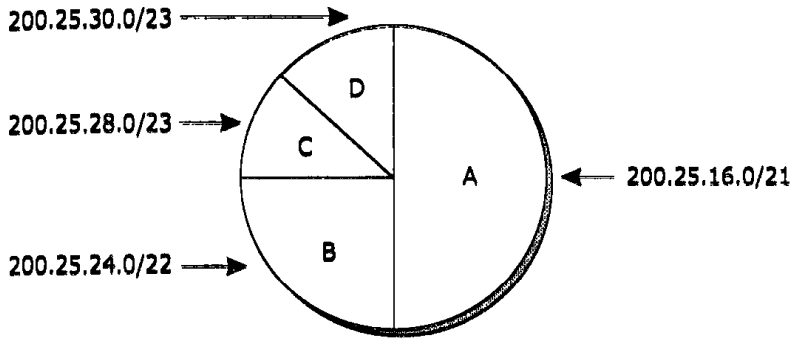


Рис. 2.26. Разделение адресного пространства на основе технологии CIDR

Необходимо отметить, что рассмотренные выше примеры работы организаций с провайдерами несколько оторваны от реальной жизни. Это связано с тем, что в настоящее время получить сеть класса B практически невозможно, а связь маршрутизаторов провайдера и организаций осуществляется при помощи статических таблиц маршрутизации, настроенных вручную. Технология CIDR (и протокол BGP4) может быть использована, если организация подключается к Интернету через нескольких разных провайдеров, а адреса ей выделяет InterNIC. Однако если организация имеет свою собственную крупную распределенную сеть, то применение технологии CIDR в ее центре может быть хорошим решением.

Для поддержания работоспособности технологии CIDR необходимо выполнение трех основных условий.

1. Протокол маршрутизации должен в своих служебных сообщениях передавать дополнительную информацию о сетевом префиксе.
2. Все маршрутизаторы должны поддерживать алгоритм передачи, основанный на технологии наибольшего совпадения.
3. Чтобы обеспечить возможность объединения маршрутов, адреса нужно назначить в соответствии с существующей сетевой топологией.

Другим важным достоинством технологии CIDR является возможность контролировать размеры таблиц маршрутизации в Интернете. Для уменьшения размеров маршрутной информации необходимо разделение Интернета на адресные домены. Внутри каждого домена циркулирует только внутренняя маршрутная информация о всех сетях в домене. Однако вне домена рекламируется только общий сетевой префикс. Это позволяет одной записи в таблице маршрутизации указывать маршрут во множество индивидуальных сетей.

Необходимо учитывать, что объединение маршрутов не выполняется автоматически. Администратор должен настроить каждый маршрутизатор для выполнения объединения маршрутов. При этом важно помнить, что технология CIDR является частью нового протокола маршрутизации BGP-4. Успешное внедрение технологии CIDR позволит увеличить число индивидуальных сетей, подключенных к Интернету, с сохранением размеров таблиц маршрутизации.

Технологии CIDR и маски подсети переменной длины позволяют рекурсивно делить порции адресного пространства на небольшие части. Основное различие между ними в том, что при использовании маски подсети переменной длины рекурсия выполняется на адресном пространстве, выделенном организации ранее. При этом схема деления пространства остается спрятана внутри организации (то есть, например, для пользователей в сети Интернет подобная схема разделения остается прозрачной и незаметной).

Пример использования технологии CIDR

Рассмотрим на практике работу с технологией CIDR. Для этого выберем адресный блок 200.56.168.0/21. На первом этапе запишем индивидуальные номера сетей для этого блока.

В двоичном виде блок представляется следующим образом:

200.56.168.0/21 — **11001000.00111000.10101**000.00000000.

Маска, определяемая префиксом /21, на три бита меньше, чем традиционная маска для сети класса С (/24). А это означает, что рассматриваемый блок CIDR идентифицирует блок из 8 (2^3) последовательных (упорядоченных) сетей с префиксом /24:

Сеть #0: 11001000.00111000.10101**000**.xxxxxxxx — 200.56.168.0
Сеть #1: 11001000.00111000.10101**001**.xxxxxxxx — 200.56.169.0
Сеть #2: 11001000.00111000.10101**010**.xxxxxxxx — 200.56.170.0
Сеть #3: 11001000.00111000.10101**011**.xxxxxxxx — 200.56.171.0
Сеть #4: 11001000.00111000.10101**100**.xxxxxxxx — 200.56.172.0
Сеть #5: 11001000.00111000.10101**101**.xxxxxxxx — 200.56.173.0
Сеть #6: 11001000.00111000.10101**110**.xxxxxxxx — 200.56.174.0
Сеть #7: 11001000.00111000.10101**111**.xxxxxxxx — 200.56.175.0

Другой пример. Приведем индивидуальные номера сетей, определяемых блоком CIDR 195.24.0/13. Указанный блок CIDR можно записать в двоичном представлении:

195.24.0.0/13 — **11000011.00011**000.00000000.00000000

Маска, определяемая префиксом /13, на 11 бит короче, чем традиционная маска для сети класса С. В рассматриваемом примере это указывает на то, что блок CIDR идентифицирует 2048 (2^{11}) последовательных сетей с сетевым префиксом /24:

Сеть #0:11000011.00011**000.00000000**.xxxxxxxx — 195.24.0.0
Сеть #1:11000011.00011**000.00000001**.xxxxxxxx — 195.24.1.0
Сеть #2:11000011.00011**000.00000010**.xxxxxxxx — 195.24.2.0
...
Сеть #2045:11000011.00011**111.11111101**.xxxxxxxx — 195.31.253.0
Сеть #2046:11000011.00011**111.11111110**.xxxxxxxx — 195.31.254.0
Сеть #2047:11000011.00011**111.11111111**.xxxxxxxx — 195.31.255.0

В третьем примере требуется объединить набор следующих четырех адресов с префиксом /24 в максимально вместительный адрес:

212.56.132.0/24
212.56.133.0/24
212.56.134.0/24
212.56.135.0/24

На первом этапе запишем каждый адрес сети в двоичном виде, что поможет в дальнейшем:

212.56.132.0/24 11010100.00111000.10000**10**0.00000000
212.56.133.0/24 11010100.00111000.10000**10**1.00000000
212.56.134.0/24 11010100.00111000.10000**11**0.00000000
212.56.135.0/24 11010100.00111000.10000**11**1.00000000

Общий префикс для всех адресов — **11010100.00111000.10000**100.00000000.
Объединение CIDR будет следующим — 212.56.132.0/22.
В четвертом примере объединим набор из других четырех сетей:

212.56.146.0/24
212.56.147.0/24
212.56.148.0/24
212.56.149.0/24

Запишем эти адреса в двоичной форме:

212.56.146.0/24 11010100.00111000.1001001**0**.00000000
212.56.147.0/24 11010100.00111000.1001001**1**.00000000
212.56.148.0/24 11010100.00111000.1001010**0**.00000000
212.56.148.0/24 11010100.00111000.1001010**1**.00000000

Следует обратить внимание на то, что рассматриваемый набор из четырех сетей нельзя объединить в адрес с префиксом /23:

212.56.146.0/23 11010100.00111000.**1001001**0.00000000
212.56.148.0/23 11010100.00111000.**1001010**0.00000000

Учитывая это, объединение CIDR будет следующим:

212.56.146.0/23
212.56.148.0/23

Если две сети с префиксом /23 нужно объединить в сеть с префиксом /22, то они должны соответственно «помещаться» в один адресный блок /22. Однако раз каждая из полученных сетей с префиксом /23 является членом различных блоков /22, то они не могут быть объединены в блоке /22, хотя объединение в 222.56.144/21 формально действительно. Но такое объединение включало бы четыре адреса сети, которых не было в исходных сетях. Учитывая все вышеизложенное, отметим, что объединение, состоящее из минимального количества адресных блоков (два) будет иметь префикс /23.

Завершим практикум еще один примером — объединим набор из 64 адресов сетей класса C в единый блок:

202.1.96.0/24
202.1.97.0/24
202.1.98.0/24
202.1.126.0/24
202.1.127.0/24
202.1.128.0/24
202.1.129.0/24
202.1.158.0/24
202.1.159.0/24

Запишем эти адреса в двоичном виде для выявления общего префикса:

202.1.96.0/24 11001010.00000001.01**100000**.00000000
202.1.97.0/24 11001010.00000001.01**100001**.00000000
202.1.98.0/24 11001010.00000001.01**100010**.00000000
202.1.126.0/24 11001010.00000001.01**111110**.00000000
202.1.127.0/24 11001010.00000001.01**111111**.00000000
202.1.128.0/24 11001010.00000001.10**000000**.00000000
202.1.129.0/24 11001010.00000001.10**000001**.00000000
202.1.158.0/24 11001010.00000001.10**011110**.00000000
202.1.159.0/24 11001010.00000001.10**011111**.00000000

Следует обратить внимание на то, что все 64 адреса сетей не могут быть объединены в блок с префиксом /19:

202.1.96.0/19 11001010.00000001.**01**100000.00000000
202.1.128.0/19 11001010.00000001.**10**000000.00000000

Учитывая это, объединение CIDR будет следующим:

202.1.96.0/19
202.1.128.0/19

По аналогии с предыдущим примером, если два адресных блока /19 нужно скомпоновать в блоке /18, то они должны «помещаться» в один блок /18. Так как каждый из рассматриваемых блоков входит в разные адресные блоки /18, то они, соответственно, не могут быть объединены в блоке /18, хотя возможно их слияние в один 202.1.0.0 /16. Однако такое объединение будет включать 192 дополнительных адреса сетей, которые не являются частью исходных сетей.

Настройка протокола IP на маршрутизаторе Cisco Systems

Базовым шагом при настройке протокола IP на маршрутизаторах является назначение IP-адреса интерфейсу. После этого интерфейс может участвовать в обмене информацией в распределенной сети. Интерфейс маршрутизатора вправе

иметь только один первичный (primary) адрес, для настройки которого следует воспользоваться командой ip address, выполняемой в режиме конфигурирования интерфейсов. В качестве параметра этой команды следует указать собственно IP-адрес и маску подсети.

```
Cisco1601#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco1601(config)#interface ethernet 0
Cisco1601(config-if)#ip address 192.168.3.254 255.255.255.0
```

Помимо первичного адреса администратор волен настроить неограниченное количество вторичных адресов (secondary), которые могут потребоваться для различных целей. Например, вероятна ситуация, когда выбранная схема выделения подсетей позволяет поддерживать не более 254 устройств в одной логической подсети, в то время как в физической подсети необходимо существование порядка 300 устройств. В этом случае администратор может настроить на маршрутизаторе вторичный адрес, что позволит получить две логические подсети в одном физическом сегменте сети. Следует отметить, что если любой маршрутизатор в сегменте поддерживает вторичные адреса протокола IP, то все другие маршрутизаторы в этом сегменте также должны использовать вторичные адреса с теми же адресами сети или подсети. Для настройки вторичных адресов служит команда ip address <маска IP-адреса> secondary.

```
...
Cisco1601(config)#interface ethernet 0
Cisco1601(config-if)#ip address 192.168.3.205 255.255.255.0 secondary
```

В приведенном выше примере на интерфейсе Ethernet0 маршрутизатора настраивается вторичный адрес протокола IP. До этого такой же командой (только без ключевого слова secondary) был настроен первичный адрес. Таким образом, маршрутизатор (точнее, один из его интерфейсов) имеет два адреса. Если теперь выполнить команду show running-config, то можно увидеть настроенные на интерфейсе адреса.

```
Cisco1601#show running-config
...
interface Ethernet0
   description connected to Internal LAN - 192.168.3.x
   ip address 192.168.3.205 255.255.255.0 secondary
   ip address 192.168.3.254 255.255.255.0
   no keepalive
...
```

В том случае, если для настройки маршрутизатора используется программное обеспечение Cisco ConfigMaker, при создании схемы сети администратору потребуется ввести адреса для интерфейсов маршрутизатора (рис. 2.27). При нажатии кнопки IP... становится доступен калькулятор.

Обычно канал связи между маршрутизаторами требует организации отдельной подсети. Однако такое требование достаточно проблематично удовлетворить, так как впустую расходуется множество возможных адресов устройств. Оптимальным может быть решение, которое заключается в использовании технологии VLSM

и создании небольших, содержащих всего два устройства подсетей (маска подсети /30, или 255.255.255.252), или в востребовании специального механизма операционной системы Cisco IOS, который называется *ip unnumbered*. В последнем варианте каналу связи между маршрутизаторами вообще не присваивается адрес, что позволяет говорить об экономии адресного пространства (к тому же это значительно упрощает администрирование сети). Данная возможность применима к каналам связи типа «точка-точка» между парами маршрутизаторов, примером чему служат выделенные каналы связи, постоянные виртуальные соединения Frame Relay, ATM и т. п. При этом происходит заимствование адреса сетевого уровня у одного из интерфейсов маршрутизатора (например, Ethernet0).

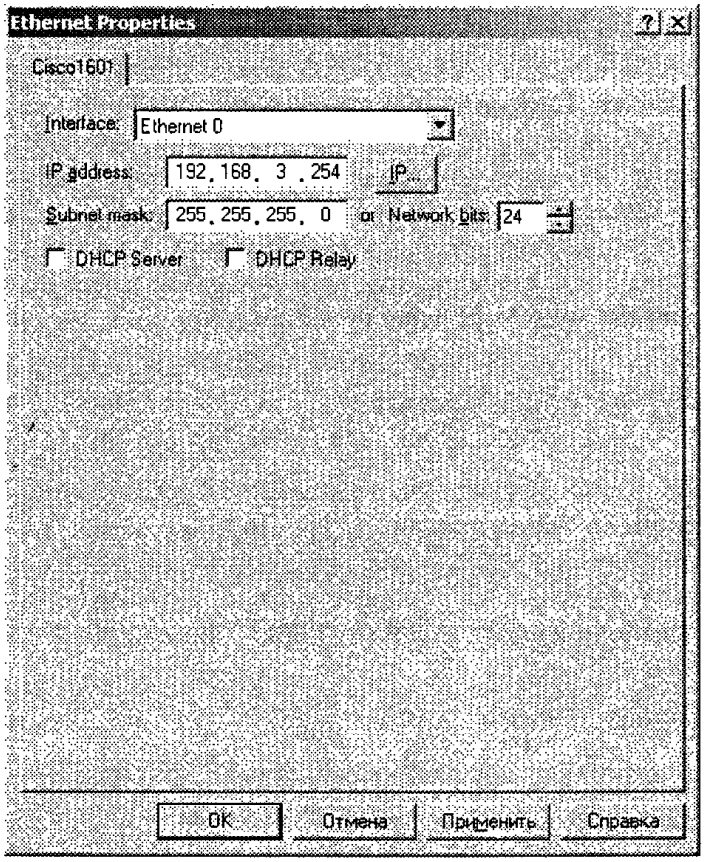


Рис. 2.27. Использование Cisco ConfigMaker для назначения адреса

Для того чтобы настроить рассматриваемый механизм, нужно выполнить команду ip unnumbered в режиме конфигурирования интерфейсов. Приведенный ниже пример показывает настройку подинтерфейса Serial0.1 (отражающий постоянное виртуальное соединение Frame Relay, глава 8) как ненумерованного с привлечением адреса интерфейса локальной сети.

```
Cisco805#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco805(config)#interface serial 0.1
Cisco805(config-subif)#ip unnumbered Ethernet0
```

Если для настройки сети используется Cisco ConfigMaker, то администратор сети также может использовать ip unnumbered. На рис. 2.28 показано диалоговое окно, в котором производится настройка соединения двух маршрутизаторов по технологии Frame Relay. Точнее, показано окно мастера настройки, где администратор может выбрать механизм ip unnumbered и указать, адрес какого интерфейса следует задействовать.

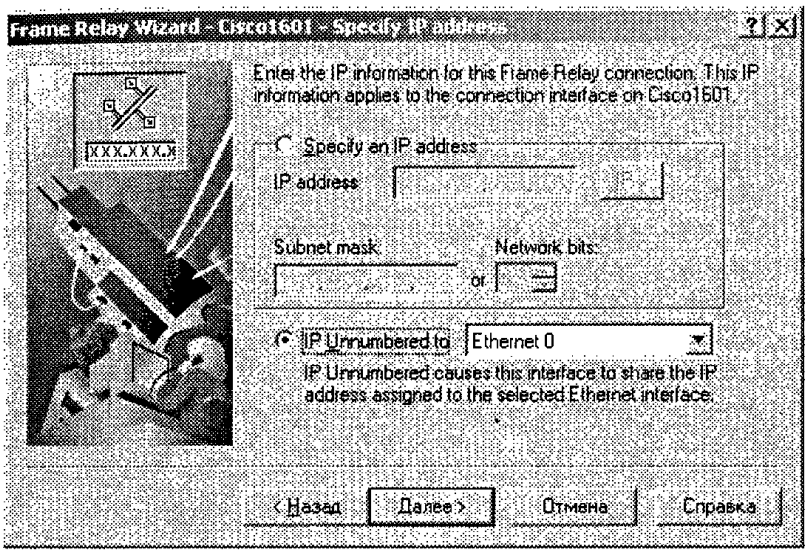


Рис. 2.28. Настройка ip unnumbered в программе Cisco ConfigMaker

Для ненумерованных интерфейсов следует учитывать ряд важных моментов. Интерфейсы, инкапсулирующие данные по типу HDLC, PPP, LAPB и Frame Relay, могут быть ненумерованными. Кроме того, при инкапсуляции Frame Relay это могут быть только интерфейсы типа «точка-точка». Невозможно использовать рассматриваемый механизм при инкапсуляции X.25 и SMDS. Другим ограничением является то, что администратор сети не вправе воспользоваться командой ping для определения статуса интерфейса. Выполняя команду show interfaces, можно получить информацию о том, что интерфейс не нумерован, и узнать, какой адрес ему назначен.

```
Cisco1005 #show interfaces serial 0.1
Serial0.1 is up, line protocol is up
   Hardware is QUICC Serial
   Interface is unnumbered. Using address of Ethernet0 (192.168.6.254)
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation FRAME-RELAY
```

Протокол ARP

Протокол ARP (Address Resolution Protocol) — это протокол разрешения адресов. Он описан в документе RFC 826. Необходимость в протоколе ARP продиктована тем обстоятельством, что IP-адреса устройств в сети назначаются независимо

от их физических адресов. Поэтому для доставки сообщений по сети требуется установить соответствие между физическим адресом устройства и его IP-адресом. Эта процедура называется разрешением адресов (address resolution). В большинстве случаев прикладные программы работают именно с IP-адресами. А так как схемы физической адресации устройств весьма разнообразны, то необходим специальный, универсальный протокол.

Для того чтобы получить адрес протокола IP, зная физический адрес устройства, предусмотрена процедура, называемая обратным разрешением адресов (reverse address resolution). Программное обеспечение маршрутизаторов Cisco IOS поддерживает следующие протоколы, связанные с процедурами разрешения адресов: протокол ARP, proxy ARP, reverse ARP, — которые описаны в документах RFC 826, 1027 и 903. Кроме того, поддерживается протокол Probe, разработанный компанией Hewlett-Packard для использования в сетях IEEE-802.3.

Reverse ARP (или обратное разрешение) работает аналогично протоколу ARP за исключением того, что в его задачи входит определение физического адреса по известному адресу сетевого уровня. Этот протокол требует наличия в сети сервера RARP, подключенного к тому же сегменту сети, что и интерфейс маршрутизатора. Наиболее часто протокол reverse ARP используется для запуска бездисковых рабочих станций. Маршрутизатор, работающий под управлением операционной системы Cisco IOS, примет во внимание этот протокол, если в процессе своей загрузки он обнаружит, что интерфейсу не назначен адрес протокола IP. Отметим тот факт, что маршрутизатор может работать как сервер RARP, отвечая на запросы клиентов.

Разрешение адресов может быть выполнено двумя способами: с помощью прямого отображения и с помощью динамического связывания (dynamic mapping). Протокол ARP задействует механизм динамического связывания. Функциональность протокола ARP сводится к решению двух задач. Одна часть протокола определяет физические адреса при посылке дейтаграммы, другая отвечает на запросы устройств в сети. Протокол ARP предполагает, что каждое устройство «знает» как свой IP-адрес, так и свой физический адрес.

Для того чтобы уменьшить количество посылаемых запросов ARP, каждое устройство в сети, использующее протокол ARP, должно иметь специальную буферную память. В ней хранятся пары адресов (IP-адрес, физический адрес) устройств в сети. Всякий раз, когда устройство получает ARP-ответ, оно сохраняет в буферной памяти соответствующую пару. Если адрес есть в списке пар, то нет необходимости посылать ARP-запрос. Эта буферная память называется *ARP-таблицей*.

В ARP-таблице могут содержаться как статические, так и динамические записи. Динамические записи добавляются и удаляются автоматически, статические заносятся вручную.

Так как большинство устройств в сети поддерживает динамическое разрешение адресов, то администратору, как правило, нет необходимости собственноручно указывать записи протокола ARP в таблице адресов.

Кроме того, ARP-таблица всегда содержит запись с физическим широковещательным адресом (0xFFFFFFFFFFFF) для локальной сети. Эта запись позволяет устройству принимать широковещательные ARP-запросы. Каждая запись в ARP-таблице имеет свое время жизни, например для операционной системы

Microsoft Windows NT оно составляет 10 минут. При добавлении записи для нее активируется таймер. Если запись не востребована в первые две минуты, она удаляется. Если используется — будет существовать на протяжении 10 минут. В некоторых реализациях протокола ARP новый таймер устанавливается после каждого обращения к записи в ARP-таблице.

Для просмотра содержимого таблицы на маршрутизаторе предусмотрена команда **show arp**, выполняемая в пользовательском режиме. Если необходимо очистить таблицу, следует воспользоваться командой **clear arp-cache**, доступной в привилегированном режиме.

```
Cisco1601>show arp
Protocol  Address          Age (min) Hardware Addr   Type   Interface
Internet  192.168.3.113    16        0090.2751.c4f0  ARPA   Ethernet0
Internet  192.168.3.14     3         0090.27e0.041e  ARPA   Ethernet0
Internet  192.168.3.3      79        0090.271d.c8ee  ARPA   Ethernet0
Internet  192.168.3.5      9         00a0.c9da.f6f6  ARPA   Ethernet0
Internet  192.168.3.4      5         00a0.c928.2d07  ARPA   Ethernet0
...
```

Когда требуется изменить время жизни записей в таблице, можно обратиться к команде **arp timeout**, выполняемой в режиме конфигурирования интерфейсов. В качестве параметра указывается временной интервал в секундах.

По умолчанию на маршрутизаторах применяется стандартный тип инкапсуляции протокола ARP, который обозначается ключевым словом **ARPA**. Администратор сети имеет право изменить метод инкапсуляции на SNAP или HP Probe исходя из требований конкретной сети. Для изменения метода инкапсуляции следует выполнить команду **arp** в режиме конфигурирования интерфейсов, добавляя к ней параметры **arpa**, **probe**, **snap**. Получить информацию о текущем методе инкапсуляции можно с помощью команды **show interfaces**.

```
Cisco1601#show interfaces ethernet 0
Ethernet0 is up, line protocol is up
  Hardware is QUICC Ethernet, address is 0010.7bef.4478 (bia 0010.7bef.4478)
  Description: connected to Internal LAN - 192.168.3.x
  Internet address is 192.168.3.254/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive not set
  ARP type: ARPA, ARP Timeout 04:00:00
```

Протокол proxy ARP маршрутизатор использует (как это определено в документе RFC 1027) для того, чтобы помочь устройствам, не имеющим информации о маршрутизации, определить физические адреса устройств, находящихся в другой подсети. Например, если маршрутизатор получает широковещательный запрос протокола ARP для устройства, которое расположено в сети, отличной от сети отправителя запроса, и если этот маршрутизатор знает все маршруты к этому устройству через другие свои интерфейсы, он генерирует ответ ARP с указанием физического адреса своего интерфейса. Затем устройство, пославшее запрос ARP, будет адресовать свои пакеты маршрутизатору, который и передаст их действительному получателю. Данный механизм посредничества включен по

умолчанию, и если его необходимо отключить, то следует воспользоваться ко-
мандой no ip proxy-arp, выполняемой в режиме конфигурирования интерфейсов.
Просмотреть результат работы этого механизма можно с помощью команды
show ip interface.

```
Cisco1601#show ip interface ethernet 0
Ethernet0 is up, line protocol is up
   Internet address is 192.168.3.254/24
   Broadcast address is 255.255.255.255
...
   Proxy ARP is enabled
```

Сообщения протокола ARP при передаче по сети инкапсулируются в поле данных кадра. Они не содержат IP-заголовка. В отличие от сообщений большинства протоколов, сообщения ARP не имеют фиксированного формата заголовка. Это объясняется тем, что протокол был разработан таким образом, чтобы он был применим для разрешения адресов в различных сетях. Фактически протокол способен работать с произвольными физическими адресами и сетевыми протоколами.

Рисунки 2.29 и 2.30 содержат результаты работы анализатора пакетов, которые наглядно показывают структуру запросов и ответов протокола ARP. Анализируя структуру запроса, можно увидеть, что компьютер с адресом 192.168.3.2 делает попытку узнать MAC-адрес компьютера с IP-адресом 192.168.3.12. Для этого он посылает широковещательный запрос, содержащий IP-адрес, с MAC-адресом, установленным в 00:00:00:00:00:00. Следует обратить внимание на заголовок MAC header — точнее, на поле Destination, установка которого в FF:FF:FF:FF:FF:FF как раз и указывает на широковещательный характер запроса.

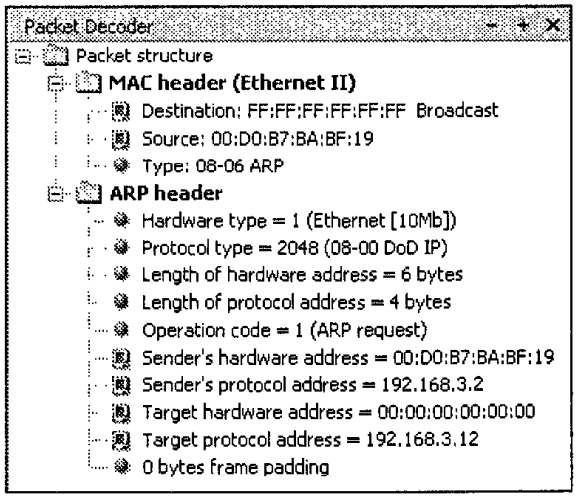


Рис. 2.29. Запрос протокола ARP

Когда компьютер с адресом 192.168.3.12 получает этот широковещательный запрос, он анализирует IP-адрес, для которого выполняется разрешение. Определив, что его адрес совпадает с искомым, он формирует ответ протокола ARP, где указывает свой MAC-адрес (рис. 2.30).

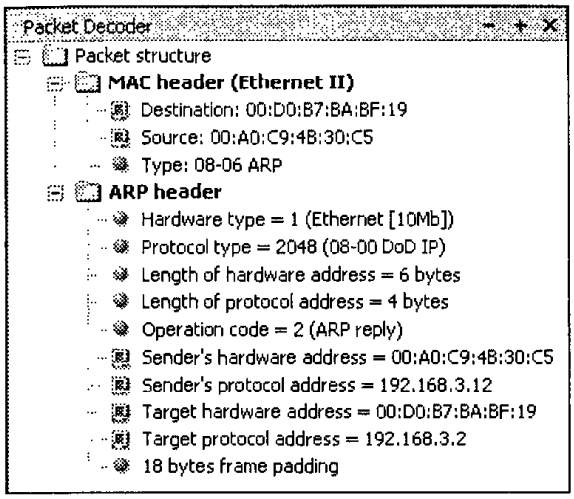


Рис. 2.30. Ответ протокола ARP

Обратите внимание на то, что ответ посылается уже не широковещательно — отправитель знает MAC-адрес инициатора запроса и поэтому передает пакет целенаправленно.

Возможность использования протокола ARP для сетевых атак

Рассмотрев особенности протокола ARP, необходимо остановиться на тех его возможностях, которые он «предоставляет» для проведения компьютерных атак на сеть. Атаки с применением протокола ARP можно отнести к условно активным, так как они требуют выполнения определенных действий в сети (в том время как перехват пакетов с помощью анализатора протоколов можно отнести к пассивным атакам). С точки зрения атакующего, именно активные атаки в состоянии принести ему наибольший успех. Поэтому администратору сети необходимо иметь четкое представление об их механизме. Далее, в главе 5, рассматриваются основы атак типа spoofing (имитация соединений — их можно также отнести к классу атак DoS), которые достаточно популярны и предоставляют злоумышленнику широкие возможности. Но они относительно сложны в реализации, и атакующий, кроме всего прочего, должен надеяться на свою удачу. В противоположность этим атакам, атака на протокол ARP (по аналогии ее можно также назвать *ARP spoofing*) несравненно проще в реализации.

Несмотря на то что сама возможность атаки ARP spoofing подразумевает доступ к сегменту локальной сети, она может быть удачным дополнением при проведении комплексной атаки, когда атакующий уже выявил существование бреши в системе безопасности и получил контроль над одним из компьютеров в сегменте. Собственно, сама вероятность атаки посредством протокола ARP возможна ввиду его простоты и отсутствия элементов защиты. Как нетрудно догадаться, целью атаки является попытка выдать один хост за другой с помощью рассылки ложных сообщений протокола ARP.

Предположим, что атакующий желает выдать свой компьютер сегмента за другой, считая, что последний имеет некоторые привилегии при доступе к серверу. Один из сценариев проведения атаки подразумевает создание одинаковых IP-адресов. Однако такой подход ненадежен, так как большинство современных операционных систем поддерживают механизм обнаружения совпадающих адресов в сети с последующим оповещением пользователей. Такая схема работы необходима для быстрого поиска ошибок в конфигурации IP-адресов.

Однако даже такая ситуация играет на руку атакующему. В Интернете не так сложно найти программу (например, send_arp.c), которая посылает ответы протокола ARP на соответствующие запросы, и в этих ответах возвращается требуемая информация — IP и MAC-адреса. Обратите внимание на то, что при отправке сообщений протокола ARP в сеть другие устройства в сети будут их принимать и учитывать даже в том случае, если не нуждаются в них.

Естественно, из-за устаревания информации в кэше протокола ARP рассылаемые сообщения следует повторять, причем частота повторения зависит от конкретной операционной системы, однако 40 секунд должно хватить для большинства систем. Но может возникнуть проблема с некоторыми системами, которые пытаются обновить записи в кэше с помощью посылки целенаправленных запросов ARP по кэшируемым адресам, для того чтобы убедиться в том, что адреса активны. Это в силах несколько помешать злоумышленнику, так как не исключается изменение записи ARP для компьютера-жертвы, которая была только что подменена атакующим.

ПРИМЕЧАНИЕ

На маршрутизаторах Cisco Systems администратор имеет возможность вручную установить время хранения информации в кэше. Для этого следует воспользоваться командой arp timeout с параметром, указывающим промежуток времени в секундах. Команда выполняется в режиме конфигурирования интерфейсов и соответственно влияет на записи, относящиеся именно к этому интерфейсу. По умолчанию параметр установлен в 14 400 секунд, что соответствует четырем часам. Если в качестве параметра указать значение 0, то маршрутизатор не будет помещать записи в кэш. Для получения информации о текущем значении таймера выполните команду show interface:

```
Router#sh int e0
...
ARP type: ARPA, ARP Timeout 04:00:00
...
```

Рассмотренная атака работает в пределах локальной сети, хотя вообще сфера ее воздействия ограничена только областью распространения пакетов ARP, то есть не очень далеко, так как эти сообщения не маршрутизируются в распределенную сеть. Однако одним из расширений рассматриваемой атаки может быть подмена адреса маршрутизатора. Например, вполне допустима ситуация, когда какой-либо пользователь в локальной сети, подменяя маршрутизатор, выдает фальшивое приглашение Telnet администратору сети с просьбой ввести имя и пароль доступа.

Вообще говоря, работа с протоколом ARP предоставляет довольно широкие возможности атакующему. Имеются шансы блокировать работу практически любого компьютера в локальной сети. Самый простой способ, хотя и не самый

надежный, это анонсирование несуществующего адреса. Его ненадежность в том, что операционная система быстро обнаружит тот факт, что она не получает ответ, и пошлет новый запрос ARP. Оптимальной является подстановка адреса другого работающего компьютера. Последующее поведение жертвы в этом случае зависит от конкретной реализации операционной системы, но чаще всего она будет посылать различные пакеты некорректному получателю, который будет их игнорировать.

Другой способ использования особенностей протокола ARP заключается в указании в сообщении одинакового IP-адреса отправителя и получателя. Некоторые системы используют подобные сообщения в то время, когда эта система объявляет о себе в сети и кэширует информацию. Таким образом, одним общением ARP можно нарушить работу всей сети.

СОВЕТ

Говоря о вероятности атак изнутри, следует также обратить внимание на протокол 802.1 spanning tree, с помощью которого атакующий также способен нарушить работу сети. Но только в том случае, если этот протокол используется в сети.

3 Дальнейшая настройка маршрутизатора

В этой главе продолжается рассмотрение базовой работы с маршрутизаторами Cisco Systems, причем упор делается на работу со службами стека TCP/IP, о которых рассказывалось в предыдущей главе.

Разрешение адресов маршрутизатором

Когда в пользовательском или привилегированном режиме вводятся символьные данные, операционная система подразумевает, что введенные символы соответствуют команде на выполнение. В том случае, если это не команда, то предполагается, что это имя хоста (host name), с которым следует установить сеанс Telnet. На сетевом уровне эталонной модели OSI устройства в сети идентифицируются с помощью адресов, а не имен. Таким образом, маршрутизатору необходимо будет выполнить преобразование имени в адрес. Если говорить о стеке протоколов TCP/IP, то существует два способа разрешения адресов (address resolution). Первый способ использует службу DNS, а второй основан на локальной таблице соответствия имен хостов их адресам (local host table).

Служба системы доменных имен (DNS, Domain Name System) представляет собой приложение, работающее на сервере, которое принимает запросы на преобразование имен и в ответ высылает найденные адреса хостов. Операционная система Cisco IOS способна посылать запрос на сервер имен для установления сеанса Telnet или выполнения других действий (например, на обработку команды ping). Возможность посылки запросов серверу DNS включена по умолчанию, при этом адрес сервера соответствует значению 255.255.255.255, что является широковещательным адресом протокола IP. То есть если оставить данные настройки без изменения, то маршрутизатор будет рассылать широковещательный запрос по всем своим интерфейсам. В последнем случае сервер DNS должен находиться в локальной сети, напрямую подключенной к маршрутизатору.

Если администратор сети не настроил адрес сервера DNS и не хочет, чтобы маршрутизатор посылал широковещательный запрос на разрешение имени, он может воспользоваться командой no ip domain-lookup. Соответственно, при выпол

нении этой команды без префикса no маршрутизатор будет осуществлять поиск, что и определено по умолчанию. Ниже приведен пример, в котором с помощью данной команды маршрутизатору запрещается выполнение разрешения имени.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no ip domain-lookup
Router(config)#exit
Router#ping www.cisco.com
Translating "www.cisco.com"
% Unrecognized host or address, or protocol not running.
Router#
```

Обычно на маршрутизаторе указывается адрес сервера(ов) DNS, поэтому он посылает запросы напрямую по известному адресу. Следующий пример демонстрирует работу команды ping с указанием имени сервера в качестве параметра. Как видно из примера, маршрутизатор выполнил преобразование имени в адрес с помощью известного сервера DNS.

```
Router>ping www.cisco.com
Translating "www.cisco.com"...domain server (193.124.83.69) [OK]
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 198.133.219.25, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 204/206/212 ms
```

Администратор сети может настроить до шести известных ему адресов серверов DNS, выполнив команду ip name-server с адресом (адресами) сервера в качестве параметра, в глобальном режиме конфигурирования.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip name-server ?
   A.B.C.D  Domain server IP address (maximum of 6)
Router(config)#ip name-server 193.124.22.65
```

Напомним, что есть два варианта разрешения имени, один из которых (использование сервера DNS) был рассмотрен выше. Второй вариант основан на установлении соответствия между именами устройств и их адресами при помощи локальной таблицы. Администратор сети может работать с данной таблицей на маршрутизаторе, помещая в нее записи посредством команды ip host, выполняемой в глобальном режиме. Эта команда имеет параметры, первый из которых является именем устройства, а остальные соответствуют его IP-адресам. Причем если администратор сети устанавливает сеанс Telnet с устройством по его имени, то маршрутизатор будет по очереди использовать указанные адреса в процессе подключения. А при выполнении таких команд, как ping или traceroute, маршрутизатор примет во внимание только первый адрес.

```
Router#ping cisco
Translating "cisco"...domain server (193.124.83.69) (193.124.22.65)
% Unrecognized host or address, or protocol not running.
```

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip host cisco 198.133.219.25
Router(config)#exit
Router#ping cisco

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 198.133.219.25, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 204/218/276 ms
```

В варианте локальной таблицы маршрутизатор всегда пытается проверить соответствие имени до того, как отошлет запрос серверу DNS. Администратор сети имеет возможность просмотреть содержимое этой таблицы с помощью команды **show hosts**, выполняемой в пользовательском режиме.

```
Router>show hosts
Default domain is not set
Name/address lookup uses domain service
Name servers are 193.124.83.69, 193.124.22.65

Host                  Flags       Age   Type      Address(es)
www.cisco.com         (temp, OK)  1     IP        198.133.219.25
cisco                 (perm, OK)  0     IP        198.133.219.25
```

Результат выполнения команды содержит список устройств с их адресами, которые известны маршрутизатору. В выходной таблице присутствует столбец Flags, который указывает на тип записи. Это может быть постоянная запись (perm), которая была вручную добавлена в таблицу с помощью команды ip host, или временная запись (temp). Временной запись является в том случае, если она была получена в результате запроса к серверу DNS. Временные записи добавляются в таблицу, так как маршрутизатор кэширует результаты разрешения имен, чтобы не повторять запрос для одного и того же имени устройства. Для очистки таблицы можно воспользоваться командой clear host с указанием имени устройства, запись о котором нужно удалить, или с символом * для удаления всех записей. Для выполнения этой команды следует перейти в привилегированный режим.

Работа с интерфейсами и портами маршрутизатора

Настройка индивидуальных интерфейсов выполняется в режиме конфигурирования интерфейсов или подинтерфейсов. Для перехода в эти режимы следует выполнить команду interface, находясь в глобальном режиме. Упомянутая команда имеет два параметра: тип интерфейса (подинтерфейса) и его номер.

Все интерфейсы, доступные на маршрутизаторе компании Cisco System, делятся на два основных типа: физические и логические. Первый тип характеризуется тем, что он имеет подключение вне маршрутизатора (например, это может быть интерфейс Ethernet в локальную сеть, реализованный с помощью раз-

ема RJ45). Второй тип существует только в памяти маршрутизатора. Логический интерфейс не имеет внешнего физического подключения и присутствует только на уровне настроек маршрутизатора. Так, например, подинтерфейсы представляют собой разновидность логического интерфейса.

Наличие разных типов физических интерфейсов зависит от модели маршрутизатора. Если маршрутизатор имеет физический интерфейс, команды для работы с ним доступны, даже если данный интерфейс не используется. Эти команды также должны вводиться в режиме конфигурирования интерфейсов.

```
Cisco1601#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco1601(config)#interface serial1
Cisco1601(config-if)#ip address 192.168.10.4 255.255.255.0
```

В приведенном выше примере показана настройка адреса сетевого уровня протокола IP на интерфейсе Serial1. Как видно, для перехода в режим настройки интерфейса была выполнена команда interface serial1 в глобальном режиме маршрутизатора, после чего маршрутизатор изменил приглашение на config-if. Из приглашения, выдаваемого маршрутизатором, нельзя узнать, какой интерфейс сейчас находится в стадии настройки.

Логические интерфейсы, помимо всего прочего, отличаются от физических тем, что администратор сети может создавать их по мере необходимости. Примерами логических интерфейсов служат интерфейсы loopback и tunnel. Логический интерфейс loopback создается, если требуется интерфейс, ни к чему не подключенный и отвечающий, например, за управление маршрутизатором с консоли по протоколу SNMP. Такое решение обосновано в случае, если управление маршрутизатором осуществляется через один интерфейс, который периодически переходит в состояние down. При этом система управления способна посчитать, что не работает весь маршрутизатор, а последнее неверно.

Другой логический интерфейс — tunnel — используется для передачи трафика сетевого протокола через распределенную сеть, не поддерживающую работу с соответствующими пакетами. Чаще всего это необходимо для передачи трафика протокола IPX через сеть, понимающую только протокол IP. В такой передаче участвуют два маршрутизатора.

Для создания логического интерфейса также следует воспользоваться командой interface. Вот пример создания интерфейса loopback.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface loopback 0
Router(config-if)#ip address 192.168.11.1 255.255.255.0
```

Логические интерфейсы могут иметь любой номер, отсчитываемый с нуля. В представленном выше примере был создан интерфейс loopback с номером 0. Маршрутизатор немедленно переводит его в состояние up, делая интерфейс доступным для работы. Это можно проверить, выполнив команду show interface loopback0.

```
Router#show interfaces loopback0
Loopback0 is up, line protocol is up
```

```
Hardware is Loopback
   Internet address is 192.168.11.1/24
...
```

При создании распределенной сети с применением таких технологий, ка Frame Relay или X.25, наибольший интерес представляют подинтерфейсы (sub interfaces). Это разновидность логических интерфейсов, привязанных к определен ному физическому. Тип подинтерфейса, в свою очередь, зависит от физическог интерфейса. Несмотря на то что подинтерфейсы наследуют некоторые характе ристики физического интерфейса, к которому они привязаны, они могут иметь свои собственные настройки сетевого уровня, например индивидуальный IP-адрес

ИЗ ОПЫТА

Понимание механизма подинтерфейсов наиболее актуально при подключении маршрутизатора к сети Frame Relay. В такой ситуации на маршрутизатор (точнее, на один физический порт) может замы каться несколько логических (виртуальных) соединений от других маршрутизаторов в сети. Таки соединения называются постоянными виртуальными соединениями (Permanent Virtual Circuits — PVC) и, как правило, они формируются провайдером услуг сети Frame Relay. Выделяют два типа подин терфейсов в сети Frame Relay, зависимые от топологии виртуального соединения: «точка-точка» (двухточечного, point-to-point) и «точка-группа» (многоточечного, multipoint). Первый тип обслужи вает только одно постоянное виртуальное соединение, в то время как второй — несколько таки соединений.

Несмотря на то что подинтерфейсы привязаны к физическому интерфейсу, администратор сети может вручную перевести один из них в нерабочее состоя ние (down). Однако если в нерабочее состояние переходит физический интер фейс, это в полной мере относится и к привязанным к нему подинтерфейсам. В том случае, если подинтерфейс обслуживает одно постоянное виртуальное со единение, которое по каким-либо причинам перестает поддерживаться провай дером, подинтерфейс также переходит в нерабочее состояние.

Схема именования подинтерфейсов включает в себя имя физического ин терфейса с его номером, за которыми следует другой номер, отделенный точкой. Так, например, если интерфейс маршрутизатора Serial0 подключен к сети Frame Relay и нужно настроить подинтерфейс, поддерживающий постоянное виртуаль ное соединение, то его имя будет Serial0.1 (нумерация подинтерфейсов начина ется с 1). В приведенном ниже примере показана настройка подинтерфейса с но мером Serial0.7, который соответствует постоянному виртуальному соединению и имеет тип «точка-точка».

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial0
Router(config-if)#interface serial0.7 point-to-point
Router(config-sub)#description connected to Wharehouse
Router(config-sub)#ip unnumbered ethernet 0
Router(config-sub)#frame-relay interface-dlci 22
Router(config-fr-)#exit
```

Вернемся к вопросу инкапсуляции данных сетевого и высших уровней в еди ницы передачи информации (MTU), использующиеся на канальном уровне

Напомним, что для интерфейсов локальной сети применимы различные виды инкапсуляции, в зависимости от настроенного протокола. Так, для интерфейса Ethernet и протокола IP характерна инкапсуляция ARPA. Для интерфейсов глобальной сети может использоваться только один тип инкапсуляции, который зависит от устройства, подключенного к маршрутизатору, с которым он взаимодействует на канальном уровне. Применительно к технологии Frame Relay на интерфейсе должна быть задана инкапсуляция Frame Relay. Для того чтобы настроить необходимый тип инкапсуляции, следует выполнить команду encapsulation, в режиме конфигурирования интерфейсов с параметром-словом, указывающим на тип инкапсуляции.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial 1
Router(config-if)#encapsulation ?
   atm-dxi      ATM-DXI encapsulation
   frame-relay  Frame Relay networks
   hdlc         Serial HDLC synchronous
   lapb         LAPB (X.25 Level 2)
   ppp          Point-to-Point protocol
   smds         Switched Megabit Data Service (SMDS)
   x25          X.25
```

Хорошей практикой является настройка описания (description) на каждом интерфейсе, которое помогает понять, за что отвечает тот или иной интерфейс. Описание представляет собой текстовую информацию, записываемую в конфигурационный файл. Для создания описания предусмотрена команда description, выполняемая в режиме конфигурирования интерфейсов.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial1
Router(config-if)#description This interface is shutdown
```

Администратор сети может ввести любую информацию, но, на взгляд автора, наиболее полезным будет указание места подключения данного интерфейса. При этом администратор ограничен в наборе текста 240 символами. Созданное описание отображается при выполнении команды show interfaces. Ниже приведен пример выполнения этой команды для подинтерфейса Serial0.7, представляющего собой постоянное виртуальное соединение сети Frame Relay с удаленным объектом. Как видно, описание помогает легко определить, какой именно объект находится на другом конце данного виртуального соединения.

```
Router#show interfaces serial0.7
Serial0.7 is up, line protocol is up
   Hardware is QUICC Serial
   Description: connected to Wharehouse
   Interface is unnumbered. Using address of Ethernet0 (192.168.3.254)
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation FRAME-RELAY
```

Выполняя команду show interfaces, администратор сети «снимает показания» различных счетчиков (counters), показывающих объемы передаваемой через интерфейс информации. В его распоряжении есть команда, которая позволяет обнулить эти счетчики. Выполнение в привилегированном режиме команды clear counters без параметров приведет к обнулению счетчиков для всех интерфейсов, но только после подтверждения этого действия. Можно обнулить счетчики для определенного интерфейса, указав его имя и номер в качестве параметра команды.

Ранее мы уже рассматривали терминальные порты, а сейчас остановимся на их настройке. Для того чтобы начать настраивать терминальный порт, следует перейти в соответствующий режим (line configuration mode). Для этого предназначена команда line, выполняемая в глобальном режиме. При указании номера порта применяется либо абсолютная, либо относительная нумерация. В случае последней обязательно следует указать тип порта, например CON или VTY. Ниже приведен пример перехода в режим настройки консольного порта.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#line console 0
Router(config-lin)#
```

На маршрутизаторе существует только один консольный порт, следовательно, его относительный (и абсолютный тоже) номер всегда равен нулю. После выполнения команды line изменяется приглашение маршрутизатора. Оно указывает на то, что теперь работа ведется в другом режиме настройки. Настройка порта не ограничена одним вариантом, например виртуальные порты VTY администратор вправе настраивать как индивидуальный порт или как группу. Как уже отмечалось, обычно все порты VTY на маршрутизаторе имеют одинаковые настройки, следовательно, можно указывать группу номеров, разделяемых пробелом.

```
Router(config)#line vty 0 4
Router(config-lin)#
```

После 10 минут неактивности маршрутизатор автоматически отключит сеанс, установленный на терминальном порте. Администратор сети имеет возможность изменить этот интервал с помощью команды exec-timeout, требующей указания двух параметров. Первый параметр определяет минуты ожидания, а второй — секунды. Если задать нулевые параметры, это приведет к тому, что маршрутизатор не станет отключать сеанс автоматически, а будет ожидать ручного отключения.

```
Router(config)#line console 0
Router(config-lin)#exec-timeout 0 0
```

В распоряжении администратора есть возможность принудительно завершить тот или иной сеанс работы с маршрутизатором. Это отключение осуществляется командой clear line. Если администратор обнаружил, что кто-то, не имеющий на то особых оснований, выполнил подключение к маршрутизатору, например через сеанс Telnet, то в его власти выполнить указанную команду с параметром, соответствующим номеру виртуального порта. Покажем это на примере. Предположим, что было осуществлено подключение с удаленного маршрутизатора Cisco 805 к центральному Cisco 1601 с адреса 192.168.3.254.

```
Cisco805#telnet 192.168.3.254
Trying 192.168.3.254 ... Open
```

Администратор сети, работая с центральным маршрутизатором, может просмотреть текущие подключения к нему с помощью команды **show users**, которая покажет, с каких адресов выполнялось подключение. Символом * отмечено подключение самого администратора. Обнаружив факт подключения с адреса 192.168.7.254, администратор вправе принудительно закрыть сеанс посредством команды **clear line 4**.

```
Cisco1601#show users
   Line          User     Host(s)        Idle Location
*  3 vty 0       idle                00:00:00 192.168.3.20
   4 vty 1       idle                00:00:11 192.168.7.254
Cisco1601#clear line 4
[confirm]y [OK]
```

После этого на удаленном маршрутизаторе появится сообщение, говорящее о том, что соединение было закрыто: **Connection to 192.168.3.254 closed by foreign host.**

Работа с маршрутизатором через сеанс Telnet

Очень удобным средством управления маршрутизатором является возможность доступа к нему через распределенную сеть путем установления сеанса Telnet на одном из виртуальных портов маршрутизатора. При создании сеанса требуется указать либо адрес, либо имя удаленного маршрутизатора. Причем подключение можно делать как с компьютера, с которого осуществляется управление (из одноименной программы Telnet, например в операционных системах Microsoft Windows NT/2000), так и прямо с маршрутизатора. В последнем случае команда **telnet** вводится либо в пользовательском, либо в привилегированном режиме.

```
Cisco1601#telnet 192.168.7.254
Trying 192.168.7.254 ... Open

User Access Verification

Password:******
Cisco805>
```

Если вместо адреса удаленного маршрутизатора было указано его имя, то маршрутизатор, с которого инициируется сеанс Telnet, будет пытаться выполнить процедуру разрешения имени. Хотя в примере вместо пароля указаны символы *, на самом деле при вводе пароля на экране ничего не отображается. Напомним, что если бы на маршрутизаторе Cisco 805 не был настроен пароль для терминальных портов VTY, то установить сеанс Telnet не удалось бы вообще. Для того чтобы после установления сеанса Telnet с удаленным маршрутизатором

временно отложить его, необходимо «подавить» (suspend) этот сеанс, который остается активным и к нему можно вернуться позже без прохождения повторной авторизации. Для подавления сеанса нужно нажать следующую комбинацию клавиш: Ctrl+Shift+6 и x (сначала одновременно нажимаются клавиши Ctrl+Shift+6, а затем клавиша x).

```
...
Cisco805><Ctrl-Shift-6><x>
Cisco1601#
```

Хотя после нажатия такой комбинации мы вернулись в приглашение маршрутизатора Cisco 1601, сеанс Telnet с удаленным маршрутизатором по-прежнему активен, в чем можно убедиться с помощью команды show sessions, которая показывает подавленные сеансы Telnet.

После подавления одного сеанса Telnet можно установить еще один, а затем подавить его и т. д. В этом случае после выполнения команды show sessions будет показан список всех удаленных устройств, с которыми установлены сеансы Telnet. Для примера установим с маршрутизатора Cisco 1601 еще один сеанс Telnet, а затем переведем его в подавленное состояние и выполним команду show sessions.

```
Cisco1601#show sessions
Conn Host            Address          Byte  Idle  Conn Name
   1 192.168.7.254   192.168.7.254    0     0     192.168.7.254
*  2 192.168.4.254   192.168.4.254    0     0     192.168.4.254
```

Следует обратить внимание на символ *. Он указывает на сеанс, который был подавлен последним. Если теперь просто нажать клавишу Enter в пустой командной строке (не вводя никакой команды), маршрутизатор, с которого устанавливались сеансы, восстановит тот из них, который отмечен символом *.

```
Cisco1601#<Enter>
[Resuming connection 2 to 192.168.4.254 ... ]

Cisco805>
```

Если требуется восстановить сеанс Telnet, не отмеченный символом *, можно воспользоваться командой resume. В качестве параметра команды нужно указать номер сеанса, полученный из результата выполнения команды show sessions.

```
Cisco1601#show sessions
Conn Host            Address          Byte  Idle  Conn Name
   1 192.168.7.254   192.168.7.254    0     6     192.168.7.254
*  2 192.168.4.254   192.168.4.254    0     0     192.168.4.254
Cisco1601#resume 1
[Resuming connection 1 to 192.168.7.254 ... ]

Cisco805>
```

Для того чтобы закрыть сеанс Telnet с удаленным маршрутизатором, следует выполнить команду disconnect, с параметром, соответствующим номеру сеанса. Если выполнить команду без параметра, то она завершит сеанс, отмеченный символом *.

```
Cisco1601#show sessions
Conn Host              Address            Byte  Idle  Conn Name
*  1 192.168.7.254     192.168.7.254      0     0     192.168.7.254
   2 192.168.4.254     192.168.4.254      0     3     192.168.4.254

Cisco1601#disconnect
Closing connection to 192.168.7.254 [confirm]y
Cisco1601#show sessions
Conn Host              Address            Byte  Idle  Conn Name
*  2 192.168.4.254     192.168.4.254      0     3     192.168.4.254
```

Протокол CDP

Протокол CDP (Cisco Discovery Protocol) разработан компанией Cisco Systems и является протоколом канального уровня. По умолчанию он работает на большинстве произведенного компанией оборудования и используется для оповещения соседних устройств о присутствии базовой информации о конфигурации. Так как протокол работает на канальном уровне эталонной модели OSI, он не является маршрутизируемым и работает в любых сетях, поддерживающих тип инкапсуляции SNAP, например Ethernet, FDDI, HDLC, PPP и Frame Relay. Устройства, поддерживающие протокол CDP, периодически рассылают сообщения через интерфейсы, на которых данный протокол настроен. При этом промежуток между посылками сообщений составляет 60 секунд. После получения сообщений в памяти устройства формируется таблица, которая содержит информацию о соседях. Кроме того, запускается таймер, который начиная со 180 секунд ведет обратный отсчет. При достижении таймером нулевого значения запись об устройстве удаляется из таблицы. На маршрутизаторах компании Cisco Systems администратор имеет возможность просмотреть таблицу протокола CDP с помощью команды show cdp neighbors.

```
Cisco1601#show cdp neighbors
Capability Codes:  R - Router, T - Trans Bridge, B - Source Route Bridge
                   S - Switch, H - Host, I - IGMP, r - Repeater

Device ID    Local Intrfce    Holdtme  Capability  Platform    Port ID
Cisco1005_1     Ser 0.6         122        R        1000        Ser 0.1
Cisco805_2      Ser 0.5         124        R        Cisco C805  Ser 0.1
Cisco805_3      Ser 0.3         152        R        Cisco C805  Ser 0.1
Cisco1005_4     Ser 0.1         123        R        1000        Ser 0.1
Cisco1005_5     Ser 0.2         125        R        1000        Ser 0.1
```

По результатам выполнения этой команды на маршрутизаторе Cisco 1601 видно, что он соединен с пятью соседними маршрутизаторами, от которых были получены сообщения протокола CDP. Также выдается информация о таймере и о платформе соседнего маршрутизатора (1000, Cisco C805). Кроме того, можно узнать, через какой интерфейс были получены сообщения этого протокола. Для того чтобы увидеть полное содержимое таблицы, выполним команду **show cdp neighbors detail**.

```
Cisco1601 #show cdp neighbors detail
...
-------------------------
Device ID: Cisco805_2
Entry address(es):
IP address: 192.168.4.254
Platform: Cisco C805, Capabilities: Router
Interface: Serial0.5, Port ID (outgoing port): Serial0.1
Holdtime : 143 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) C805 Software (C805-Y6-MW), Version 12.0(4)XM, EARLY DEPLOYMENT
   RELEASE
 SOFTWARE (fc1)
TAC:Home:SW:IOS:Specials for info
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 17-Jun-99 16:51 by linda
-------------------------
...
```

Можно получить и более подробную информацию о соседних маршрутизаторах, такую, например, как адрес сетевого уровня, сведения об используемой версии операционной системы Cisco IOS. Информация об адресе сетевого уровня бывает полезной в случае, когда требуется установить связь с маршрутизатором, адрес которого неизвестен. В дополнение к полному перечню информации о соседних маршрутизаторах, выполнив команду **show cdp entry** с параметром, обозначающим имя устройства из таблицы, можно просмотреть индивидуальную запись в таблице CDP.

Администратор сети в состоянии управлять работой этого протокола на маршрутизаторе в целом или только на нескольких его интерфейсах. Если необходимо отключить поддержку протокола CDP на определенном интерфейсе, нужно выполнить команду **no cdp enable** в режиме конфигурирования интерфейсов. Для отключения протокола на всех интерфейсах одновременно следует, находясь в глобальном режиме, ввести команду **no cdp run**. Кроме того, может потребоваться изменить значение интервала отправки сообщений CDP (update interval), которое будет воздействовать на все интерфейсы сразу. Это делается с помощью команды **cdp timer**, где параметром является количество секунд между обновлениями. Для настройки таймера, работающего с таблицей соседей CDP, служит команда **cdp holdtime** с аналогичным параметром. После выполнения всех настро

ек протокола можно просмотреть его статус с помощью команд **show cdp** и **show cdp interface**.

```
Cisco1601#show cdp
Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds

Cisco1601#show cdp interface serial0.1
Serial0.1 is up. line protocol is up
   Encapsulation FRAME-RELAY
   Sending CDP packets every 60 seconds
   Holdtime is 180 seconds
```

Первая команда показывает значения таймера и интервала обновления, а вторая выводит информацию по протоколу CDP для конкретного интерфейса (или всех интерфейсов) маршрутизатора.

СОВЕТ

Следует отметить, что на маршрутизаторах, располагающихся на границе частной сети и Интернета, поддержку протокола CDP обычно отключают, чтобы снизить риск получения злоумышленником важной информации о внутренней структуре сети.

Работа со встроенным отладчиком

Стоит отметить еще одну полезную возможность операционной системы Cisco IOS, а именно получение отладочной информации в целях предсказания реакции операционной системы на определенные действия. Обычно это бывает необходимо для решения возникающих проблем или просто ради эксперимента. Для этой цели предназначена команда **debug**, позволяющая отслеживать различные типы активности, соответствующие происходящим внутри маршрутизатора процессам.

Чтобы узнать возможные варианты использования этой команды и получить справку, следует ввести после ее имени знак вопроса. По умолчанию вывод отладочной информации включен для консольного порта маршрутизатора. К команде **debug** следует относиться осторожно, так как ее выполнение связано с большой загрузкой маршрутизатора и значительными объемами отображаемой на экране информации. Ниже приведен пример отладочной информации маршрутизатора для протокола TCP (команда **terminal monitor** позволяет просматривать отладочную информацию при работе в сеансе Telnet).

```
Router#terminal monitor
Router#debug ip tcp packet vty 0
TCP Packet debugging is on for line number 3
Router#
3w3d: tcp3: O ESTAB 192.168.3.20:4175 192.168.3.254:23 seq 2023153705
        DATA 64 ACK 3169145 PSH  WIN 2949
```

```
3w3d: tcp3: I ESTAB 192.168.3.20:4175 192.168.3.254:23 seq 3169145
        ACK 2023153769  WIN 7802
3w3d: tcp3: O ESTAB 192.168.3.20:4175 192.168.3.254:23 seq 2023153769
        DATA 216 ACK 3169145 PSH  WIN 2949
3w3d: tcp3: I ESTAB 192.168.3.20:4175 192.168.3.254:23 seq 3169145
        ACK 2023153985  WIN 7586
3w3d: tcp3: O ESTAB 192.168.3.20:4175 192.168.3.254:23 seq 2023153985
        DATA 217 ACK 3169145 PSH  WIN 2949
...
```

Если необходимости в отладке нет, этот вывод можно отключить, выполнив команду с префиксом no. Когда для одного маршрутизатора одновременно включено несколько отладочных терминалов, выключить их все сразу позволяет команда no debug all.

4 Протокол ICMP

Введение

Протокол управляющих сообщений сети Интернет (ICMP, Internet Control Message Protocol) был задуман и реализован как средство для уведомления о возникающих ошибках в Сети и для формирования ответов на простые запросы, в расчете на ненавязчивое потребление сетевых ресурсов и внесение минимальных возмущений в картину общего трафика. Протокол описан в документе RFC 972, а его последующие обновления в документах RFC 896 (Source Quench — подавление источника), RFC 950 (Address Mask Extensions — расширения адресной маски), RFC 1191 (Path MTU Discovery — метод обнаружения MTU в пути до получателя) и RFC 1256 (Router Discovery — обнаружение маршрутизатора).

Без знания теоретических основ трудно осмыслить трафик, порождаемый протоколом ICMP, однако стоит обратить внимание, что в сети протокол будет работать и без понимания администратором принципов его функционирования, то есть его можно воспринимать как данность. Настройки этот протокол практически не требует. По сравнению с другими протоколами ICMP используется реже и полосу пропускания при нормальной работе практически не занимает.

Наибольший интерес представляет нетрадиционное, не предусмотренное разработчиками протокола его использование. И если администратор сети имеет четкое представление, чем оно может обернуться, то ему по силам продумать стратегию защиты и реализовать саму защиту сети от такого применения. По крайней мере, знание особенностей нетрадиционного использования протокола ICMP позволит понять, какие именно цели преследует атакующий. Кстати, для сбора информации при сканировании из сети Интернет злоумышленники часто прибегают именно к этому протоколу.

Со временем возможности протокола ICMP позволили атакующим взять его на вооружение качестве инструмента противоправных действий. По этой причине важно понимать отличие в работе протокола в легальных условиях от случая проведения злонамеренных (malicious) операций. В процессе рассмотрения материала будут освещены некоторые аспекты работы протокола, начиная, естественно, с базовых сведений о нем. Затем пойдет речь о способах, какими ICMP осуществляет поиск работающих устройств (хостов) в сети. А дальше читатель найдет информацию о том, какой трафик этого протокола можно встретить в локальной сети при нормальных условиях и при его противоправном исполь-

зовании. Опираясь на эту информацию, можно будет попробовать поискать некорректные (не с точки зрения протокола, а с позиций права) сообщения в своей сети. В заключение будут даны практические рекомендации о способах защиты сети от входящих из сети Интернет сообщений протокола ICMP с помощью списков управления доступом, настроенных на граничном устройстве.

Сделаю небольшое отступление и напомню, что широко распространенный протокол TCP (главу 5) примечателен тем, что выполняет массу действий для гарантированной надежной передачи данных. В противовес ему, протокол UDP, хотя и не гарантирует таковую, тем не менее требует наличия на стороне сервера порта, с которым он будет взаимодействовать. Простой запрос, например, запрос по определению активности хоста в сети (известен как ping), не нуждается в портах для взаимодействия и в повышенной надежности передачи. Кроме того, представьте ситуацию, когда маршрутизатор в сети обнаружил ошибку и ему необходимо известить отправителя о данном факте, то как лучше это сделать? Протокол TCP — довольно мощный механизм, способный обрабатывать самые разнообразные сбои в пересылке данных. Например, если сервер получил большой объем информации от клиента, он может установить размер своего окна приема равным нулю, указывая на то, что отказывается принимать данные до тех пор, пока содержимое буфера не будет им обработано. В противоположность протоколу TCP протоколы UDP и IP не настолько развиты для того, чтобы обеспечивать подобный уровень контроля за ошибками. По этой причине им в помощь и создан протокол ICMP, формирующий извещения о возникновении проблемных ситуаций и пересылающий их между хостами или между маршрутизатором и хостом.

Следует думать, что читатель имеет представление о модели OSI и стеке протоколов TCP/IP, но не грех напомнить кое-какие базовые понятия. Пользователи, как правило, работают на прикладном уровне, представляющем приложения, например Telnet. Далее (учитывая разницу в теории и практической реализации) следует транспортный уровень, где такие протоколы, как TCP и UDP, обеспечивают взаимодействие между сетевыми устройствами. Ниже располагается сетевой уровень, отвечающий за доставку дейтаграмм от отправителя к получателю. Именно на этом уровне протокол ICMP нашел себе место — там же, где и протокол IP. Несмотря на то, что сообщения протокола ICMP инкапсулируются в дейтаграммы IP, примыкая к заголовкам последнего, принято считать, что уровни этих двух протоколов одинаковы. Общая структура инкапсуляции сообщений протокола ICMP показана на рис. 4.1.

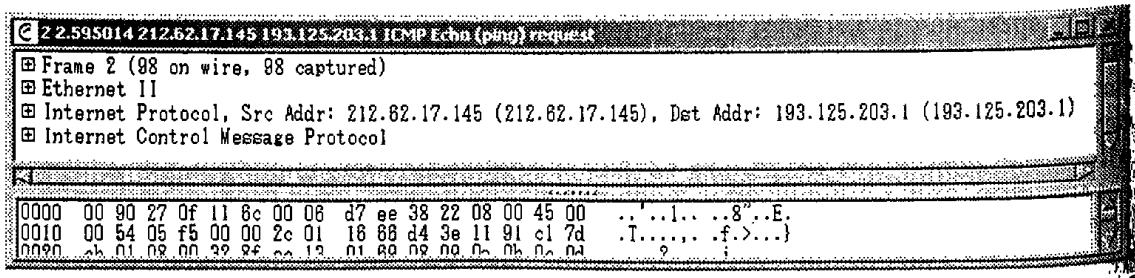


Рис. 4.1. Общая схема инкапсуляции сообщений протокола ICMP

Как видно из рисунка, в данном случае базовым является кадр сети Ethernet. Этот кадр обозначен как Ethernet II, в его полях содержатся MAC-адреса получателя и отправителя. Кроме того, предоставляется информация о типе протокола. В кадр Ethernet «завернута» дейтаграмма протокола IP, которая, в свою очередь, как матрешка, инкапсулировала некое сообщение протокола ICMP.

Перехват информации в сетях

Можно с уверенностью заявить, что многие специалисты сталкивались с ситуацией, когда после изучения какого-либо вопроса путем штудирования специализированной литературы чисто теоретические аспекты становятся предельно ясны, но эта ясность куда-то исчезает, как только начинаешь реализовывать полученные знания на практике. Возникают проблемы, которые раньше и не приходили в голову. Подобная аналогия применима и к сети — по умолчанию она работает и практически не требует к себе особого внимания. На сетевом оборудовании мигают лампочки, битики-байтики снуют туда сюда и, вроде бы, пользователи не имеют повода для недовольства. Но стоит только установить анализатор пакетов, как появляется множество вопросов — а почему от этого хоста в сеть был отправлен такой пакет? Что вызвало его отправку и чем это закончится? Отсюда альтернатива: либо оставить все как есть и не забивать себе голову тем, как это все там происходит и почему, либо разбираться до самого конца. Второй путь тернист и предназначен для пытливых умов. Наградой за этот большой труд станет полное подчинение всех сетевых процессов вашим желаниям, способность реализовать на практике любые поставленные перед вами задачи.

Данная глава (как и глава 10 о виртуальных частных сетях) содержит большое количество примеров с перехваченными сетевыми пакетами. По этой причине имеет смысл рассмотреть инструменты, с помощью которых эти перехваты были выполнены. В процессе подготовки материала для главы автору пришлось опробовать несколько программных продуктов, которые предназначены для решения одной задачи — для перехвата и анализа трафика в сети, но при этом имеют множество отличий как в удобстве пользования, так и в представлении информации. Если администратор сети работает с платформой Microsoft Windows, то он может задействовать входящий в установочный комплект Windows NT/2000 Server программный продукт Microsoft Network Monitor. Но, к сожалению, версия эта слегка «обрезана», и для более детального ознакомления с трафиком предлагается продукт, включенный в состав комплекта Microsoft SMS Server.

Далее, можно приобрести очень мощный программный комплекс Observer компании Network Instrumens или, например, eEye Iris Network Traffic Analizer. Вначале автор использовал последний продукт, пятнадцатидневная пробная версия которого доступна всем желающим после регистрации с сервера www.eeye.com. Однако при всем удобстве работы с этой программой сохранение перехваченного трафика в читаемом виде было делом не самым легким — приходилось копировать целиком внешний вид окна программы в буфер обмена, а затем с помощью графического редактора «вырезать» нужные куски (именно таким образом были получены фрагменты сегментов TCP, приведенные в следующей главе данной книги). Кроме того, вопрос правомерности подобных действий был до конца не

ясен. В конечном итоге была выбрана программа WinDump, которая свободно распространяется в сети Интернет. Следует отметить, что сама программа WinDump является версией для платформы Windows популярного инструмента TCPdump, ориентированного на системы UNIX. Возможности обеих программ, за малым исключением, одинаковы и практически все сказанное далее про WinDump будет верно и для TCPdump. Сам продукт и дополнительную информацию о способах его использования можно найти по адресу http://WinDump.polito.it.

Программа WinDump представляет собой простой (но не с точки зрения функциональности) и компактный перехватчик сетевого трафика. Кстати, именно компактность в большинстве случаев помогает принять решение о том, какой именно инструмент подходит больше прочих. Автору часто приходилось устанавливать программу на разные компьютеры в сети и, как оказалось, вся она целиком помещается на обычный 3,5-дюймовый гибкий диск, оставляя место под документацию. Важно отметить, что перед тем как запустить программу WinDump, необходимо установить специальный драйвер WinPcap, который также доступен по адресу http://winpcap.polito.it.

При этом был приобретен опыт установки этого драйвера на критичные серверы Windows 2000, который показал отсутствие каких-либо проблем. Справедливости ради следует отметить, что один раз, при сборе программой WinDump трафика на защитном экране с двумя сетевыми адаптерами, внезапно появился «синий экран смерти», что было не очень приятно. И вдвойне было досадно то, что продолжал действовать параметр разрешения записи дампа (dump) памяти при сбое системы, не отключенный заблаговременно. В результате, в то время, пока пользователи сети звонили и спрашивали, что происходит с Интернетом, автор сидел и смотрел на цифры и символы сгенерированного дампа памяти. Важно отметить, что хотя процедура установки драйвера не требует перезагрузки системы, ее рекомендуется выполнить. Вообще говоря, если после прочтения этого материала появится желание поэкспериментировать, то смело ставьте драйвер WinPcap на каждый настраиваемый сервер сети — опыт показал, что никаких побочных эффектов, как то снижение производительности или периодические сбои, он не привносит.

Сама программа WinDump не требует процедуры установки. Достаточно просто запустить одноименный файл WinDump.exe, как программа сразу начинает выводить на экран проходящий через выбранный интерфейс трафик, естественно, при наличии установленного драйвера WinPcap. Собственно, за выбор нужного интерфейса, на котором следует производить перехват, выбор типа трафика для перехвата и т. п. отвечают параметры командной строки. Если в системе присутствует несколько сетевых адаптеров, то при запуске WinDump можно указать тот трафик, который вас интересует. Для этого предназначен параметр –i, за которым указывается имя или номер адаптера. Имя, к сожалению, может быть не очень удобным для восприятия, номер в этом смысле выигрывает. А чтобы его узнать, достаточно набрать WinDump –D, в ответ на экран будут выведены имена и номера доступных для работы адаптеров.

Обычно весь трафик перехватывать не требуется, да и для анализа это сложно — попробуйте запустить программу на загруженном хосте — экран (а именно на экран по умолчанию выводится информация) моментально заполнится и все это будет малоинформативно. Поэтому следует воспользоваться фильтрами,

Фильтр — это выражение, которое указывает программе, какой тип трафика перехватывать. Выражения формулируются на простом языке, фактически наборе ключевых слов. Например, для сбора только трафика TCP нужно выполнить WinDump.exe tcp. Сейчас не будем останавливаться на разъяснении всех ключевых слов, тем более, что они все описаны в документации. Единственное, что хочется отметить, это параметр –F с указанием имени файла, содержащего настроенный фильтр. Его использование удобно, когда записывать фильтрующее выражение в виде параметра командной строки не хочется.

Как уже было отмечено, по умолчанию программа выводит перехваченный трафик на экран в текстовом виде, как показано ниже.

```
C:\Program Files>WinDump
WinDump: listening on\Device\Packet_{D63D72F6-785A-451E-A7C7-2C6137CE9634}
13:55:13.091782 beetle.alternativa.spb.ru > ALL-SYSTEMS.MCAST.NET: ICMP: router
advertisement lifetime 30:00 1: {beetle.alternativa.spb.ru 0} [ttl 1]
13:55:13.292800 i-mk.alternativa.spb.ru.3398 > bobcat.alternativa.spb.ru.53: 84
+ PTR? 1.0.0.224.in-addr.arpa. (40)
13:55:14.219554 a-sbd.alternativa.spb.ru.138 > 192.168.3.255.138:
>>> NBT UDP PACKET(138) Res=0x110E ID=0x8036 IP=192 (0xc0).168 (0xa8).3 (0x3).15
6 (0x9c) Port=138 (0x8a) Length=187 (0xbb) Res2=0x0
SourceName=A-SBD        NameType=0x20 (Server)
DestName=TO             NameType=0x00 (Workstation)
```

Однако для проведения последующего анализа чаще всего программа будет настроена на запись перехваченных данных в двоичном виде в указанный файл (так называемый «сырой» вывод — raw output). Если в текстовом формате представляется не вся возможная информация о пакете, то в двоичном виде пакет записывается целиком, включая поле данных.

Для того чтобы начать собирать трафик в двоичном виде, можно выполнить программу с ключом –w <имя файла>. Имя указывается для того файла, в который будут записываться данные. Для того чтобы прочитать сохраненные двоичные данные, добавьте параметр –r <имя файла>, после этого данные из файла будут представляться в удобном для восприятия виде. Можно и сразу перенаправлять поток перехваченного трафика именно в текстовый файл, в той форме, как его выводит программа WinDump. Однако, несмотря на то, что информация будет полной, она не станет более удобочитаемой. Например, при рассмотрении принципов работы протокола PPTP (Point-to-Point Tunneling Protocol — протокол сквозного туннелирования; глава 10) очень интересно показывать результаты инкапсуляции дейтаграмм, а также результаты работы алгоритма шифрования именно на примере поля данных.

В результате поисков в Интернете был найден очень удобный анализатор трафика под названием Ethereal. Его уникальность в том, что при своей работе он опирается на предустановленный драйвер WinPcap и умеет показывать в удобном виде двоичный файл, записанный программой WinDump. Практически все приведенные примеры перехвата в книге выполнялись с помощью программы WinDump с параметром –w, а затем представлялись в виде дерева вложений. Саму программу и информацию по ее применению можно найти по адресу http://www.ethereal.com. Для отображения перехваченных пакетов использовалась версия Ethereal 0.9.8.

Перед тем как приводить примеры перехватов, остановимся еще на одн параметре, который позволяет указать объем перехватываемых данных. По ум чанию WinDump не пытается анализировать всю сохраняемую дейтаграмму для снижения потребления ресурсов. Однако, зачастую для анализа требуе не только заголовок, но и данные, следующие за ним. Особенно это полезно, гда сами данные, а не заголовок представляют интерес для анализа (наприм пакет PPTP). Для этого вводится понятие размера перехватываемых данн (snapshot length). Если программе напрямую не указывается данный парамет то она выбирает только первые 68 байт.

Рассмотрим обычный кадр сети Ethernet. Первые его 14 байт занимает заг ловок кадра, за ними следуют 20 байт заголовка дейтаграммы IP, далее, буде считать, что 20 байт заголовка пакета TCP и собственно данные. То есть п умолчанию из 68 байт, сохраняемых программой WinDump, после вычета все заголовков остается порядка 14 байт полезной нагрузки. Для того чтобы из менить эту настройку, при запуске программы нужно задать параметр **-s** с значением желаемого количества байтов для перехвата. Если требуется записы вать содержимое всего кадра Ethernet, то это значение следует сделать равны 1514 (максимальный размер кадра Ethernet). Тогда за вычетом 14 байт заголов ка кадра, оставшиеся 1500 байт данных и будут представлять собой MTU дл сети Ethernet. Обобщая сказанное, отмечу, что для перехвата трафика с после дующим анализом с помощью программы Ethereal предпочтительно набрать сле дующую команду: **WinDump -s 1514 -w <имя файла>**.

Несмотря на то, что чаще всего в книге будут приводиться так называемы «экранные снимки» (screenshots) перехватов пакетов, важно понимать форма текстовых данных, представляемых WinDump. Это становится вдвойне важно если в добавление к простому «ручному» анализу читатель захочет установит в своей сети систему обнаружения атак Snort, которая существует как для пла формы UNIX, так и для платформы Microsoft Windows. Snort Intrusion Detectio System (IDS) — это бесплатная (freeware) система обнаружения атак, разраб танная Мартином Рошем, основателем и генеральным директором компани Sourcefire, первая IDS с открытыми исходными кодами, которая ныне становит ся все более и более популярной. Существует великолепный сайт **www.snort.or** содержащий массу подробной информации об установке, настройке, расширени системы Snort и об использовании готовых сигнатур (signatures). Можно ска зать, что сигнатуры представляют собой описания наиболее популярных ата количество которых постоянно умножается. В базе данных сигнатур Snort в найдете все самые последние поступления. Сама программа Snort невелика п размеру. Например, по сравнению с коммерческими продуктами ISS RealSecure

Важность понимания текстового формата вывода информации программы WinDump обусловлена тем, что информация об обнаруженных атаках может за писываться в текстовый файл вместе с заголовками перехваченных пакетов. Од нако далее я буду приводить всю перехваченную информацию в виде фрагмен тов иерархической структуры, так как такое представление гораздо нагляднее чем последовательность текстовых строк. Кроме того, весомыми достоинства представления данных в перехваченных пакетах анализатором Ethereal являет ся его «интуитивная понятность» и большое количество поддерживаемых пр

токолов. Практически все приведенные перехваты были получены следующим образом — на выбранном сервере запускалась программа WinDump.exe с параметрами, указывающими на то, что нужно записывать «сырые» данные в указанный файл. Затем этот файл переносился на компьютер автора, где он анализировался с помощью Ethereal. К слову, размеры файлов зависят от интенсивности проходящего трафика и настроенных фильтров перехвата, но обычно файлы умещались на один гибкий диск. В окне анализатора вид каждого перехваченного пакета зависит от конкретных протоколов, участвовавших в передаче информации. Например, если был осуществлен перехват пакета протокола HTTP, то общая картина может напоминать следующую (рис. 4.2).

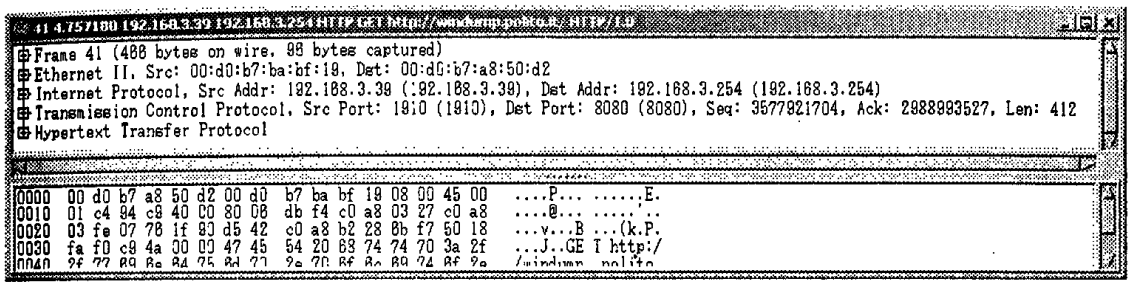


Рис. 4.2. Пример перехвата пакета протокола HTTP

В верхней части окна приводится сводная информация о вложенных протоколах разных уровней после декодирования анализатором. В нижней части та же информация представлена в шестнадцатеричном виде и в кодах ASCII. Перед каждой строкой в верхней половине окна присутствует символ «+», который означает, что соответствующую запись можно отобразить в более детальном виде — это очень полезно, особенно если нужно посмотреть содержимое определенного поля протокола — например, поле контрольной суммы дейтаграммы IP, которая по умолчанию не отображается. Вообще говоря, анализатор отображает только те поля, которые чаще всего используются при анализе. Первая строка содержит общую информацию о перехваченном пакете (рис. 4.3).

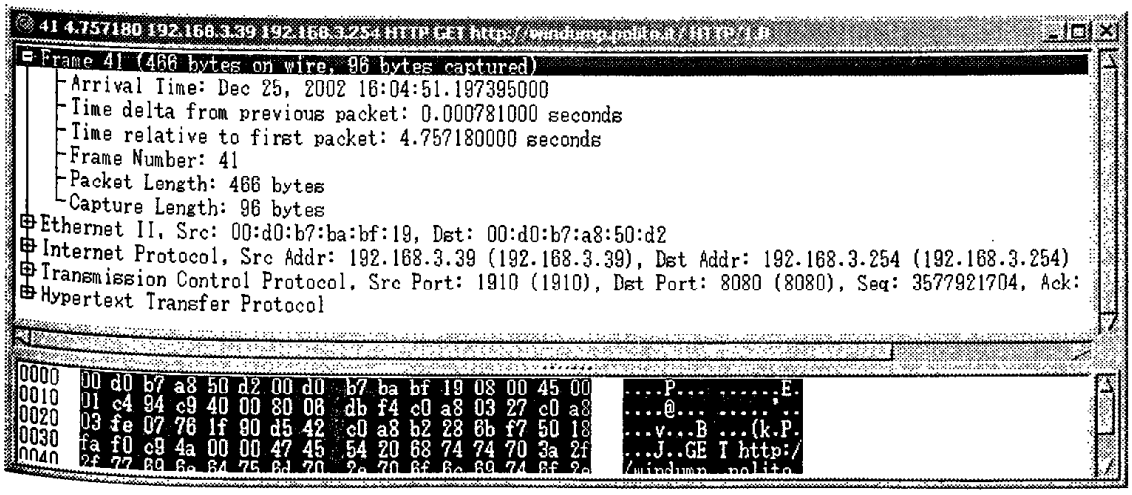


Рис. 4.3. Анализ общей информации о перехваченном пакете

Если развернуть эту строку, то мы увидим информацию о самом кадре: врем его перехвата программой WinDump, порядковый номер, временные отклоне ния от других пакетов и самое интересное — его длину (466 байт). Это обща длина пакета, включая все вложенные данные. Поскольку пакет был перехваче в сети Ethernet, то далее приведена информация, относящаяся к кадру Etherne (рис. 4.4). Все перехваченные пакеты, представленные в данной главе, буду иметь такой заголовок.

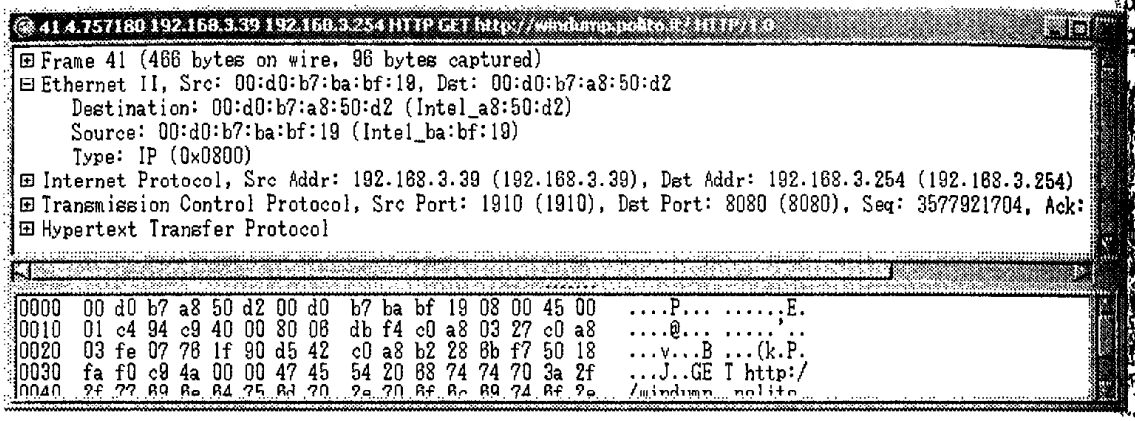


Рис. 4.4. Информация о кадре Ethernet

На этом уровне приводятся MAC-адреса (аппаратные) отправителя и получателя, а также информация о типе вложенного протокола верхнего уровня. Она редко будет использоваться, так как имеет практический смысл только при рассмотрении принципов работы протокола ARP (глава 2). В остальных случаях будем подразумевать, что отправитель владеет информацией о MAC-адресе получателя. Для оборудования, имеющего интерфейс Ethernet, аппаратный адрес постоянен и устанавливается на заводе-изготовителе. Каждый производитель имеет свой уникальный префикс в адресе, проверяя который анализатор может выяснить, кем произведен сетевой адаптер (в нашем случае на компьютере-отправителе и компьютере-получателе кадра установлены сетевые карты производства компании Intel — префикс 00:d0:b7). Данный префикс занимает первые 24 бит и назначается организацией IEEE (в англоязычной литературе префикс часто называют vendor code — кодом поставщика). Относительно свежий список префиксов, закрепленных за производителями, можно найти в сети Интернет по адресу http://www.cavebear.com/CaveBear/Ethernet/vendor.html (домашняя страница Карла Ауербаха, сотрудника консультационной фирмы CaveBare). Дополнительно к полям MAC-адресов кадр содержит также и поле Type (тип), идентифицирующее протокол более верхнего уровня, принимавшего участие в инкапсуляции кадра. Для протокола IP его значение равно 0x8000.

Далее представлена информация о протоколах следующих уровней — сетевом (протокол IP), транспортном (протокол TCP) и сеансовом (протокол HTTP).

От пакета к пакету эта информация может меняться, но, как правило, всегда будет присутствовать вложенная дейтаграмма IP, поскольку все примеры будут рассматриваться начиная от сетевого уровня и выше. Но вместо протокола TCP вероятно появление протокола UDP, а за ним информации DNS — все зависит от того, какое приложение сформировало и послало пакет в сеть. В книге упомянутые протоколы будут рассмотрены детальным образом, причем только применительно к их практическому действию.

ТЕРМИНОЛОГИЯ

Далее по тексту под термином «пакет» будем иметь в виду весь объем передаваемой информации, включая заголовки всех «вложенных» протоколов, «кадр» — это историческая терминология сетей Ethernet и в данном контексте этот термин по объему данных приравнивается «пакету» и будет употребляться редко и только для того, чтобы подчеркнуть, что речь идет именно о канальном уровне. «Дейтаграмма» — охватывает часть информации в пакете, принадлежащей протоколу IP (то есть все данные за исключением заголовка «кадра» Ethernet). Вся другая информация протоколов более высоких уровней будет также обобщенно называться «пакетом», но с добавлением ключевого слова, указывающего на то, о какой единице данных идет речь, например пакет DNS или пакет UDP. К сожалению, с определением единицы передаваемых данных (не имеет отношения к MTU) есть некая путаница. Так, очень часто в технической литературе встречается термин «дейтаграмма UDP», но в своем материале автор будет придерживаться описанной схемы наименований, чтобы не вводить лишние термины. Исключением является лишь термин «сегмент» TCP, который уже устоялся при рассмотрении принципов работы этого протокола, хотя, по мнению автора, можно смело говорить «пакет» TCP, ведь в конечном итоге оба термина ссылаются на одно и то же.

Сообщения протокола ICMP

Возвратимся к рассмотрению самого протокола ICMP. Так как этот протокол при своей работе опирается на доставку, осуществляемую протоколом IP, важное место в заголовке дейтаграммы IP отводится полям Source IP (адрес отправителя) и Destination IP (адрес получателя). То есть в распределенной сети дейтаграмма поступает на стек TCP/IP хоста, которому она адресована. Хост анализирует тип вложенного в дейтаграмму протокола, определяет факт использования протокола ICMP и выполняет определенные действия (или не выполняет). Далее следует заголовок сообщения ICMP. И как видно из рис. 4.5, количество полей в заголовке значительно меньше, чем в протоколе IP, что еще раз подчеркивает ограниченную функциональность обсуждаемого протокола.

Протокол ICMP отличается от протоколов TCP и UDP в нескольких аспектах. Эти отличия определяются его целями и расположением в стеке — так, протокол не требует для своей работы номеров портов. Наибольшее же сходство в функциональности наблюдается в указании типа и кода сообщения (message type and code). Это первые два байта в заголовке, которые и идентифицируют функцию, возложенную на конкретное сообщение. Поле Type указывает на тип сообщения, а поле Code содержит код ошибки, интерпретация которого зависит от типа сообщения (табл. 4.1)

Таблица 4.1. Типы сообщений ICMP

Тип	Обозначение	Код
0	Echo Reply (ответ на запрос echo)	0
1	— (не используется)	0
2	— (не используется)	0
3	Destination Unreachable (получатель недостижим)	0 — Net Unreachable (сеть недостижима) 1 — Host Unreachable (хост недостижим) 2 — Protocol Unreachable (протокол недостижим) 3 — Port Unreachable (порт недостижим) 4 — Fragmentation Needed and Don't Fragment was Set (требуется фрагментация, но установлен бит «Не фрагментировать») 5 — Source Route Failed (сбой при маршрутизации от источника) 6 — Destination Network Unknown (требуемая сеть неизвестна) 7 — Destination Host Unknown (получатель неизвестен) 8 — зарезервировано 9 — зарезервировано 10 — зарезервировано 11 — Destination Network Unreachable for Type of Service (для указанного типа сервиса целевая сеть недостижима) 12 — Destination Host Unreachable for Type of Service (для запрошенного типа сервиса целевой хост недостижим) 13 — Communication Administratively Prohibited (взаимодействие запрещено в административном порядке)
4	Source Quench (подавление источника)	0
5	Redirect (перенаправление)	0 — Redirect Datagram for the Network (перенаправление дейтаграммы, адресованной в сеть или подсеть) 1 — Redirect Datagram for the Host (перенаправление дейтаграммы, адресованной хосту) 2 — Redirect Datagram for the Type of Service and Network (перенаправление дейтаграммы для запрошенного типа сервиса или сети) 3 — Redirect Datagram for the Type of Service and Host (перенаправление дейтаграммы для запрошенного типа сервиса или хоста)
6	Alternate Host Address (альтернативный адрес хоста)	0
8	Echo request (запрос echo)	0
9	Router Advertisement (анонсирование маршрутизатора)	0

Тип	Обозначение	Код
10	Router Selection (выбор маршрутизатора)	0
11	Time Exceeded (превышение времени)	0 — Time to Live exceeded in Transit (значение поля TTL превышено) 1 — Fragment Reassembly Time Exceeded (превышено время сборки фрагментов)
12	Parameter Problem (проблема с параметром)	0 — Pointer indicates the error (указание на ошибку) 1 — Missing a Required Option (пропущена требуемая опция) 2 — Bad Length (некорректная длина)
13	Timestamp Request (запрос временной отметки в формате Timestamp)	0
14	Timestamp Reply (ответ на запрос временной отметки)	0
15	Information Request (запрос информации)	0
16	Information Reply (ответ на запрос информации)	0
17	Address Mask Request (запрос маски адреса)	0
18	Address Mask Reply (ответ на запрос маски адреса)	0
19–29	Зарезервированы	
30	Traceroute (трассировка маршрута)	
31	Datagram Conversion Error (ошибка преобразования дейтаграммы)	
32	Mobile Host Redirect (перенаправление мобильного хоста)	
33	IPv6 Where-Are-You («Где я?» — запрос протокола IPv6)	
34	IPv6 I-Am-Here («Здесь» — ответ на запрос)	

Для отправки тех или иных сообщений существует ряд условий, отраженных в документе RFC 792. Так, не посылается сообщение об ошибке ICMP в ответ на получение аналогичного сообщения, чтобы предотвратить возникновение петель. Далее, для фрагментированных дейтаграмм IP сообщения протокола ICMP посылаются только для первого фрагмента. Другое требование заключается в том, что сообщения об ошибках никогда не посылаются в ответ на получение дейтаграммы с групповым или широковещательным адресом, во избежание вещательного

«шторма» трафика в сети. Кроме того, сообщение об ошибке отсутствует в случае получения дейтаграммы с нулевым адресом отправителя, адресом 127.0.0.1 (адрес обратной связи, loopback), широковещательным и групповым адресом. Интересно отметить, что при поступлении на хост сообщения ICMP неизвестного типа это сообщение должно игнорироваться (чтобы снизить вероятность некорректного влияния на хост ложных сообщений).

Все сообщения протокола можно условно разделить на две группы — сообщения об ошибках (error messages — табл. 4.2) и запросы (query messages — табл. 4.3). Первая группа, как несложно понять из названия, служит для извещения о произошедших ошибках, которые препятствовали доставке дейтаграммы (рекомендуется использовать при возникновении некратковременных проблем доставки — так называемых non-transient delivery problem. Важно, что каждое сообщение об ошибке включает в себя IP-заголовок (от 20 до 60 байт) и по крайней мере 8 байт данных дейтаграммы, послужившей причиной отправки сообщения, то есть количество присоединяемых байтов исходной дейтаграммы может быть больше. Соответственно, учитывая сам заголовок ICMP, размер получаемых вами сообщений об ошибках будет варьироваться от 36 до 72 байт.

Таблица 4.2. Сообщения об ошибках ICMP

Сообщение об ошибке	Тип
Destination Unreachable (получатель недостижим)	3
Source Quench (подавление источника)	4
Redirect (перенаправление)	5
Time Exceeded (превышение времени)	11
Parameter Problem (проблема с параметром)	12

На рис. 4.5 показан пример сообщения протокола ICMP об ошибке (Destination Unreachable — получатель недостижим). Формат сообщений об ошибке для различных типов последних мало чем отличается. Так, непременно присутствуют поля Type (тип — 1 байт), Code (код — 1 байт) и Checksum (контрольная сумма — 2 байт). Кроме того, в сообщение обязательно включаются заголовок исходной дейтаграммы (20–60 байт) и первые 64 бит (8 байт) данных.

Анализируя рис. 4.5, можно увидеть, что перехваченное сообщение об ошибке содержит весь заголовок IP и еще 8 байт данных дейтаграммы, следующей за заголовком. В нашем случае в дейтаграмму IP был вложен пакет TCP (начиная с заголовка). Но из всего заголовка TCP было взято только 8 байт. В принципе, для целей анализа причины сбоя подобного объема информации должно быть достаточно. В других типах сообщений об ошибках может присутствовать другая, специфичная для них информация.

Сообщения запросов протокола ICMP (табл. 4.3) используются для посылки запросов и анализа полученных ответов в целях определения общих характеристик сети (как например, доступности определенного хоста или для измерения сетевой задержки — network latency). Для сообщения запросов общими являются поля Type, Code, Checksum, Identifier (2 байт; идентификатор — для селекции

сообщений, посланных разным хостам) и Sequence Number (2 байт; номер последовательности — для селекции нескольких сообщений, посланных одному хосту). Размер сообщений запросов отличается в зависимости от их типов. Если размер заголовка постоянен — 8 байт, то общая длина дополнительных полей является переменной.

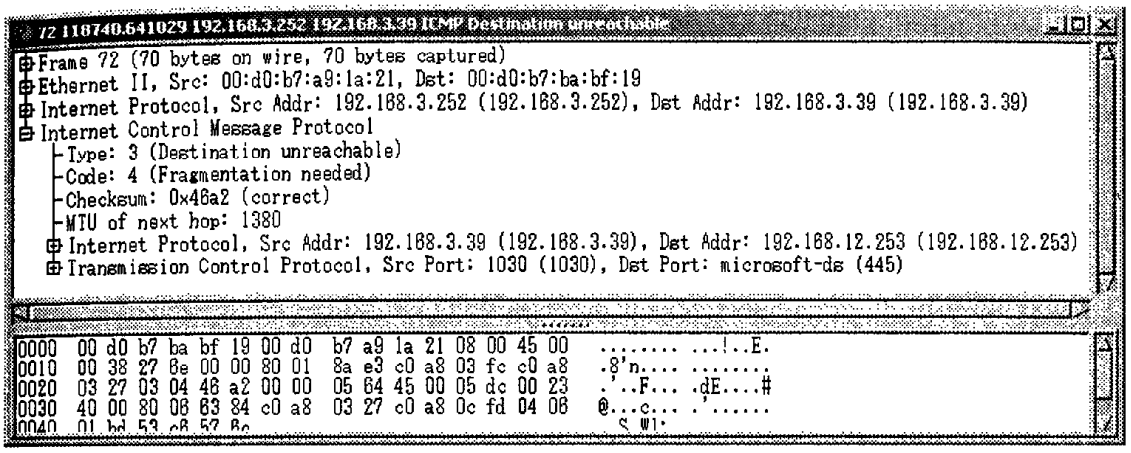


Рис. 4.5. Пример сообщения об ошибке

Таблица 4.3. Сообщения запросов ICMP

Сообщение	Типы
Echo request (запрос echo), Echo Reply (ответ на запрос echo)	8, 0
Timestamp Request (запрос временной отметки), Timestamp Reply (ответ на запрос временной отметки)	13, 14
Information Request (запрос информации), Information Reply (ответ на запрос информации)	15, 16
Address Mask Request (запрос маски адреса), Address Mask Reply (ответ на запрос маски адреса)	17, 18

Сообщение ICMP echo request

Каждое сообщение протокола ICMP служит для определенных целей и посылается при образовании определенных условий в сети. Рассмотрим наиболее часто встречающиеся типы сообщений и проанализируем причины их возникновения. Начнем с разбора, пожалуй, наиболее «популярного» сообщения echo request (запрос echo — рис. 4.6), генерируемого командой ping. Интересно отметить, что данный тип сообщения является общим для всех операционных систем — документ RFC 1122 требует, чтобы каждый хост предоставлял возможность пользователю отправить запрос echo другому хосту (подразумевается наличие команды или программы — для других сообщений подобного требования нет).

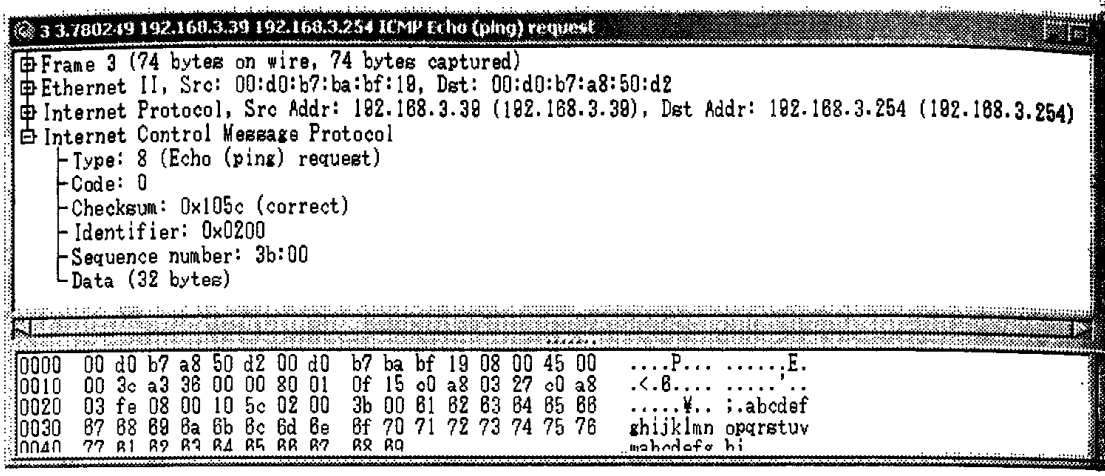


Рис. 4.6. Пример сообщения ICMP echo request

Присмотревшись, можно увидеть схему инкапсуляции и 2 байта, идентифицирующие само сообщение (на рис. 4.6 они отмечены как type: 8 и code: 0). Поле «Идентификатор» (Identifier) служит для идентификации запросов, направленных различным хостам в сети, а поле номера последовательности (Sequence number) задействуется в случае отправки нескольких запросов одному хосту. Так, в документе RFC 792 оговорено, что «поля идентификатора и номера последовательности *могут* (*may*) быть использованы отправителем запроса echo для связывания с поступающими ответами. Например, идентификатор может использоваться как порт в протоколах TCP или UDP для идентификации различных сеансов, а номер последовательности может увеличиваться при каждом отправленном запросе. В ответе возвращаются те же самые значения, что были в запросе». В сказанном несложно убедиться на практике путем одновременного запуска двух команд ping из-под операционной системы Microsoft Windows 2000 Professional Rus (SP3) с указанием двух различных целевых адресов (адрес первого получателя запросов 192.168.3.254, второго — 192.168.3.253).

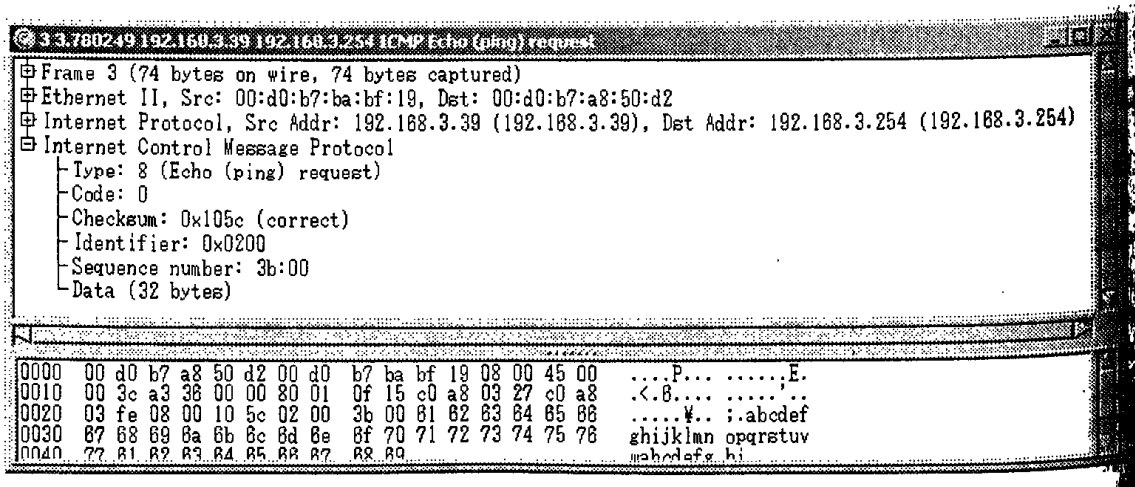


Рис. 4.7. Пакет первого сеанса ping

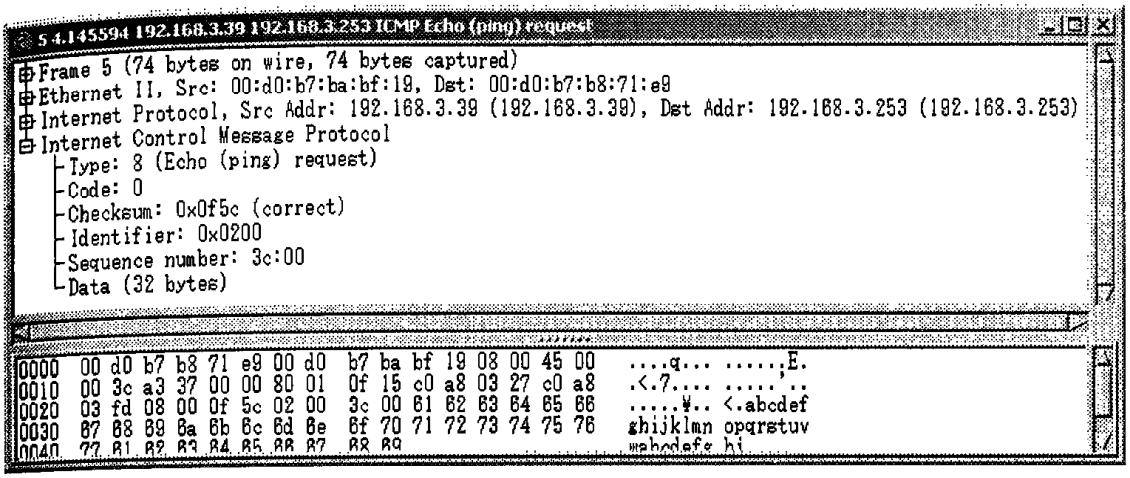


Рис. 4.8. Пакет второго сеанса ping

Как видно из примеров (рис. 4.7 и 4.8), значение идентификатора одинаково для обоих сеансов (0x0200), несмотря на предыдущую выдержку из документа RFC (скорее всего цитата подразумевала рекомендацию, что подчеркивалось ключевым словом may). При этом, для того чтобы отличить запросы друг от друга, используется поле номера последовательности — для каждого запроса значение этого поля уникально, а различие между номерами для двух сеансов равно 0x100 (256 в десятичной системе). Для первого запроса значение номера последовательности 0x3b00 (или 15 104), для второго (но другого хоста) — 0x3c00 (или 15 360). Разница между ними как раз и равна 256 в десятичной системе. Характерно то, что для всех остальных запросов значение этого поля увеличивается на 256, причем для каждого запроса **echo request** даже в рамках одного сеанса **ping** (в операционных системах Microsoft Windows команда генерирует 4 запроса ICMP echo request).

Интересно отметить, что помимо идентификации запросов поле идентификатора может служить источником информации о типе операционной системы, с которой данный запрос был отправлен (особенно это касается семейства Microsoft Windows в свете предыдущего эксперимента).

Так, если запрос проистекал из операционной системы Microsoft Windows NT (с установленным пакетом обновлений Service Pack 6) то поле будет иметь значение (или начинаться) с 0x100, а для семейства Microsoft Windows 2000 — с 0x200 (десятичное 512) либо 0x300 в зависимости от установленного пакета обновлений. Для начального заполнения этого поля в операционных системах Microsoft используется некая константа, различная для разных типов систем. То есть если вы, перехватив запрос ICMP echo, скажем, поступивший на внешний интерфейс вашего защитного экрана, увидите, что поле идентификатора равно 256, 512 или 768 (для Microsoft Windows ME), то можно с высокой долей уверенности утверждать, что запрос был инициирован с одной из операционных систем Microsoft. Другой интересной особенностью является то, что для указанных операционных систем данное значение будет повторяться и для последующих запросов, в то время как для ОС семейства UNIX рассматриваемое поле заполняется на основании идентификатора процесса (Process ID, PID) и может меняться от случая к случаю.

Кроме того, как видно из предыдущих перехватов, четко прослеживается зак
симость изменений поля номера последовательности для разных сеансов. П
начальной загрузке системы Windows и выполнении первой команды ping з
чение этого поля устанавливается равным 0x100 и будет увеличиваться на 0x1
для каждого следующего запроса. В принципе, можно путем перехвата и деле
ем значения поля в перехваченном запроса на число 0x100 узнать, сколько
просов инициировалось с момента загрузки. Например, если вернуться к рис. 4
то деление значения 15 104 на 256 даст в результате 59. То есть в момент пер
хвата система сгенерировала 58 запросов echo request. Эксперименты показа
что самый первый запрос echo request посылается серверу — контроллеру доме
Microsoft Active Directory (естественно, это относится к ситуации, когда ра
чая станция входит в домен Active Directory). Для каких целей так сделано, ос
ется до конца не ясным, так как запрос посылается уже после того, как сер
был опрошен по протоколу LDAP и с ним было установлено соединение TCP.

Возвратимся к рассмотрению общей схемы работы запроса echo request. В то
случае, если на запрос ICMP echo request поступает ответ echo reply, то можн
сделать вывод о том, что получатель запроса активен в сети и доступен. Но сле
дует отметить, что некоторые защитные экраны способны генерировать ответ
за получателей. Если ответ не поступает, то логично предположить, что либ
хост неактивен, либо защитный экран настроен на блокирование входящих и
вне запросов с последующим «молчаливым» их удалением.

Если используется операционная система Microsoft Windows с установленны
стеком TCP/IP, то для формирования сообщения echo request, как правил
привлекается популярная команда ping с указанием адреса получателя запро
Эта команда очень популярна и часто описывается в технической литературе.

Поэтому интересно будет остановиться на таком понятии, как ICMP swe
(трудно переводимый термин, поэтому оставим его как есть). Также мож
встретиться термин ping sweep, что означает одно и то же. Если для небольши
сетей команда ping приемлема, то для крупных распределенных сетей подобны
метод опроса работающих хостов может показаться утомительным. В этом слу
чае подойдут инструменты, которые организуют массовый опрос хостов в сет
практикуя принцип ping sweep, который реализуется отправкой множества з
просов различным хостам, с последующим анализом результатов. Для реализа
ции этой возможности неплохим вариантом будет популярная программа nma
(http://www.insecure.org), запущенная, например, со следующими параметрам
nmap -sP -PI 192.168.3.1-254. Важно отметить, что подобный массированны
опрос работающих устройств в сети обнаруживается системами IDS.

```
C:\Program Files\nmapNTsp1\Nmapnt>nmapnt -sP -PI 192.168.3.1-254
Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org ( www.insecure.org/nmap/ )

Host 192.168.3.3 appears to be up.
Host 192.168.3.6 appears to be up.
Host 192.168.3.7 appears to be up.
Host 192.168.3.11 appears to be up.
... (удалено для краткости)
Nmap run completed -- 254 ip addresses (74 hosts up) scanned in 32 seconds
```

При выполнении команды nmap, в данном конкретном случае — версии программы для платформы Windows, адаптированной компанией eEye (www.eeye.com), параметры –sP и -PI указывают программе производить сканирование только с помощью запросов ICMP. Следует отметить, что в некоторых случаях сканируемые хосты располагаются за защитным экраном, который блокирует входящие запросы ICMP echo request. Тогда следует отказаться от параметра –PI и программа nmap будет в дополнение к запросам ICMP осуществлять TCP-сканирование посредством отправки сегмента с установленным флагом ACK и с портом, имеющим номер 80 (значение по умолчанию). Как будет показано далее в главе 5, рассматривающей особенности протокола TCP, если в ответ поступает сегмент с установленным флагом RST, значит хост активен.

В терминологии протокола ICMP нет таких понятий, как «клиент» и «сервер». Например, если хост получает сообщение этого протокола, указывающего на определенную ситуацию, он может выполнить какие-то определенные внутренние действия, но не обязан делать что-либо для обратного взаимодействия с отправителем — ICMP не дает никаких гарантий относительно доставки сообщения.

Протокол ICMP также примечателен тем, что хосту не требуется поддерживать какой-либо сервис или порт. Практически любая сетевая операционная система может отвечать на запросы ICMP echo request (команда ping). Говоря точнее, в соответствии с рекомендациями, данными в документах RFC, каждое сетевое устройство (хост или маршрутизатор) должно обладать функциональностью, обеспечивающей ответное сообщение echo reply на запрос echo request. Сложности возникают при необходимости как раз изменить обычное поведение протокола, реализованного в той или иной операционной системе — например, отключить отправку ответного сообщения ICMP echo reply на запрос echo request (результат команды ping x.x.x.x, где x.x.x.x — адрес удаленного хоста).

Дополнительной уникальной возможностью протокола является поддержка широковещательного трафика. Так, если протокол TCP опирается исключительно на взаимодействие в стиле «тет-а-тет» между клиентом и сервером, не принимая во внимание другие хосты в сети, то протокол ICMP не столь категоричен. И именно указанная возможность породила такую атаку, как Smurf, которая будет рассматриваться в главе 8. Кстати, читатель может сам попробовать выполнить команду ping x.x.x.255 в своей сети (предполагается, что в сети используется адрес класса C). Например, автор увидел следующие результаты:

```
C:\>ping -n 10 192.168.3.255
Обмен пакетами с 192.168.3.255 по 32 байт:

Ответ от 192.168.3.255: число байтов=32 время=10мс TTL=64
Ответ от 192.168.3.255: число байтов=32 время<10мс TTL=64
Ответ от 192.168.3.255: число байтов=32 время<10мс TTL=64
Ответ от 192.168.3.255: число байтов=32 время<10мс TTL=255
Ответ от 192.168.3.255: число байтов=32 время<10мс TTL=64
Ответ от 192.168.3.255: число байтов=32 время<10мс TTL=64
...
```

Все компьютеры в сети, где выполнялась данная команда, автоматически получают адреса (работает служба DHCP) сети класса C — 192.168.3.0/24. Приведен-

ная команда десять раз (параметр –n 10) отправляет запрос ICMP echo reque
Обратите внимание, что в дейтаграмме IP в качестве адреса получателя устанавл
вается 192.168.3.255 (рис. 4.9), являющийся указанием на то, что это направле
ное широковещание (напомню, что есть также и ограниченное широковещани
а в кадре сети Ethernet адрес получателя равен FF:FF:FF:FF:FF:FF (сообщен
достигает всех устройств в рамках сети).

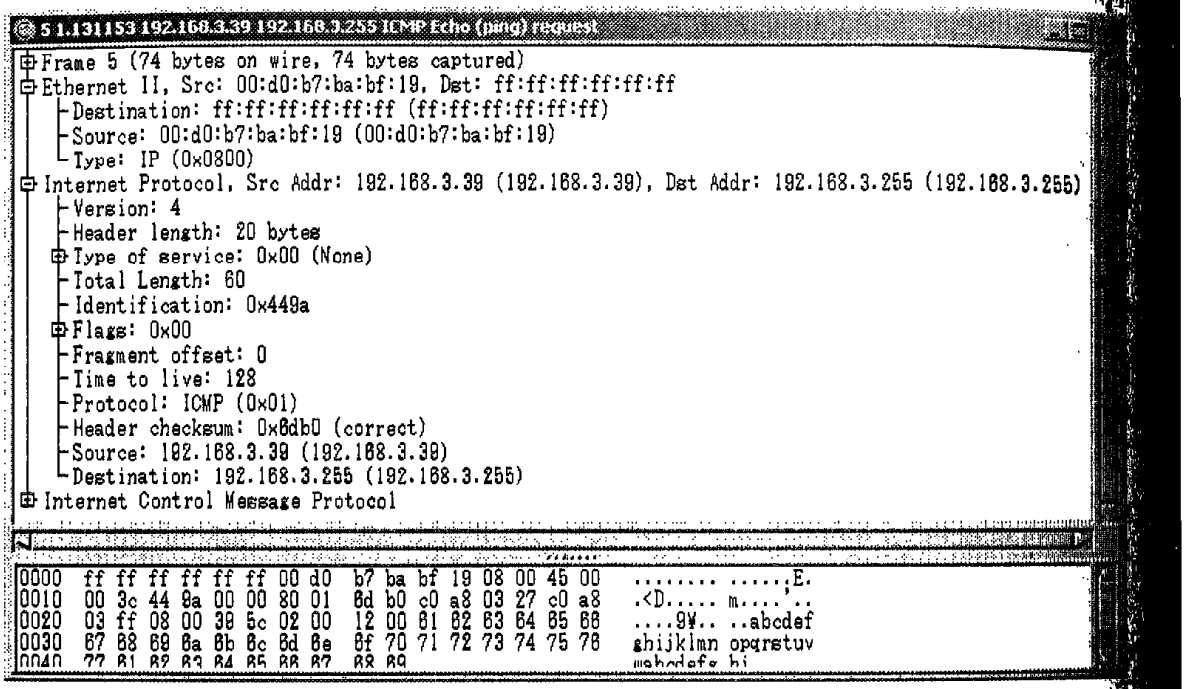


Рис. 4.9. Широковещательный запрос echo request

В ответ на одно подобное сообщение ICMP отправитель получает несколько откликов. Наиболее удобно отобразить результат в следующем виде:

```
39 6.847163 192.168.3.39 192.168.3.25    ICMP    Echo (ping) request
40 6.848198 192.168.3.65 192.168.3.39    ICMP    Echo (ping) reply
41 6.849207 192.168.3.245192.168.3.39    ICMP    Echo (ping) reply
42 6.849406 192.168.3.247192.168.3.39    ICMP    Echo (ping) reply
43 6.849621 192.168.3.248192.168.3.39    ICMP    Echo (ping) reply
44 6.878286 192.168.3.74 192.168.3.39    ICMP    Echo (ping) reply
```

Видно, что в данном конкретном случае на один запрос поступило пять ответов ICMP echo reply. Вообще говоря, так как атака Smurf была придумана сравнительно давно, многие производители включили в свои операционные системы защиту от подобных запросов, точнее расширения, препятствующие услужливому реагированию на подобные широковещательные запросы. В сегменте сети, в котором выполнялась подобная команда, располагается около 130 компьютеров, работающих под управлением Microsoft Windows 2000, Windows 98 и Windows NT. Ни одна из этих операционных систем не откликнулась на запрос, а ответ прислали маршрутизатор Cisco Systems 805 (программное обеспечение c805-y6-mw.120–4.XM), коммутаторы Hewlett-Packard ProCurve Switch 4000M

обладающие IP-адресами для управления через сеанс Telnet, и источники бесперебойного питания компании APC (также имеющие интерфейс для подключения к локальной сети).

Важно отметить, что подобный подход к определению работающих хостов применим только к тем из них, которые работают под некоторыми версиями операционной системы UNIX (насколько известно автору, отвечать будет, например, RedHat Linux 6.2 Kernel 2.2.14 или, например, Solaris 2.8), в то время как все операционные системы Microsoft проигнорируют отправителя. Кстати, если сослаться на стандарты, а именно на документ RFC 1122, содержащий требования к хостам в сети Интернет, то в нем можно увидеть утверждение, что при получении запроса echo request с широковещательным (или групповым) адресом хосты вправе его игнорировать.

Обратите внимание на то, что MAC-адрес получателя запросов широковещательный (FF:FF:FF:FF:FF:FF). Даже если сеть построена исключительно на основе коммутаторов, запрос дойдет до всех локальных хостов, так как по умолчанию коммутаторы передают широковещательный адрес. Кроме того, хочу отметить еще одну особенность, связанную с широковещательным запросом ICMP echo request. В настоящее время многие компании предпочитают использовать модемы серии xDSL, снабженные портом Ethernet. Подобные модемы работают как мосты и, соответственно, таким же образом будут фильтровать широковещательный трафик.

Любопытным свойством протокола ICMP является возможность использования запросов ICMP echo с указанием размера поля данных. Если вернуться к предыдущим рисункам, можно увидеть, что в заголовках есть поле Data, которое, как вы уже догадались из названия, говорит об объеме данных в сообщении протокола. Для сообщений об ошибках протокола поле Data будет содержать часть оригинальной дейтаграммы IP, для которой это сообщение было сформировано. Длина поля данных равна длине дейтаграммы IP, за вычетом ее заголовка.

В качестве примера здесь можно выполнить следующую команду ping:

```
C:\>ping -l 5000 192.168.3.255
Обмен пакетами с 192.168.3.255 по 5000 байт:

Ответ от 192.168.3.255: число байтов=5000 время=30мс TTL=64
Ответ от 192.168.3.255: число байтов=5000 время=20мс TTL=64
Ответ от 192.168.3.255: число байтов=5000 время=10мс TTL=64
...
```

В приведенной команде ping параметр –l 5000 определяет как раз объем посылаемых данных в запросе (5000 байт). На сетевом уровне это выглядит следующим образом:

```
11 3.057392 192.168.3.39 192.168.3.255 ICMP Echo (ping) request
12 3.057448 192.168.3.39 192.168.3.255 IP Fragmented IP protocol (proto=ICMP
   0x01, off=1480)
13 3.057473 192.168.3.39 192.168.3.255 IP Fragmented IP protocol (proto=ICMP
   0x01, off=2960)
14 3.057496 192.168.3.39 192.168.3.255 IP Fragmented IP protocol (proto=ICMP
   0x01, off=4440)
```

Так как протокол ICMP использует дейтаграммы IP, а последние ограни
ны максимальным размером кадра сети Ethernet (имея в виду наш конкретн
случай), то выполняется процедура фрагментации, которая ниже будет рассм
рена более подробно. Сейчас нужно заострить внимание на том, что в результа
фрагментации в сети от отправителя запроса передаются четыре кадра Ether
каждый из которых несет в себе дейтаграмму IP и сообщение ICMP. Поскольк
в команде ping вновь указывался широковещательный адрес, MAC-адрес отпр
вителя всех этих четырех кадров установлен в FF:FF:FF:FF:FF:FF.

Все хосты в сети получают такой запрос, но как уже было отмечено, не в
отвечают. Хост с адресом 192.168.3.65 получает все четыре кадра, обнаружива
что это фрагменты одной дейтаграммы, для которой выполнялась фрагмент
ция, собирает в памяти фрагменты в одно целое, а затем, анализируя заголово
определяет, что тело дейтаграммы несет сообщение ICMP echo request и, след
заложенной в него логике, отправляет ответ. Еще раз поставлю ударение на то
что «эхо» содержит в себе тот же объем данных, что был в запросе. В результа
в ответный путь также посылаются четыре кадра, возникающие после процед
ры еще одной фрагментации:

```
66 3.077251 192.168.3.65 192.168.3.39 ICMP Echo (ping) reply
67 3.077278 192.168.3.65 192.168.3.39 IP Fragmented IP protocol (proto=ICMP
   0x01, off=1472)
68 3.077290 192.168.3.65 192.168.3.39 IP Fragmented IP protocol (proto=ICMP
   0x01, off=2944)
69 3.077298 192.168.3.65 192.168.3.39 IP Fragmented IP protocol (proto=ICMP
   0x01, off=4416)
```

Вообще говоря, при ответе на запрос echo request получатель ограничивается
только лишь изменением типа сообщения на echo reply и пересчетом контроль
ной суммы заголовка новой дейтаграммы, а остальные поля остаются в непри
косновенности.

Именно на данной особенности протокола ICMP или, точнее говоря, на его
реализации во многих операционных системах играет атака Smurf. Легко понять
что посылая один запрос размером 5000 байт, нам следует ожидать два (или
более) ответа, каждый объемом в те же 5000 байт. К счастью, не все хосты в сети
радушны и отзывчивы, а лишь некоторые. В противном случае в сети из ста
компьютеров на один запрос в 5000 байт пришли бы ответы с совокупным раз
мером 100 × 5000 байт, или 500 Кбайт! Ниже атака Smurf будет рассмотрена
подробнее, но уже сейчас можно сделать предварительный вывод о ее послед
ствиях в случае, если атакующий направил в крупную распределенную сеть
большой запрос по относительно медленному каналу связи — ответ займет его
полностью.

Посмотрите на первый кадр запроса ICMP echo (см. рис. 4.10). Видно, что
поле данных сообщения ICMP равно 1472 байт. Наблюдается соответствие
теории, согласно которой максимальный размер кадра в сети Ethernet равен
1500 (или 1514, если считать 14 байт заголовка самого кадра) байт, заголово

дейтаграммы IP занимает 20 байт, заголовку сообщения ICMP echo request отводится 8 байт. Это объясняет наличие четырех кадров в сети (или одной дейтаграммы IP и трех ее дополнительных фрагментов) от одного запроса размером в 5000 байт — первые три кадра имеют размер 1514 байтов, а третий кадр — 602 байт. В итоге по сети передается 5144 байт (3 × 1514 + 602 = = 5144).

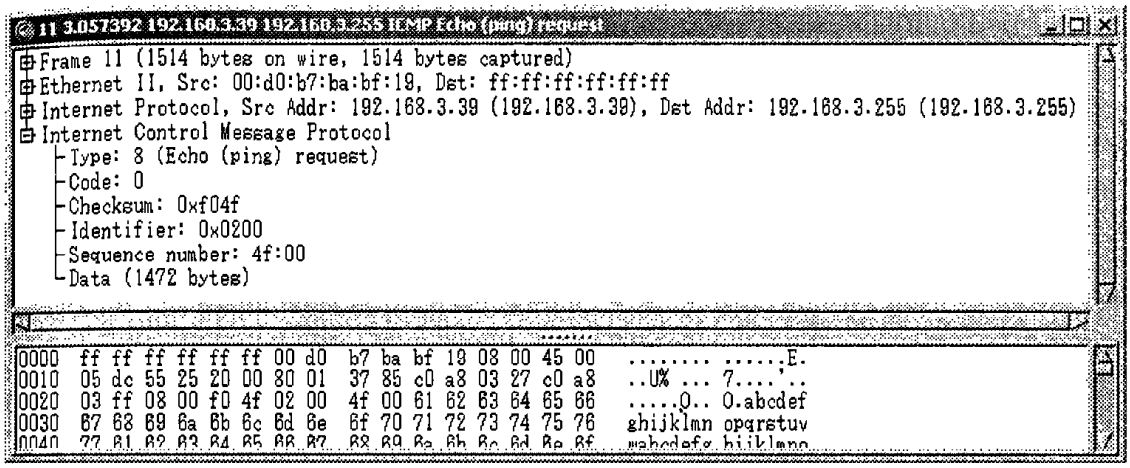


Рис. 4.10. Сообщение ICMP echo request с максимальным полем Data

Уже упоминавшийся документ RFC 792 не определяет, как много данных должно посылаться с запросом ICMP echo request. В нем есть только фраза «добавить некоторые данные», но при этом какие именно данные и в каком количестве должны быть добавлены, не указывается. Тем самым возникает потребность в рассмотрении следующих вопросов — смещение начала порции данных относительно заголовка сообщения ICMP, размер этой порции и содержимое.

В операционных системах семейства Microsoft Windows размер поля данных по умолчанию (если явно не указано другое значение) составляет 32 байт, а общий размер дейтаграммы равен 60 байт (74–14; размер кадра для стандартного сообщения ICMP echo request за вычетом заголовка Ethernet). При этом поле данных начинается сразу за заголовком сообщения ICMP. Для большинства UNIX-систем размер поля данных равен 56 байт, а итоговая дейтаграмма IP включает в себя 80 байт, где первые 8 байт в поле данных служат для вычисления значения времени кругового обращения (Round-Trip Time — RTT). Аналогично, может отличаться и полезная нагрузка поля данных (напомню, что стандарт не указывает на определенные шаблоны заполнения). Так, для систем Microsoft шаблон заполнения представляет собой повторяющийся набор символов «abcdefghijklmnopqrstuvw». На рис. 4.11 показан запрос с акцентом на повторяющееся содержимое поля Data. Кстати, именно на примере такого простого содержимого интересно и наглядно наблюдать процедуру шифрования передаваемых данных (или их сжатия), например с помощью протокола PPTP (глава 10).

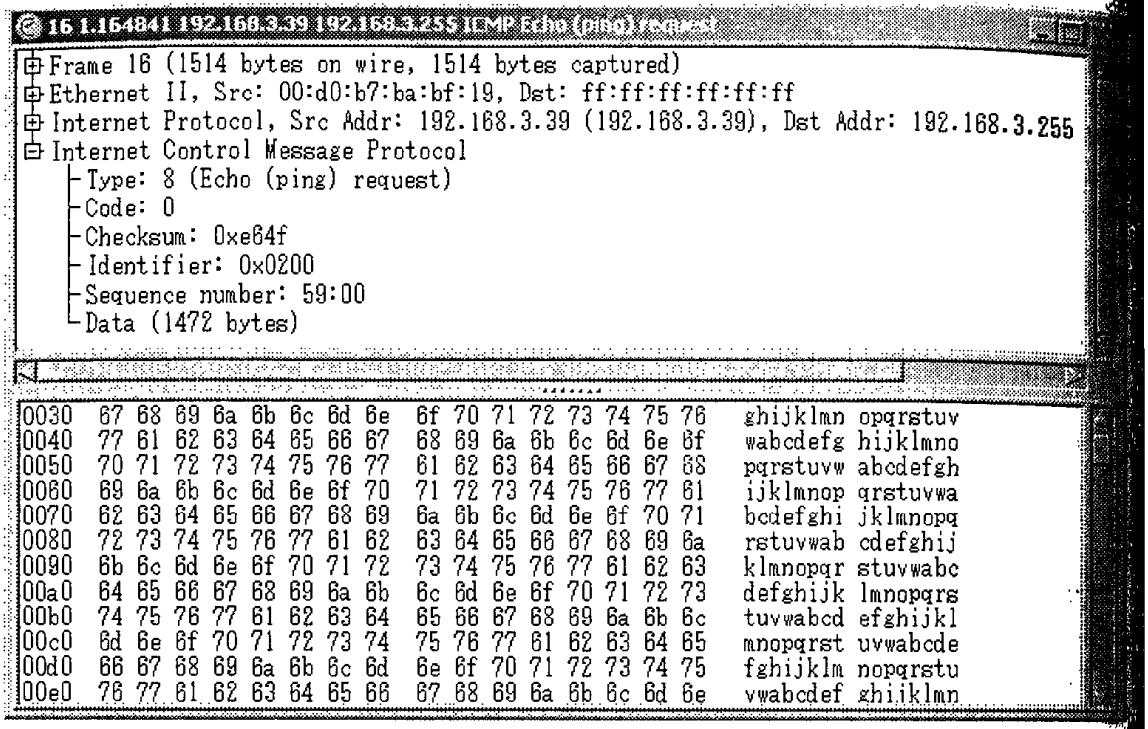


Рис. 4.11. Содержимое поле Data сообщения ICMP echo request

Сообщение ICMP source quench (подавление источника)

Протокол ICMP применим в ситуации, когда, например, получатель испытывает трудности при обработке объемного трафика от отправителя. В этом случае одним из способов избежать проблем может быть обращение к отправителю с просьбой снизить скорость отправки данных — с помощью посылки сообщения ICMP source quench (подавление источника). При этом поле Type устанавливается в значение 4, а поле Code всегда равно нулю. Сообщение source quench может генерироваться как маршрутизатором, так и конечным хостом. Первый случай означает, что маршрутизатор не имеет свободного буферного пространства, необходимого ему для постановки дейтаграммы в очередь (глава 6 — настройка очередей на маршрутизаторах) для передачи по каналу связи. Если обратиться к документу RFC 1812, то там можно найти информацию о том, что маршрутизатору не следует (should not) генерировать эти сообщения. Если они все же посылаются отправителю, то он должен иметь возможность ограничить скорость, с которой происходит их отправка. В противном случае не исключается, что попытки маршрутизатора уйти от перегрузки будут напрасными. Если же сам маршрутизатор получил такое сообщение, то он вправе его игнорировать. В том случае, когда сообщение генерируется получателем информации, это означает, что дейтаграммы от отправителя поступают слишком быстро — быстрее, чем они могут быть обработаны.

Получатель сообщения source quench обязан снизить скорость отправки информации. После получения этого сообщения сигнал о его получении должен быть передан на транспортный уровень протоколам TCP или UDP. Протокол TCP в состоянии снизить скорость отправки сегментов не прерывая установленное соединение с получателем. На рис. 4.12 показан пример сети (реальной), в которой может задействоваться сообщение source quench. Внутренний хост работает в сети Интернет с неким сервером (например, по протоколу HTTP). Все взаимодействие осуществляется через защитный экран (как именно построен защитный экран, сейчас не суть важно). За защитным экраном располагается маршрутизатор Cisco 1720, который с помощью модема «Натекс» FlexDSL подключается к поставщику услуг Интернета. В какой-то момент, ввиду интенсивного трафика, некий промежуточный маршрутизатор в сети Интернет переходит в состояние перегрузки и в ответ на поступление дейтаграмм IP посылает сообщение source quench с просьбой снизить интенсивность поступающего трафика. Сообщение отправляется по адресу отправителя трафика и, например, если защитный экран выполняет трансляцию адресов (технология NAT – глава 6), то сообщение ICMP поступит на внешний интерфейс защитного экрана.

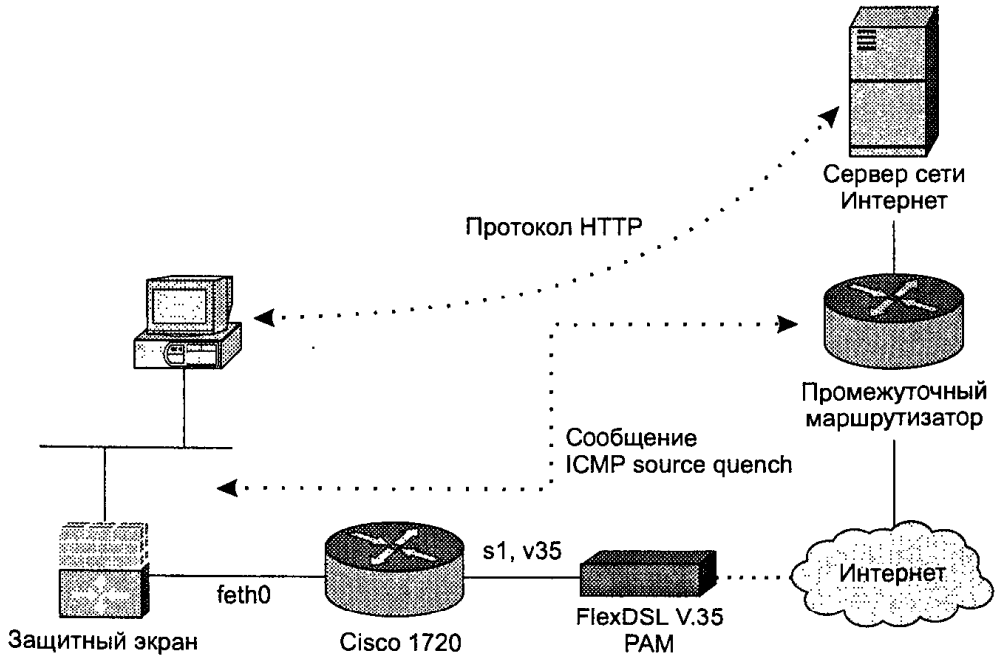


Рис. 4.12. Пример ситуации для формирования сообщения ICMP source quench

Рассмотрим пример перехваченного сообщения, иллюстрирующего описанную ситуацию. На рис. 4.13 видно, что устройство с адресом 194.190.134.35 (предположительно, это промежуточный маршрутизатор) отправило по адресу внешнего интерфейса защитного экрана (193.125.203.1) сообщение ICMP source quench, которое содержит оригинальные заголовки IP и TCP.

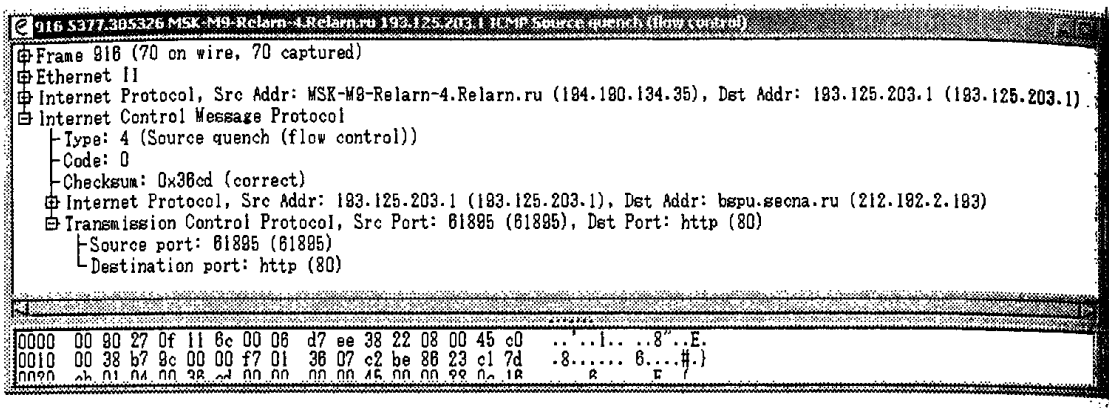


Рис. 4.13. Пример сообщения ICMP source quench

Анализируя их, можно предположить, что некий внутренний хост работал с WWW-сервером в сети (с адресом 212.192.2.193) через защитный экран и в какой-то момент промежуточный маршрутизатор стал испытывать проблемы с обработкой трафика. Отмечу, что в перехваченном пакете адрес отправителя исходной дейтаграммы 193.125.203.1 — это адрес внешнего интерфейса защитного экрана, который выполняет процедуру трансляции адресов для всех хостов, работающих через него в сети Интернет. После того как маршрутизатор (устройство с адресом 194.190.134.35) определил факт состояния перегрузки, он попробовал «подавить» интенсивность поступающего трафика, отправив сообщение source quench с указанием заголовка исходной дейтаграммы, чтобы отправитель сумел понять, интенсивность трафика какого сеанса следует снизить. Самым простым способом проследить отправителя этого сообщения, точнее определить его место в цепочке до получателя трафика, является запуск команды tracert (в операционных системах Microsoft).

```
C:\>tracert 212.192.2.193
Трассировка маршрута к bspu.secna.ru [212.192.2.193]
с максимальным числом прыжков 30:
...
  7    10 ms    20 ms    10 ms  MSK-KIAE-13.Relcom.EU.net [193.125.152.13]
  8    10 ms    20 ms    10 ms  MSK-KIAE-Relarn-4.Relarn.ru [194.85.73.10]
  9    10 ms    20 ms    10 ms  MSK-M9-Relarn-1.Relarn.ru [194.226.29.129]
 10    30 ms    20 ms    20 ms  MSK-M9-Relarn-4.Relarn.ru [194.190.134.35]
 11     *     1813 ms  1732 ms  ARCI-Relarn.Relarn.ru [194.226.29.66]
 12     *      972 ms  1231 ms  secna-relarn-e1-e0-10Mb.secna.ru [212.192.0.9]
...
```

Дальнейшие действия после получения сообщения ICMP source quench в рассматриваемом примере целиком зависят от защитного экрана. В идеале, он обязан переслать поступившее на его внешний интерфейс сообщение внутреннему хосту, который инициировал его отправку.

Важно отметить, что если сообщение ICMP source quench посылается, то оно должно иметь в своей дейтаграмме поле IP Precedence установленным в аналогичное значение той дейтаграммы, которая послужила причиной отправки этого

сообщения. Забегая вперед, отмечу, что для других сообщений об ошибках протокола ICMP (как то destination unreachable, redirect, time exceeded и parameter problem) они в своих дейтаграммах могут иметь значения 6 (INTERNETWORK CONTROL) или 7 (NETWORK CONTROL), то есть заполнение поля IP Precedence дейтаграммы оставляется на усмотрение отправителя сообщения.

В процессе анализа трафика для подбора материала данной главы на границе внутренней сети и сети Интернет автор столкнулся с другим вариантом сообщения ICMP source quench, который немного не вписывается в рамки общепринятой теории (рис. 4.14).

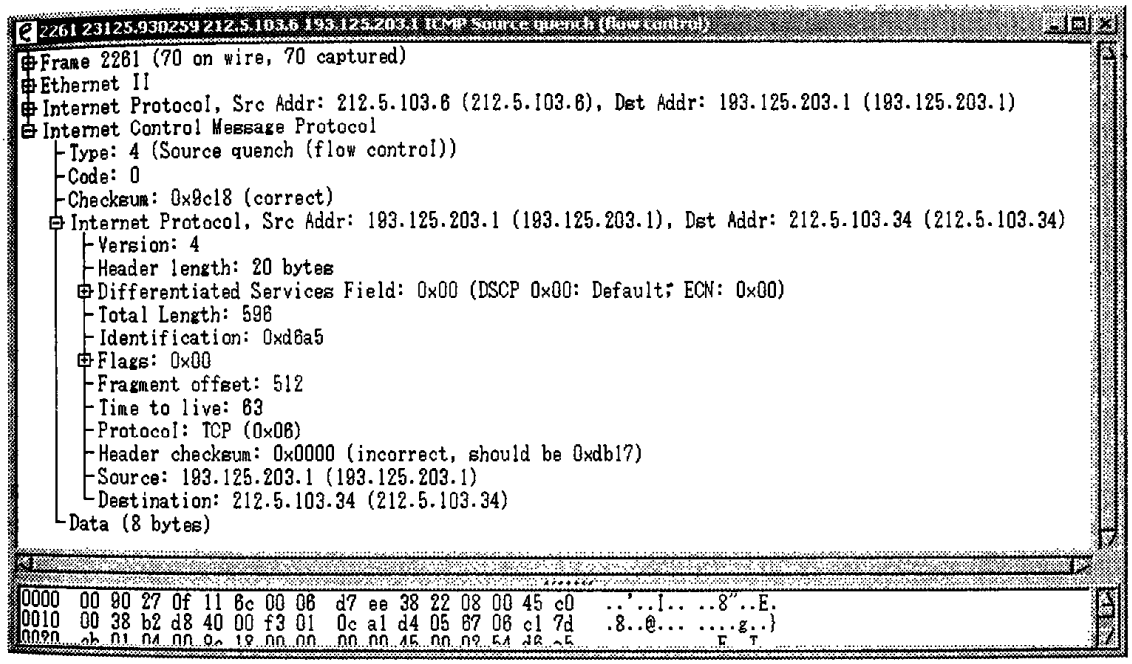


Рис. 4.14. Пример сообщения с некорректной контрольной суммой

На первый взгляд вроде бы все правильно — через защитный экран (или прямо с него) была отправлена дейтаграмма IP, в ответ на которую прислали сообщение с просьбой снизить интенсивность отправки. Но обратите внимание на поле контрольной суммы заголовка в оригинальной дейтаграмме — оно равно 0x0000, чего при нормальных условиях быть не должно. Трассировка до исходного получателя показала, что с большой долей вероятности генератором этого сообщения является маршрутизатор (по крайней мере устройство ответило на запрос Telnet) в пути до получателя. Однако логичнее было бы предположить, что в ответ на дейтаграмму с некорректным заголовком будет послано сообщение ICMP parameter problem.

Если вы установите анализатор пакетов, настроенный на перехват рассматриваемого сообщения ICMP, то увидите, что в большинстве случаев оно инициируется промежуточными маршрутизаторами и, как правило, в ответ на запросы протокола HTTP (TCP-порт назначения 80). Если говорить конкретно о маршрутизаторах компании Cisco Systems, то они отправляют сообщения ICMP source

quench только при отсутствии буферного пространства, необходимого для хра ния пакета в очереди. Если маршрутизатор не в силах поместить пакет в очер он генерирует такое сообщение и регистрирует сброс пакета. Сказанное от сится к версиям Cisco IOS Software Releases до 11.3; маршрутизаторы, управл мые следующими (более новыми) версиями (11.3 и 12.0), не поддерживают о санную функциональность.

Сообщение ICMP destination unreachable

Протокол ICMP может также использоваться маршрутизатором для инф мирования отправителя данных о возникновении различного рода проблем. пример, он может послать сообщение destination unreachable (получатель не стижим). Получение этого сообщения будет означать, что отправитель пытает передать через маршрутизатор трафик, который настроен на блокирование с пом щью списков управления доступом (access lists). В этой ситуации именно ма шрутизатор выступает в роли отправителя сообщения destination unreachab Кроме того, маршрутизатор вправе известить отправителя о том, что получате не может принять его трафик. Например, если в распределенной сети получате не активен, он, естественно, не в состоянии ответить, а маршрутизатор имеет пра известить об этом отправителя. Следует отметить, что хотя подобное поведен маршрутизатора и помогает отправителю, оно попадает под разряд инструмент для проведения негативных действий в сети, например, предоставляя «доброво ную» помощь в сканировании. То есть хорошей практикой будет проведение н стройки маршрутизатора с запретом формирования сообщения unreachable д того, чтобы ограничить атакующего в получении лишней информации.

Все сообщения протокола ICMP, относящиеся к информирующим о не стижимости получателя, имеют в своем заголовке поле Type равным 3. При проблем с доставкой может быть несколько, и для их более точного указан служит поле Code, принимающее разные значения. Таблица 4.4 содержит пер чень кодов, присваиваемых сообщениям ICMP с типом 3.

Таблица 4.4. Сообщения ICMP destination unreachable

Код	Значение	Условие отправки сообщения
0	Network Unreachable (сеть недостижима)	Генерируется маршрутизатором, если он не имеет информации о маршруте в целевую сеть
1	Host Unreachable (хост недостижим)	Генерируется маршрутизатором, если хост, подключенный в одну сеть с маршрутизатором, не достижим (то есть не отвечает на запросы ARP)
2	Protocol Unreachable (протокол недостижим)	Генерируется, если транспортный протокол, инкапсулированный в дейтаграмму, не поддерживается получателем
3	Port Unreachable (порт недостижим)	Генерируется, если транспортный протокол не может обработать пакет (например UDP) с указанным портом

Код	Значение	Условие отправки сообщения
4	Fragmentation needed and DF flag Set (требуется фрагментация, но установлен флаг DF — don't fragment — не фрагментировать)	Генерируется, если маршрутизатору нужно выполнить фрагментацию, но был установлен флаг DF
5	Source Route Failed (ошибка маршрутизации от источника)	Генерируется маршрутизатором, который не сумел отправить дейтаграмму далее при маршрутизации от источника
6	Destination Network Unknown (целевая сеть неизвестна)	Документ RFC 1812 определяет, что сообщение с таким кодом не следует формировать, а следует использовать сообщение с кодом 0 (сеть недостижима)
7	Destination Host Unknown (целевой хост неизвестен)	Формируется, если маршрутизатор смог определить, что хост — получатель дейтаграммы не существует
8	Source Host Isolated (отправитель изолирован)	Маршрутизатор был настроен так, чтобы запретить маршрутизацию от определенного отправителя
9	Communication with Destination Network is Administratively Prohibited (взаимодействие с целевой сетью запрещено в административном порядке)	Маршрутизатор настроен на блокирование трафика до определенной сети (подсети)
10	Communication with Destination Host is Administratively Prohibited (взаимодействие с получателем запрещено в административном порядке)	Маршрутизатор настроен на блокирование трафика до определенного хоста
11	Network Unreachable for Type of Service (сеть недостижима для заданного типа сервиса)	Маршрутизатор генерирует это сообщение, если маршрут в целевую сеть с запрошенным типом сервиса недоступен
12	Host Unreachable for Type of Service (получатель недостижим для заданного типа сервиса)	Маршрутизатор не может обработать дейтаграмму, так как ее маршрут до получателя не соответствует требованиям по типу сервиса для этого маршрута
13	Communication Administratively Prohibited (взаимодействие запрещено в административном порядке)	Маршрутизатор не может передать дейтаграмму далее, так как на нем настроены соответствующие фильтры
14	Host Precedence Violation (запрошенный приоритет обработки недопустим)	Посылается первым маршрутизатором в пути, сообщающим, что запрошенный уровень приоритета недопустим
15	Precedence cutoff in effect (заниженный уровень приоритета)	Был установлен минимально допустимый уровень приоритета, а дейтаграмма послана с приоритетом ниже этого уровня

Рассмотрим некоторые из сообщений destination unreachable с разными кодами, указывающими на разные причины сбоя при доставке дейтаграммы. Начнем с кода 13 — «взаимодействие запрещено». Отмечу, что иные маршрутизаторы разных производителей могут отключить формирование сообщений с кодами 13, 14 и 15. В таком случае маршрутизатор просто игнорирует (отбрасывает) дейтаграммы, обработка которых запрещена.

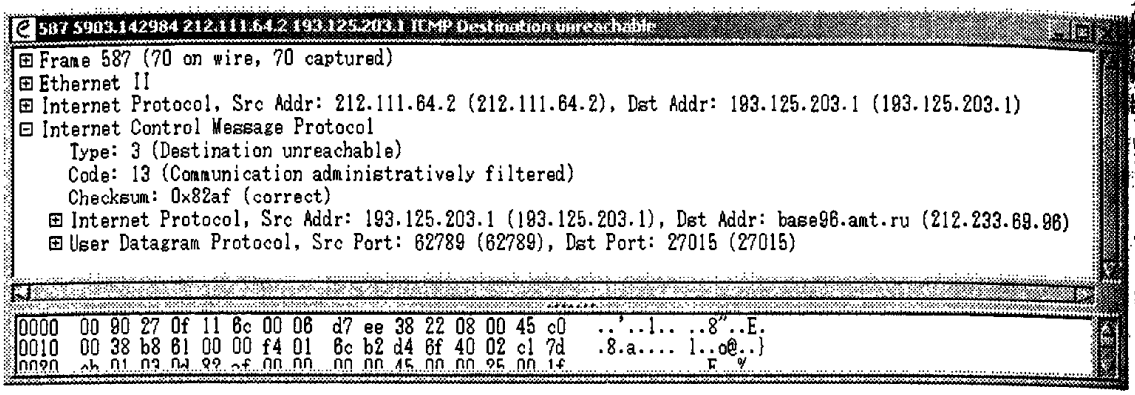


Рис. 4.15. Пример сообщения ICMP destination unreachable

В примере на рис. 4.15 через защитный экран посылалась дейтаграмма UDP с номером порта получателя 27015 (сетевая игра Counter-Strike) по адресу 212.233.69.96. Генератором исходной дейтаграммы был какой-то внутренний хост в сети, адрес которого уже трудно восстановить. Но так как дейтаграмма прошла через защитный экран, то последний отправил ее от себя (технология NAT) и выполнил трансляцию портов (технология PAT). Это легко проверить, если посмотреть на заголовок протокола UDP — номера портов отправителя с номерами большими, чем 60 000, чаще всего указывают на работу технологии PAT. По мере продвижения дейтаграммы к получателю промежуточный хост с адресом 212.111.64.2 сформировал сообщение ICMP с указанием того, что желаемый получатель недоступен в результате применения административного фильтра и вернул в этом сообщении заголовки оригинальной дейтаграммы. Можно более наглядно посмотреть путь дейтаграммы к адресу 212.233.69.96, если выполнить команду tracert с указанием в качестве параметра этого адреса.

```
C:\>tracert 212.233.69.96
...
  7   140 ms   141 ms   130 ms  MSK-M9-10.Relcom.EU.net [193.124.22.75]
  8   140 ms   180 ms   141 ms  MTU1-m9-gw.mtu.ru [193.232.246.30]
  9   140 ms   140 ms   131 ms  M9-Fex.core.mtu.ru [195.34.53.33]
 10   140 ms   140 ms   131 ms  msk-iki-cs1-fa0-0.agtel.net [193.232.244.75]
 11   140 ms   131 ms   130 ms  msk-amt-cs1-a3-0-2.agtel.net [212.111.67.14]
 12  amtcs2-e0-0.amt.ru [212.111.64.2]  сообщает: Заданная сеть недоступна.
```

Видно, что последним на пути следования дейтаграммы было устройство (вероятнее всего это маршрутизатор) с адресом 212.111.64.2.

Рассмотрим еще один пример (рис. 4.16), связанный с работой почтовой программы. В данном случае была сделана попытка отправить почту (установить соединение TCP с портом 25 на стороне получателя) серверу с адресом 195.242.2.249. В ответ промежуточный маршрутизатор отправил сообщение, уведомляя, что взаимодействие с указанным сервером запрещено администратором (скорее всего настроен фильтр на маршрутизаторе, блокирующий пакеты на определенные порты).

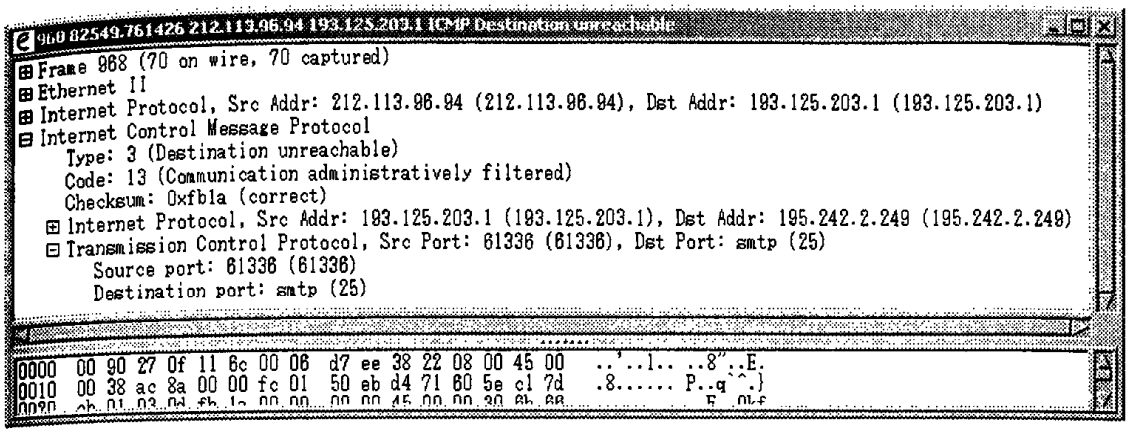


Рис. 4.16. Второй пример сообщения ICMP destination unreachable

Завершим рассмотрение сообщения ICMP destination unreachable последним примером (рис. 4.17).

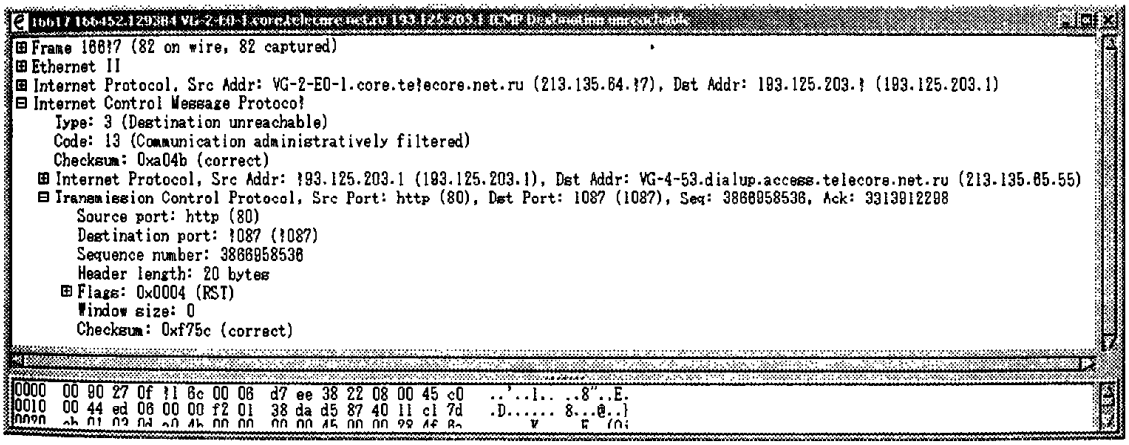


Рис. 4.17. Третий пример сообщения ICMP destination unreachable

Здесь обратите внимание на то, что в возвращенном сообщении ICMP включены заголовки IP и TCP и для последнего в оригинальной дейтаграмме номер порта отправителя был установлен в значение 80, а номер порта получателя — 1087. Учитывая то, что номер 80 зарезервирован для протокола HTTP, можно сделать вывод, что сегмент TCP с такими номерами портов был послан внутренним WWW-сервером получателю с адресом 213.135.65.55, находящемуся за защитным экраном. Кроме того, в сегменте TCP был установлен флаг RST. Это указывает на то, что сервер разрывает логическое соединение с клиентом. С большой долей вероятности можно предположить, что пользователь, работая через модем с модемным пулом провайдера (если принимать во внимание адреса DNS, то провайдера telecore — полагаю, что имя домена совпадает с именем компании), просматривал содержимое сайта внутри сети, а потом резко отключался от модемного пула. Как правило, серверы доступа, обслуживающие удаленных пользователей, настроены на автоматическое выделение адресов своим клиентам и после рассоединения адрес стал свободен. Так как клиент быстро отклю-

чился, его программа-обозреватель не успела корректно завершить соедине с сервером WWW, и последний после истечения тайм-аута отправил сегм TCP с флагом RST для принудительного завершения сеанса. Однако адрес, которому отправилась дейтаграмма IP с сегментом TCP, был уже недоступ и поэтому сервер доступа сформировал рассматриваемое сообщение ICMP.

Теперь перейдем к другому сообщению, с кодом 1 (host unreachable — «хо недостижим»), пример которого приведен на рис. 4.18.

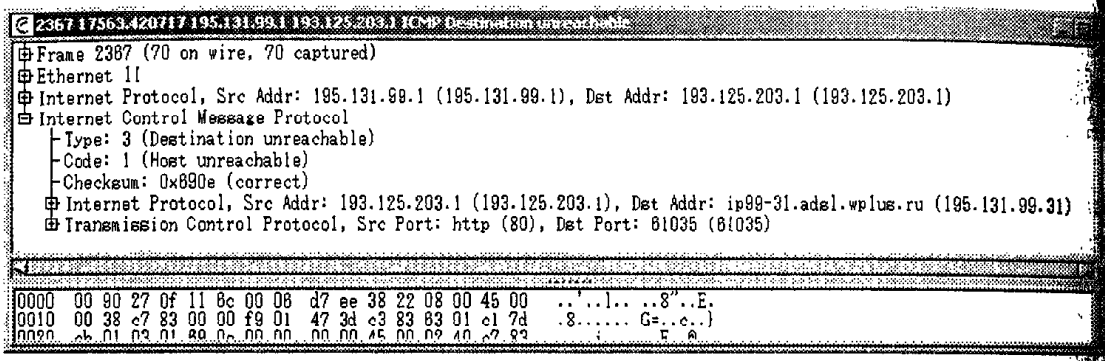


Рис. 4.18. Пример сообщения ICMP host unreachable

Источником сообщения является маршрутизатор, извещающий отправите ля о том, что по каким-то причинам получатель недостижим (маршрутизато проверяет это, анализируя поступление ответа на запрос ARP). Например, п указанному IP-адресу не найден активный хост, или же последний времен не активен, или, например, хост настроен так, чтобы не отвечать на опред ленный трафик (используется персональный защитный экран). Как уже отмеч лось, в такой ситуации получатель сам не может послать сообщение об ошиб и маршрутизатор делает это за него. Не трудно понять, что такая информац о сети очень ценна. Она помогает представить сеть в целом. Кроме того, ес для сканирования используется специальное программное обеспечение, то по лучение подобного сообщения позволит в будущем не обращать внимание н неактивные хосты, сконцентрировав его на активных. Это способствует сокра щению времени сбора информации. Если в распределенной сети используются маршрутизаторы Cisco Systems, то при настройке на них списков доступа мож но указать ключевое слово no ip unreachable, заставляя маршрутизатор отка заться от генерирования сообщения ICMP host unreachable. Кроме того, имее смысл рекомендовать на граничном маршрутизаторе настроить фильтр, кото рый будет блокировать исходящие сообщения host unreachable из внутренней сети в Интернет.

По умолчанию маршрутизаторы Cisco Systems настроены так, что отправ ляют различные сообщения ICMP unreachable. Например, когда маршрутиза тор получает на свой интерфейс нешироковещательный пакет, адресованный самому маршрутизатору с использованием неподдерживаемого протокола, в от вет посылается ICMP protocol unreachable. При получении дейтаграммы, марш рут которой до получателя не известен, отправителю также посылается ICM host unreachable.

Администратор может ограничить интенсивность отправки сообщений ICMP destination unreachable с помощью команды ip ICMP rate-limit unreachable (влияет на все интерфейсы и выполняется в глобальном режиме настройки). В качестве параметра команды указывается временной интервал в миллисекундах, в течение которого допускается отправка только одного сообщения ICMP destination unreachable. По умолчанию это значение составляет 500 мс. Другим параметром может быть ключевое слово DF, позволяющее ограничить количество посылаемых сообщений ICMP, эти сообщения будут иметь код 4 («требуется фрагментация, но установлен флаг DF»). Маршрутизатор поддерживает два таймера — один для обычных сообщений ICMP destination unreachable, а второй — для тех же сообщений, но с кодом 4.

В приведенном примере основанием для отправки этого сообщения ICMP был пакет протокола TCP с адресом назначения 195.131.99.31 и портом 80 на стороне получателя (протокол HTTP). Здесь очень велика вероятность того, что некий хост работал (с использованием технологии ADSL) через защитный экран с внутренним сервером WWW и в какой-то момент хост стал неактивен, но сервер, не зная об этом, продолжал посылать пакеты TCP. Промежуточный маршрутизатор с адресом 195.131.99.1, обнаружив факт неактивности хоста, сформировал в ответ на пакеты сообщение ICMP.

Рассмотрим другой пример перехвата сообщения ICMP (рис. 4.19), в котором получатель информирует отправителя о том, что запрошенный порт протокола UDP недостижим (сообщение Port Unreachable). В этом примере отправитель пытался послать информацию по протоколу UDP на порт с номером 53.

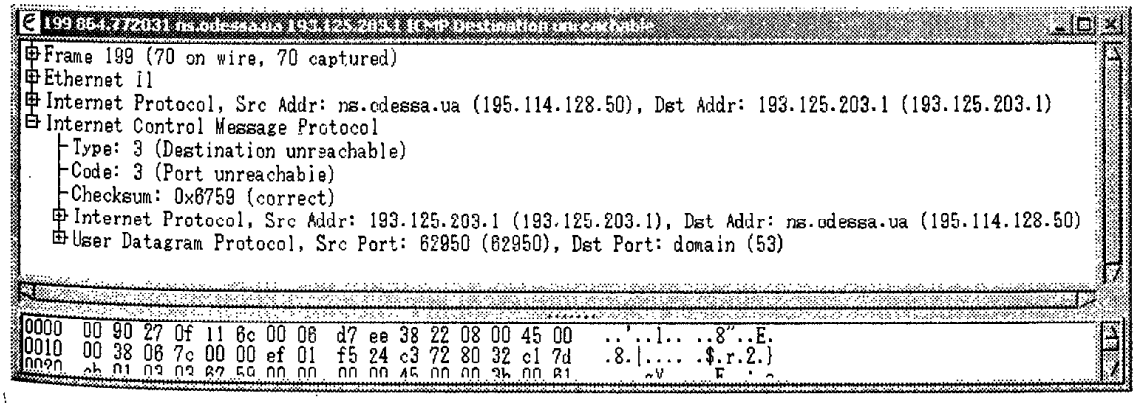


Рис. 4.19. Первый пример сообщения port unreachable

В приведенном перехвате был сделан запрос на разрешение доменного имени с помощью протокола UDP (порт получателя 53). Целевой хост в сети Интернет в ответ отправил сообщение ICMP port unreachable, указывающее на то, что запросы к требуемому порту им не могут быть обработаны. Можно предположить, что предпринималась попытка разрешить адрес и был послан запрос серверу, который по какой-то причине не смог его обработать, хотя его имя косвенно указывает на то, что это сервер разрешения имен (обычно, хотя и не всегда, в сети Интернет имя хоста ns указывает на то, что это name server — сервер

имен). Точнее говоря, причиной отправки является то, что порт с номером закрыт, но при этом сам хост работает. Следуя подобной логике, можно выбра порт протокола UDP, который заведомо будет закрытым (closed), и послать него пакет. Подобный метод годится для определения злоумышленником н строенных списков управления доступом на граничном маршрутизаторе — есл в ответ на наш запрос ничего не поступило, то это может указывать на то, ч маршрутизатор с настроенными списками доступа не пропустил запрос, а есл поступает сообщение ICMP port unreahcable, то интересующий хост работа и граничный маршрутизатор не настроен на блокирование входящих пакет с выбранным номером порта, а также на исходящие сообщений ICMP.

Рассмотрим пример пакета (рис. 4.20), который в состоянии повлечь форми рование сообщения ICMP port unreachable. Этот пакет был перехвачен пр начальной загрузке одного из компьютеров в сети, работающих под управлен ем Microsoft Windows 2000 Professional. Если обратиться к справочнику порто UDP, то можно найти информацию о том, что порт 1900 используется при ра боте протокола SSDP (Simple Service Discovery Protocol, протокол упрощенно службы обнаружения).

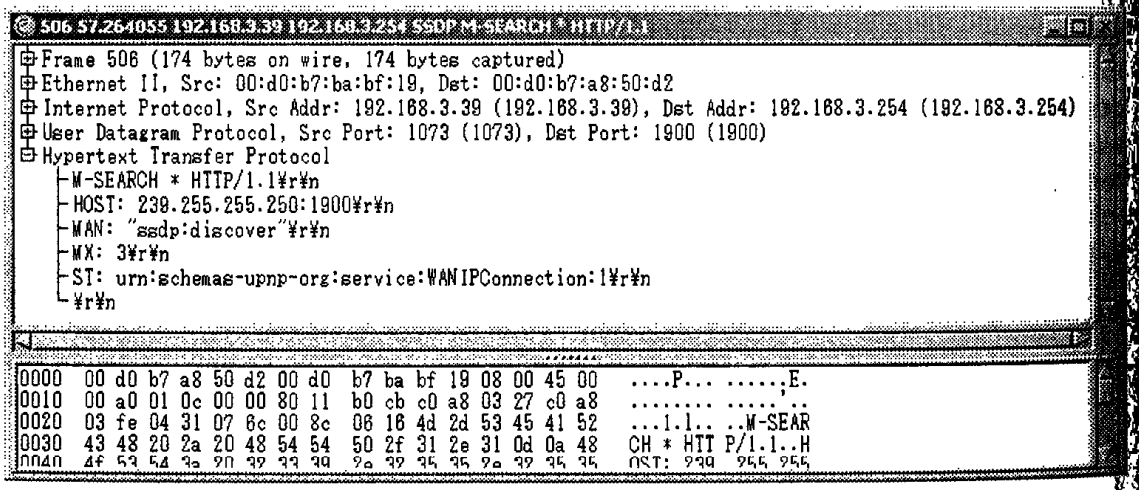


Рис. 4.20. Пакет протокола SSDP

Этот пакет отправляется хосту с адресом 192.168.3.254, который не под держивает упомянутый протокол и, соответственно, порт с номером 1900 яв ляется закрытым. Сервер также формирует сообщение ICMP port unreachable (рис. 4.21).

Следует отметить, что протокол TCP имеет возможность известить отправите ля о неактивности определенного порта посредством возврата сегмента с установ ленными флагами RST и ACK. А протокол UDP не приспособлен к извещению об ошибках, поэтому он полагается на сообщения ICMP. Этой его несамостоя тельностью пользуются при сборе информации о сети. Так, посылая дейтаграм мы протокола UDP на различные порты и анализируя ответные сообщения ICMP, можно легко сформировать список активных портов на объекте иссле дования.

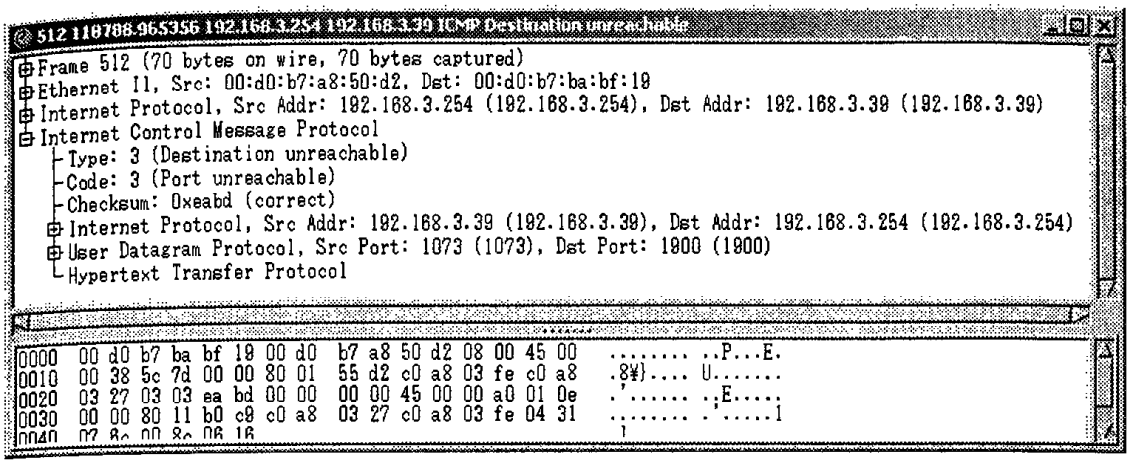


Рис. 4.21. Второй пример сообщения ICMP port unreachable

Например, из сети Интернет можно послать пакеты на различные адреса, принадлежащие интересующей сети. Когда граничный маршрутизатор получит эти пакеты, он проверит адреса получателей и будет принимать решение о дальнейшей обработке. В том случае, когда он не знает, как направить пакеты далее, маршрутизатор сформирует либо сообщение ICMP host unreachable (если получатель не ответил на запрос ARP), либо сообщение ICMP time exceeded. Последнее отправляется по той причине, что обработка пакета выполнялась очень долго и время жизни дейтаграммы истекло. Это может быть вызвано ожиданием ответа на запрос ARP для определения активности хоста, которому адресован пакет. Если обратно не поступило никаких сообщений, то логично предположить, что хост с определенным адресом внутри сети присутствует. Подобное сканирование может выявляться средствами IDS, но ему есть альтернатива — так называемое decoy-сканирование (подстановка адресов). Здесь «военная хитрость» заключается в том, что пакеты отсылаются с различными адресами отправителей, среди которых есть и адрес хоста, с которого осуществляется сканирование. Маршрутизатору, который не знает, какому получателю следует доставить пакет, все равно, кому адресовать сообщение ICMP host unreachable — он отправит его по всем «липовым» адресам (кроме лишнего исходящего трафика это не привнесет никаких проблем), но также и по адресу сканирующего хоста. Такие действия способны значительно затруднить однозначную идентификацию хоста, с которого происходит сбор информации о вашей сети. Поэтому можно порекомендовать просто запретить формирование сообщений ICMP host unreachable и time exceeded на граничном маршрутизаторе (или защитном экране).

Сообщение ICMP redirect

Рассмотрим другое сообщение протокола ICMP, которое также может оказаться лазейкой для некорректной деятельности в сети. Сообщение redirect (перенаправление) позволяет маршрутизатору известить отправителя информации о том, что он выбрал не оптимальный маршрут для отправки информации к получателю.

Маршрутизатор в любом случае будет передавать трафик получателю, однако он, генерируя сообщение redirect, предлагает отправителю изменить свою таблицу маршрутизации в пользу более предпочтительного маршрута. Важно, что маршрутизатор должен располагаться в той же подсети, что и отправитель информации.

В том случае, когда получателем сообщения redirect является маршрутизатор, он вправе игнорировать это сообщение при условии включенной поддержки протокола маршрутизации. Кроме того, сам хост не должен посылать сообщение redirect — его инициатором может быть только маршрутизатор. Интересно отметить, что хост после получения сообщения о перенаправлении обязан изменить свою таблицу маршрутизации, однако само сообщение может быть проигнорировано, если предложенный новый шлюз не располагается в той же подсети-источнике сообщения и если отправитель этого сообщения на данный момент не является первым транзитным узлом в пути к подразумеваемому получателю.

Существуют несколько кодов для типа сообщения, равного 5 (redirect). Они перечислены в табл. 4.5.

Таблица 4.5. Коды для сообщения ICMP redirect

Код	Описание
0	Redirect Datagram for the Network — перенаправление дейтаграммы для сети
1	Redirect Datagram for the Host — перенаправление дейтаграммы для хоста
2	Redirect Datagram for the Type of Service and Network — перенаправление дейтаграммы для указанного типа сервиса и сети
3	Datagram for the Type of Service and Host — перенаправление дейтаграммы для указанного типа сервиса и хоста

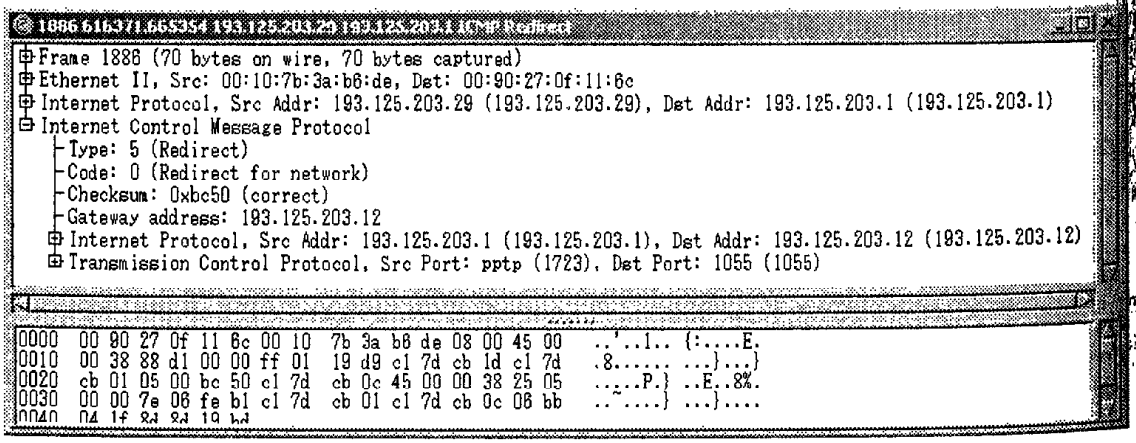


Рис. 4.22. Сообщение ICMP redirect (код 0)

На рис. 4.22 показан пример перехваченного сообщения ICMP redirect с кодом 0, что указывает на то, что перенаправление предлагается для сети. Как и другие сообщения об ошибках протокола ICMP, рассматриваемое включает в себя заголовок дейтаграммы, вызвавшей отправку сообщения redirect. В данном случае отправку сообщения вызвал пакет протокола TCP на порт 1723 — этот пакет относится к управляющему соединению PPTP.

В распределенной сети, состоящей из нескольких связанных между собой через маршрутизаторы объектов, сообщения ICMP redirect появляются довольно часто. Дело в том, что когда загружается компьютер, работающий под какой-либо операционной системой семейства Microsoft Windows, таблица маршрутизации обычно содержит несколько записей. Например, адрес шлюза по умолчанию, допустим, это маршрутизатор. Тогда, если в сеть отправляется дейтаграмма с компьютера, информация о котором не заложена в таблицу маршрутизации, то она отправляется шлюзу по умолчанию. Однако в распределенной сети вовсе не обязательно, что оптимальный маршрут до получателя проходит через шлюз по умолчанию. Когда шлюз по умолчанию получает дейтаграмму, адресованную получателю, до которого есть другой, более короткий маршрут от отправителя, он извещает последнего об этом маршруте с помощью сообщения ICMP redirect (но исходную дейтаграмму он перешлет сам). При получении сообщения ICMP хост выполняет проверку, чтобы убедиться, что оно получено от шлюза по умолчанию (а точнее, от первого в пути до получателя маршрутизатора — но при изначальной таблице маршрутизации это как раз и будет шлюз по умолчанию) и что отправитель подключен к той же сети, что и сам хост. При выполнении обоих условий новый маршрут вносится в таблицу маршрутизации. Администратор может управлять подобным поведением хоста с помощью параметра Tcpip\Parameters\ EnableICMPRedirects в реестре Windows (путь HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\ Parameter), установка которого в True (или 1; значение по умолчанию) позволит изменять таблицу маршрутизации, получая сообщение ICMP redirect.

Рассмотрим сказанное на примере. На рис. 4.23 показан фрагмент реальной сети, в которой два объекта связаны между собой через Интернет на базе технологии виртуальных частных сетей (VPN — Virtual Private Network). Связь осуществляется с помощью протокола PPTP и поддерживается двумя серверами — по одному на объект, — названных на рисунке серверами VPN (построены на базе Microsoft Windows 2000 Server). Далее в книге технология виртуальных частных сетей (а также протокол PPTP) освещается более детально, а сейчас можно считать, что эти два сервера просто маршрутизаторы, которые знают пути друг к другу.

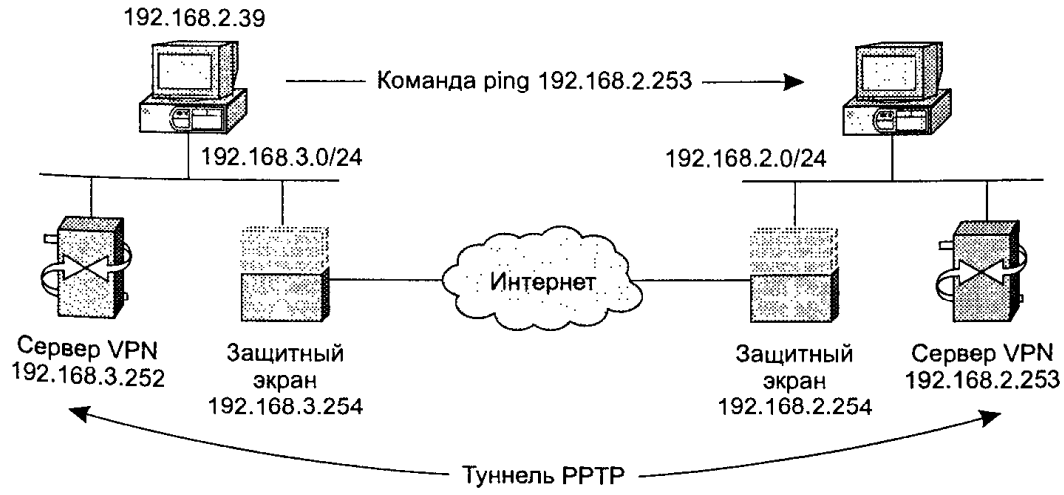


Рис. 4.23. Пример перенаправления на реальной сети

Предположим, что хост с адресом 192.168.3.39 выполняет команду ping 1
168.2.253 (рис. 4.23). Перед тем как отправить по сети дейтаграмму, содер
щую запрос ICMP echo, хост проверяет свою таблицу маршрутизации и пытае
выяснить, какой маршрут следует выбрать (сам хост располагается в сети кл
са C 192.168.3.0/24, а получатель запроса — в другой сети 192.168.2.0/24). Д
этого хост перебирает свою таблицу маршрутизации, подыскивая маршрут в
левую сеть 192.168.2.0/24. Изначально таблица имеет следующий вид.

```
C:\>route print
...
Сетевой адрес          Маска сети      Адрес шлюза     Интерфейс  Метрика
          0.0.0.0          0.0.0.0    192.168.3.254   192.168.3.39       1
        127.0.0.0        255.0.0.0        127.0.0.1     127.0.0.1       1
      192.168.3.0    255.255.255.0    192.168.3.39    192.168.3.39       1
     192.168.3.39  255.255.255.255        127.0.0.1     127.0.0.1       1
    192.168.3.255  255.255.255.255    192.168.3.39    192.168.3.39       1
        224.0.0.0        224.0.0.0    192.168.3.39    192.168.3.39       1
  255.255.255.255  255.255.255.255    192.168.3.39    192.168.3.39       1
Основной шлюз:     192.168.3.254
```

Как видно, маршрута в сеть 192.168.2.0/24 или прямо до получателя с адр
сом 192.168.2.253 в таблице маршрутизации хоста нет, но есть запись шлюза п
умолчанию (192.168.3.254). Обнаружив это, хост отправляет запрос ICMP ech
по адресу 192.168.3.254 в надежде на то, что шлюз по умолчанию (а в рассматри
ваемой сети это адрес внутреннего интерфейса защитного экрана) знает маршру
в сеть 192.168.2.0/24. Обратите внимание на то, что сама дейтаграмма имеет адре
получателя 192.168.2.253, но на канальном уровне кадр отправляется по MAC-ад
ресу, который назначен сетевой карте защитного экрана (то есть той, у которо
IP-адрес 192.168.3.254).

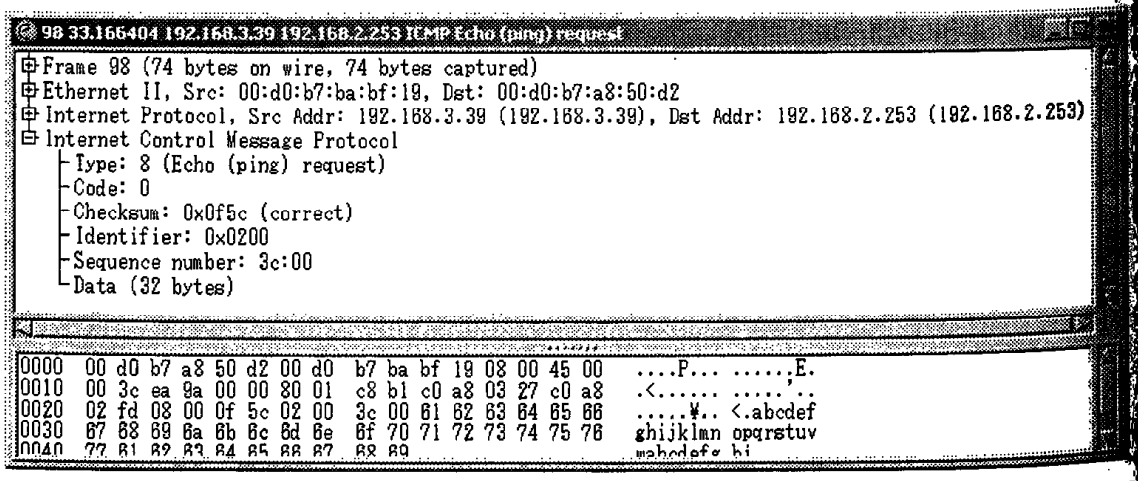


Рис. 4.24. Запрос ICMP echo request, направленный удаленному серверу

Сервер с адресом 192.168.3.254 получает дейтаграмму (рис. 4.24) и, в свою
очередь, проверяет собственную таблицу маршрутизации, пытаясь найти маршру
в сеть 192.168.2.0/24, где располагается получатель. У него такая запись при

сутствует (механизм формирования таблиц маршрутизации будет рассмотрен в главе 7) и она указывает на то, что дейтаграмму необходимо доставить хосту с адресом 192.168.2.252 (сервер VPN на рисунке). Дейтаграмма передается далее по данному адресу, и одновременно сервер-защитный экран посылает отправителю дейтаграммы сообщение ICMP redirect, указывающее на то, что есть более предпочтительный маршрут до получателя, который позволяет исключить из маршрута один маршрутизатор. Рисунок 4.25 показывает это сообщение ICMP redirect, которое содержит рекомендуемый адрес нового маршрутизатора и заголовок исходной дейтаграммы (анализатор считает, что контрольная сумма неверная, так как не все поля исходного сообщения ICMP echo, по которым она вычисляется, присутствуют в полученных данных).

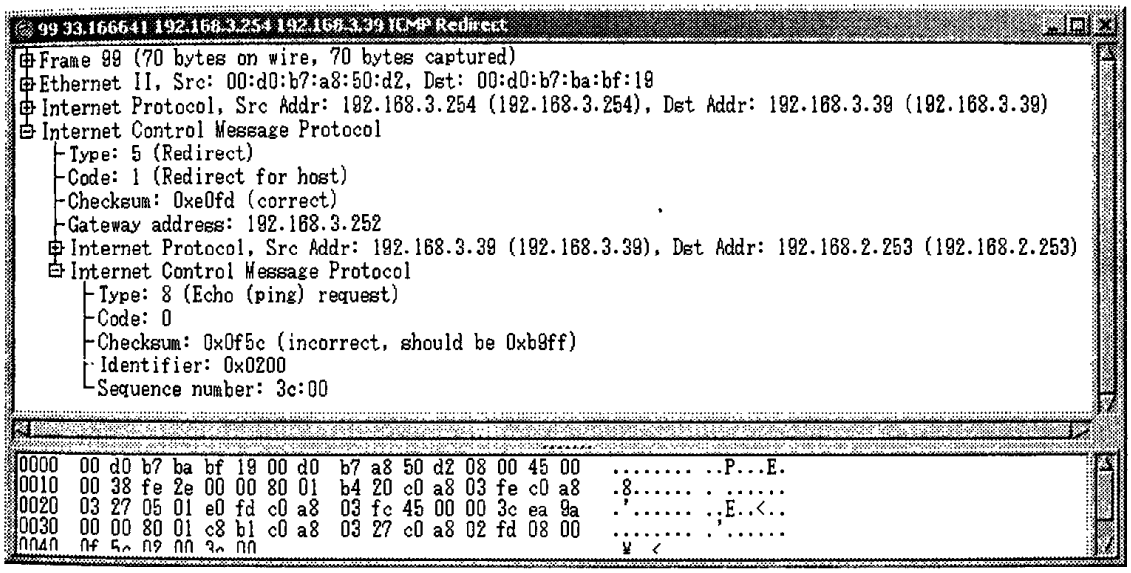


Рис. 4.25. Пример сообщения ICMP redirect (код 1)

Хост, с которого выполнялась команда ping, получает это сообщение и добавляет новую запись в свою таблицу маршрутизации, указывающую на то, что теперь до получателя 192.168.2.253 есть свой особый маршрут.

```
C:\>route print
...
Сетевой адрес          Маска сети      Адрес шлюза     Интерфейс  Метрика
          0.0.0.0          0.0.0.0    192.168.3.254   192.168.3.39       1
        127.0.0.0        255.0.0.0        127.0.0.1     127.0.0.1       1
    192.168.2.253  255.255.255.255    192.168.3.252   192.168.3.39       1
      192.168.3.0    255.255.255.0    192.168.3.39    192.168.3.39       1
    192.168.3.39   255.255.255.255        127.0.0.1     127.0.0.1       1
    192.168.3.255  255.255.255.255    192.168.3.39    192.168.3.39       1
    192.168.5.253  255.255.255.255    192.168.3.252   192.168.3.39       1
        224.0.0.0        224.0.0.0    192.168.3.39    192.168.3.39       1
  255.255.255.255  255.255.255.255    192.168.3.39    192.168.3.39       1
Основной шлюз:       192.168.3.254
```

Обратите внимание, что в данном случае используется сообщение ICMP redirect for host (перенаправление для хоста) и новая таблица маршрутизации содержит запись для хоста с адресом 192.168.2.253. Если теперь с того же хоста послать запрос ICMP echo по другому адресу — 192.168.3.254, то описанная процедура повторится вновь. Причем абсолютно неважно, активен ли получатель запроса или нет — сообщение ICMP redirect будет отправляться в любом случае.

Администратор, настраивая маршрутизатор Cisco Systems, может воспользоваться командой ip redirects, которая разрешает маршрутизатору отправлять сообщения ICMP redirect тогда, когда маршрутизатор вынужден переслать входящий пакет через интерфейс, на который он поступил. По умолчанию отправка сообщений ICMP redirect разрешена.

Информационные запросы протокола ICMP

Рассмотренные выше сообщения ICMP echo request не единственный доступный тип запроса. Так называемые non-echo-запросы ICMP довольно широко используются при реализации механизмов сканирования. Помимо наиболее «популярных» echo request и echo reply (обычная команда ping), уже обсужденных нами, можно привести следующие типы запросов: timestamp request и reply (запрос времени), information request и reply (запрос информации), address mask request и reply (запрос маски), router solicitation и router advertisement (анонсирование маршрутизатора). Остановимся на каждом из них подробнее.

Сообщение ICMP timestamp request (поле Type: 13 в запросе, Type: 14 в ответе на запрос, рис. 4.26) позволяет опросить хост в сети на предмет определения текущего времени, что также помогает отправителю определить степень задержки, вносимой сетью. При формировании запроса отправитель устанавливает временную отметку (originate timestamp) и отправляет запрос. Получатель сразу же при получении запроса заполняет временную отметку о получении (receive timestamp) и передачи (transmit timestamp), изменяет тип сообщения с 13 на 14 и посылает его в обратный путь. Временная отметка представляет собой число миллисекунд, отсчитываемых с полуночи по Гринвичу. Отметка о завершении инициализации сообщения (originate timestamp) определяет время, соответствующее моменту последнего изменения сообщения отправителем перед его посылкой. Отметка о получении (receive timestamp) указывает на момент времени, когда получатель начал работать с сообщением, а отметка передачи (transmit timestamp), соответственно, определяет последний момент перед отправкой в сеть получателем. Говоря о применении этого сообщения с некорректными целями, можно обнаружить в сети устройства, которые отвечают на запрос, хотя с точки зрения нанесения явного вреда работающему хосту подобная информация вряд ли будет полезной. В основном данный механизм может использоваться для временной синхронизации на взаимодействующих хостах и измерения задержки передачи информации в одном направлении.

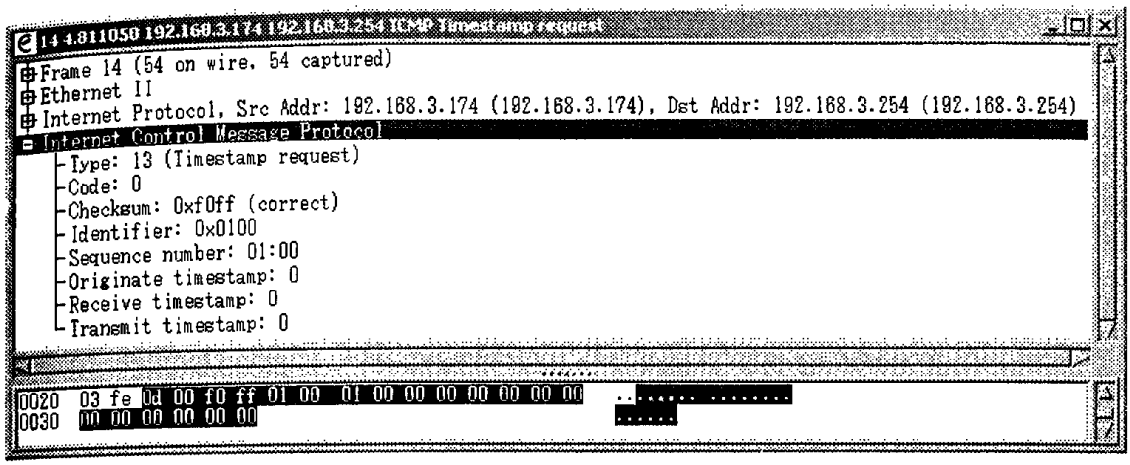


Рис. 4.26. Пример сообщения ICMP timestamp request

Для генерирования запроса, показанного на рис. 4.26, использовалась система NAI CyberCop Scanner, которая после получения приведенного на рис. 4.27 ответа сделала вывод, что хост 192.168.3.254 уязвим, так как отвечает на эти запросы и может предоставить информацию о своем системном времени. Следует отметить, что большинство операционных систем устанавливают receive timestamp и transmit timestamp в одинаковые значения.

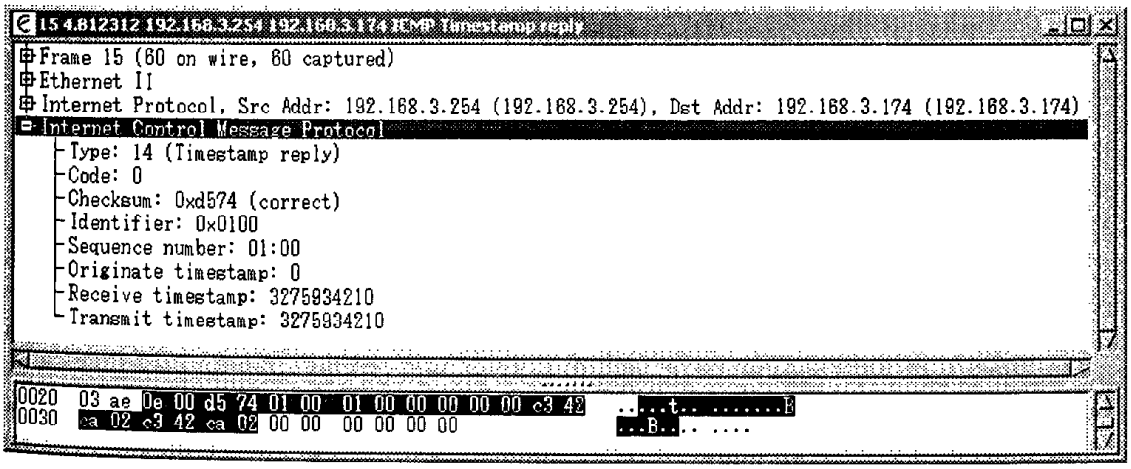


Рис. 4.27. Пример сообщения ICMP timestamp reply

В документе RFC 1122 оговорено, что хост может реализовать поддержку запроса ICMP timestamp request и ответа на него. Если поддержка реализована, то выдвигается ряд требований (которые уже должны соблюдаться). Интересна возможность игнорирования запросов с широковещательным и групповым адресом получения (все операционные системы Microsoft Windows не отвечают на запросы с широковещательным адресом, в то время как операционная система Linux 2.2.x отвечать будет). Далее, компьютер, работающий под управлением Microsoft Windows NT + Service Pack 6 не станет откликаться на запрос timestamp request (это позволяется документом RFC 1122), в то время как Windows 2000

Server/Professional будет отвечать, что, собственно, подтверждается вышеприведенным перехватом, так как хост 192.168.3.254 работает под управлением Windows 2000 Server. Обратите внимание, что речь идет о нешироковещательных запросах. Так, ничто не мешает атакующему выполнить ping sweep с использованием non-echo-запросов (например, timestamp request) и указанием широковещательного адреса в качестве адреса получателя. Впрочем ни одна операционная система семейства Windows отвечать на запрос не будет. Размер рассматриваемого сообщения укладывается между 40 и 60 байтами, из которых 20–40 байтов относятся к заголовку дейтаграммы IP, 4 байта занимает заголовок сообщения ICMP и 16 байтов отводятся полю самого сообщения.

Следующий запрос ICMP information request (запрос информации) и ответ на него направлены на поддержку систем, требующих выполнения процедуры самоконфигурирования. Это могут быть, например, бездисковые станции, использующие запрос для определения своих сетевых адресов. При формировании запроса поле получателя в заголовке дейтаграммы IP устанавливается в ноль (означая «эта сеть»), кроме того, адрес отправителя также может иметь нулевое значение. Затем отправитель устанавливает поля идентификатора и номера последовательности для сопоставления запросов с поступающими ответами, а поле Code сбрасывается в ноль. Содержимое ответа зависит от запроса — если адреса получателя и отправителя нулевые, то в ответной дейтаграмме эти поля будут заполнены сетевым адресом. Следует отметить, что в сетях Microsoft Network существует более удобный метод назначения адресов. Этот метод использует протокол DHCP (Dynamic Host Configuration Protocol, протокол динамической настройки хоста). В настоящее время данный тип сообщения ICMP устарел и в документе RFC 1812 сказано, что маршрутизатору и хосту не следует генерировать ответ на сообщение information request. Тем не менее, несмотря на рекомендацию RFC 1812, хосты в сети могут отвечать на такие запросы. А администраторам рекомендовано блокировать прохождение запросов и ответов на них через защитный экран, также как и сообщений ICMP timestamp. Кроме того, если в сети обнаружены подобные пакеты, следует обратить внимание на их источник — с него может выполняться сканирование сети с целью определения типов операционных систем, работающих на хостах. Например, злоумышленник способен установить адрес получателя в некое определенное значение, а не в ноль. В таком случае некоторые компьютеры (операционные системы) будут отвечать на запрос предоставления информации ICMP даже если он поступил не из той же сети, где они находятся. Причем несмотря на то, что отправка запроса с нулевым адресом получателя является нарушением требований RFC, формирование ответа таковым не является, так как в документе не содержатся точные инструкции, как поступать в подобной ситуации. Рассматриваемый запрос может быть также послан на широковещательный адрес — на него операционные системы Microsoft Windows отвечать не будут.

Аналогично работает запрос address mask request, который тоже используется бездисковыми станциями для получения адресной информации, точнее, для установления маски подсети (после того как хосту присвоен конкретный IP-адрес). Эта информация может помочь злоумышленнику заполучить в свое распоряжение

например, схему маршрутизации. Размер сообщений address mask request/reply находится в пределах между 32 и 52 байтами (8 байтов отводятся под поле данных).

Если следовать документу RFC 1812, то маршрутизатор должен поддерживать сообщение address mask, что помогает идентифицировать маршрутизаторы в пути к целевой сети. Однако RFC 1122 указывает, что использование хостами этого типа сообщения (запрос и ответ) является исключительно добровольным. Таким образом, если маршрутизатор поддерживает рассматриваемое сообщение, то рекомендуется предусмотреть параметры конфигурации, которые определяют для каждого логического интерфейса, должен ли он отвечать на поступающие запросы ICMP address mask request. По умолчанию маршрутизатор не должен их игнорировать, но если он не владеет информацией о корректной адресной маске, то отвечать вовсе не обязан.

Учитывая тот факт, что хост не обязательно получит адресную маску в том случае, если другой маршрутизатор выключен (или не отвечает на запросы), маршрутизатор-отправитель может периодически посылать широковещательные сообщения ICMP address mask reply на каждый из своих логических интерфейсов, после того как на них были настроены маски адресов. Указанная схема работы имеет все шансы оказаться опасной в случае использования маски подсети переменной длины (VLSM — Variable Length Subnet Mask, глава 2). В результате если маршрутизатор рассылает информацию периодически, то он исключает из этого процесса логические интерфейсы, которые либо настроены на запрет отправки ответов, либо разделяют между собой один физический интерфейс и общие сетевые префиксы.

Хосты, которые при своей загрузке пытаются определить адресную маску с помощью данного типа сообщения ICMP, отправляют его по широковещательному адресу (причем в случае неполучения ответа повторяют это несколько раз). До тех пор пока на посылаемые запросы не поступил ответ, хосту предписано считать, что маска соответствует классу его IP-адресу, или иными словами, что сеть не поделена на подсети. Информация из первого поступившего ответа и будет использоваться для установки маски адреса, а остальные сообщения будут игнорироваться. Отмечу, что факт генерации ответа на запрос address mask request зависит о того, является ли хост авторизованным агентом (authoritative agent), которому позволительно формировать ответы. Агентом может быть как хост (компьютер, работающий под какой-либо операционной системой), так и маршрутизатор — но в любом случае, он при своей инициализации должен широковещательно рассылать сообщения address mask reply.

На защитном экране рекомендуется настроить список блокировки доступа для сообщений address mask request, поступающих из сети Интернет. Дело в том, что если на запрос поступает ответ, то можно получить информацию не только о том, что хост активен и доступен, но также и адресную маску той сети, где он располагается. Так, операционные системы Microsoft Windows 95/98 будут отвечать на запрос address mask (видимо, считая себя авторизованными агентами), в то время как Microsoft Windows 2000 не будет (причем все они обойдут вниманием запрос, посланный широковещательно). Например, посмотрите на перехваченную дейтаграмму (рис. 4.28), которая поступила из Интернета на внешний адрес защитного экрана (193.125.203.1).

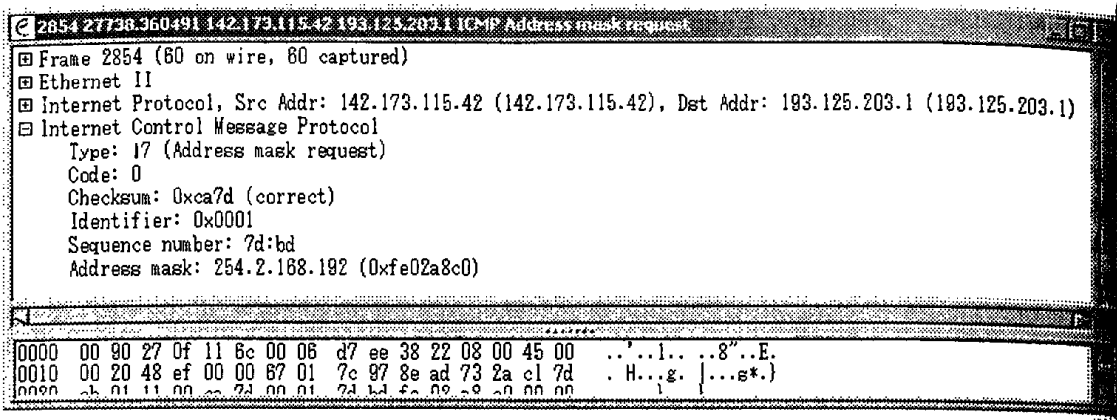


Рис. 4.28. Пример сообщения ICMP address mask request

Дейтаграмма содержит запрос address mask request для адреса 254.2.168.192. Эта дейтаграмма была перехвачена системой обнаружения атак snort, анализирующей трафик на внешнем интерфейсе защитного экрана. Обосновать причину отправки запроса сложно, но можно предположить, что запрос был сгенерирован случайно. При работе с маршрутизаторами Cisco Systems у администратора есть в распоряжении команда ip mask-reply, выполнение которой разрешит маршрутизатору отвечать на запросы ICMP mask request. По умолчанию такая возможность отключена.

Сообщение ICMP parameter problem

Остановимся на особенностях работы протокола в случае, когда в сети происходят события, вызванные, скорее всего, вмешательством посторонних. Расширенные методы обнаружения хостов в сети (сканирования) опираются на идею, которая сводится к тому, что атакующий может заставить хосты разными способами отправлять сообщения об ошибках протокола ICMP. Например, одним из методов является посылка дейтаграммы с некорректными полями в заголовке. В этом случае получатель, обнаружив проблемы с заголовком, пошлет отправителю дейтаграммы сообщение ICMP parameter problem.

Сообщение parameter problem посылается, если маршрутизатор, который обязан его сформировать, или конечный хост, который способен его сформировать, обрабатывая полученную дейтаграмму, испытывает сложности в интерпретации полей ее заголовка. Тогда дейтаграмма отбрасывается (игнорируется), а ее отправителю посылается указанное сообщение, с полем Code, равным 0 или 2. В первом варианте в заголовке ICMP будет точно указано на байт в заголовке IP-дейтаграммы, который вызвал ошибку. Второй код используется, если в полученной дейтаграмме поля «Длина заголовка» или «Общая длина» некорректны. Следует сказать, что указанный метод сканирования активных хостов из сети Интернет является довольно мощным, так как хосты могут отвечать сообщением parameter problem, а маршрутизаторы должны на них откликаться. Тем не менее, не все маршрутизаторы проверяют правильность всех полей дейтаграм

мы, как это делают конечные устройства. В общем случае, согласно документу RFC 1812, маршрутизатор проверяет только поле контрольной суммы и отбрасывает дейтаграмму, если последняя некорректна. Интересна следующая особенность: если маршрутизатор удалил дейтаграмму и послал в ответ сообщение parameter problem, то анализируя его «внутренности» можно определить производителя этого маршрутизатора. Администратору сети следует внимательно относиться к сообщениям системы IDS, которая должна среагировать на наличие в сети дейтаграмм с некорректными полями в заголовке.

Также немаловажна возможность сканирования активных хостов с определением доступных портов из сети Интернет. Данная возможность достигается формированием дейтаграммы с некорректными полями в заголовке с последующей инкапсуляцией в нее сегмента TCP или дейтаграммы UDP с установкой желаемых номеров портов. Если ответ на такую дейтаграмму получен, то это может означать, что используемый протокол и указанный порт доступны на получателе, и что сообщение parameter problem может проходить из внутренней сети через защитный экран в сеть Интернет. Кроме того, это может означать, что защитный экран не проверяет поля дейтаграммы на корректность.

Помимо указания неверной длины заголовка, несуществующее значение в поле «Тип протокола» также приводит к ошибке. В итоге получатель, зафиксировав этот факт, пошлет в ответ сообщение ICMP destination unreachable или protocol unreachable. Соответственно, если дейтаграмма поступила из Интернета, то отсутствие ожидаемого сообщения ICMP может указывать на то, что, например, защитный экран, проверив заголовок, не пропустил дейтаграмму дальше. Администратору сети рекомендуется настроить защитный экран на блокирование сообщений ICMP protocol unreachable, идущих из защищаемой сети в Интернет.

Сообщение ICMP time exceeded

Сообщение ICMP time exceeded формируется маршрутизатором, если последний обнаружил, что поле TTL в заголовке оригинальной дейтаграммы IP, которую он обрабатывает, равно нулю. В документе RFC 791 замечено, что маршрутизатор должен уменьшать значение этого поля на время, затраченное на обработку, в секундах. Максимальное значения поля TTL равно 255 (то есть 255 секунд), что соответствует 4,5 минутам. Когда значение поле TTL достигает нуля, дейтаграмма отбрасывается маршрутизатором и генерируется вышеуказанное сообщение ICMP с кодом 0 (Time-to-Live Exceeded in Transit). Рисунок 4.29 показывает пример сообщения, поступившего в ответ на отправку пакета DNS (протокол UDP, порт 53). Как видно из рисунка, подобно другим сообщениям об ошибках, рассматриваемое также включает в себя заголовок исходной дейтаграммы.

Пара слов о выбранной единице измерения времени (секунды). В настоящее время маршрутизаторы обрабатывают дейтаграмму за гораздо меньший промежуток времени (кроме ситуации, когда маршрутизатор перегружен — в таком случае обработка может занять несколько секунд). Учитывая это, транзитные маршрутизаторы обычно уменьшают значения поля TTL на 1.

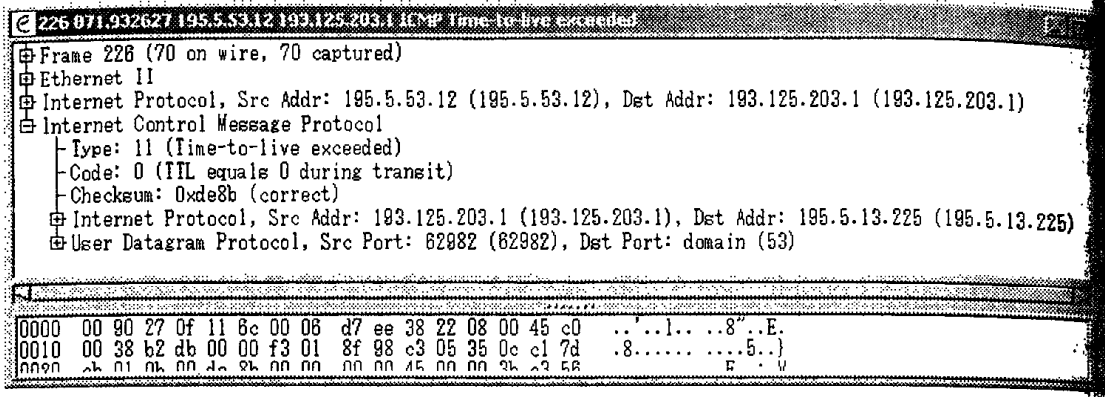


Рис. 4.29. Пример сообщения ICMP TTL exceeded (на переходе)

Наличие подобного сообщения протокола ICMP, как правило, указывает на возникновение петель маршрутизации, когда дейтаграмма плутает где-то в сети Интернет до тех пор, пока не будет удалена. Маршрутизатор может быть настроен на запрет отправки сообщений ICMP такого рода, но по умолчанию сообщения TTL exceeded должны отправляться. В то же время производитель маршрутизатора может предусмотреть возможность отключить отправку подобных сообщений для каждого интерфейса.

Обычное сообщение ICMP TTL exceeded имеет в поле Code значение 0, но если этот код равен 1, это свидетельство того, что время, отведенное на сборку фрагментированных дейтаграмм, истекло (на английском языке — Fragment Reassembly Time Exceeded). Детали фрагментирования будут рассмотрены немного ниже, а пока отмечу, что каждый фрагмент отправляется в своей дейтаграмме и маршрутизируется независимо от других фрагментов. Далее, все фрагменты имеют одинаковые значения в поле Identification (идентификация) в заголовках и уникальные значения в поле Offset (смещение), которое позволяет воссоздать исходную дейтаграмму на устройстве-получателе. Получатель, принимая все фрагменты (их может быть несколько в зависимости от размера исходной дейтаграммы и значения MTU в сети), должен собрать их в единое целое, а затем передать протоколу более верхнего уровня (TCP, UDP). В том случае, если дейтаграмму не удается собрать по причине пропущенных фрагментов, они отбрасываются и генерируется сообщение ICMP TTL exceeded с кодом 1. Такое поведение способствует сканированию извне, когда хосту специально отправляются дейтаграммы с пропусками фрагментов.

В операционных системах семейства Windows (и не только в них, естественно) широко применяется утилита tracert. Она отправляет набор из трех дейтаграмм IP по указанному адресу, но при этом устанавливает поле TTL в заголовках этих дейтаграмм в ноль. В дейтаграммах находятся сообщения ICMP echo request. Маршрутизатор перед обработкой дейтаграммы проверяет это поле и, если оно нулевое, отбрасывает дейтаграмму и посылает рассматриваемое сообщение его отправителю — в данном случае программе tracert. Затем программа tracert посылает другой набор дейтаграмм, но с увеличенным на единицу полем TTL, и все повторяется заново. При этом на экран выводится информация о промежуточных узлах в пути до получателя. Подобные итерации будут продолжаться

до тех пор, пока не будет получен ответ на дейтаграммы (не придет ответ ICMP echo reply или сообщение ICMP destination unreachable). Внутри дейтаграмм вместо сообщений ICMP можно использовать протокол UDP с портом получателя, выбранным из числа свободных. При этом появляется возможность проведения анализа правил фильтрации на защитном экране. С помощью tracert (точнее, ее аналога, который умеет посылать UDP-пакеты) генерируются пакеты UDP с интересующими номерами портов получателя. Если порт блокирован защитным экраном, то трассировка остановится на нем, иначе (считая, что защитный экран пропустит в Интернет сообщение ICMP TTL exceeded) защитный экран на мониторе компьютера появится как еще один переход (иными словами как маршрутизатор). Инструмент, способный выполнять трассировку с помощью пакетов UDP, можно найти на сайте http://www.packetfactory.net — авторы Майк Шифман и Дэвид Голдсмит (программа Firewalk).

Фрагментация дейтаграмм IP

Фрагментирование происходит в тот момент, когда дейтаграмме протокола IP, проходящей по сети, необходимо пересечь сегмент этой сети, в котором значение MTU меньше размера самой дейтаграммы. Например, значение MTU для дейтаграммы в сети Ethernet составляет 1500 байт, и если размер дейтаграммы больше этого значения, потребуется ее фрагментация. Это сделает маршрутизатор, подключенный к сети, в которую нужно доставить дейтаграмму. Кроме маршрутизатора фрагментацию может также выполнить хост, если ему нужно передать дейтаграмму с размером, превышающим MTU. Каждый фрагмент исходной дейтаграммы проходит по сети до получателя, который затем собирает полученные фрагменты вместе, формируя исходную дейтаграмму. Интересно отметить, что сами фрагменты могут еще раз пройти процедуру фрагментации, если на своем пути миновали сегмент сети с еще меньшим MTU, чем их размер. В целом, фрагментация — нормальное явление, однако, как и в случае протокола ICMP, нормальная функциональность может быть востребована для ненормальной деятельности. Например, можно разбивать дейтаграммы на части так, чтобы фрагменты прошли через фильтрующий маршрутизатор, который не поддерживает работу с фрагментами.

Каждый полученный фрагмент должен ассоциироваться с другим фрагментом при помощи общего идентификатора (так называемого common fragment identification number). Для этих целей служит специальное поле в заголовке протокола IP — IP identification number. Далее, каждый фрагмент должен содержать в себе свою позицию (смещение) в оригинальной, нефрагментированной дейтаграмме и дополнительно указывать длину переносимых им данных. И, наконец, каждый фрагмент должен знать, следуют ли за ним еще фрагменты. За это отвечает флаг More Fragments (MF) в заголовке IP. Таким образом, вся необходимая информация для последующей сборки содержится в заголовке дейтаграммы.

Наиболее наглядно можно показать механизм фрагментации с помощью перехватов пакетов. Предположим, что используется сеть Ethernet, в которой значение

MTU составляет 1500 байт. Передаваемая в сети дейтаграмма инкапсулируетс в кадр сети Ethernet, причем сначала идет заголовок, обычный размер кот рого равен 20 байт, но может быть и больше в зависимости от настроек. Есл приложение оперирует протоколами верхних уровней, например TCP, то се мент TCP, в свою очередь, инкапсулируется в дейтаграмму после ее заголовк Простая арифметика показывает, что если длина заголовка дейтаграммы соста ляет 20 байт, то остается еще 1480 байт для упаковки данных других прото колов. Для иллюстрации всех этапов фрагментирования предположим, ч администратор выполняет команду ping, получая следующие результаты:

```
C:\Program Files>ping -l 4028 192.168.3.254

Обмен пакетами с 192.168.3.254 по 4028 байт:

Ответ от 192.168.3.254: число байтов=4028 время=10мс TTL=128
Ответ от 192.168.3.254: число байтов=4028 время<10мс TTL=128
...
```

Параметр –l 4028 дает указание программе ping сформировать запрос ICMP echo request с размером передаваемых данных, равным 4028 байт. Говоря о нор мальной сетевой активности, подобные запросы ICMP не должны встречаться – они чересчур большие. В сети Ethernet результирующая дейтаграмма будет поделена на фрагменты по 1500 байт или меньше. Каждый из этих фрагмен тов будет иметь 20-байтовый заголовок дейтаграммы IP, по-прежнему оставля 1480 байт для данных (рис. 4.30).

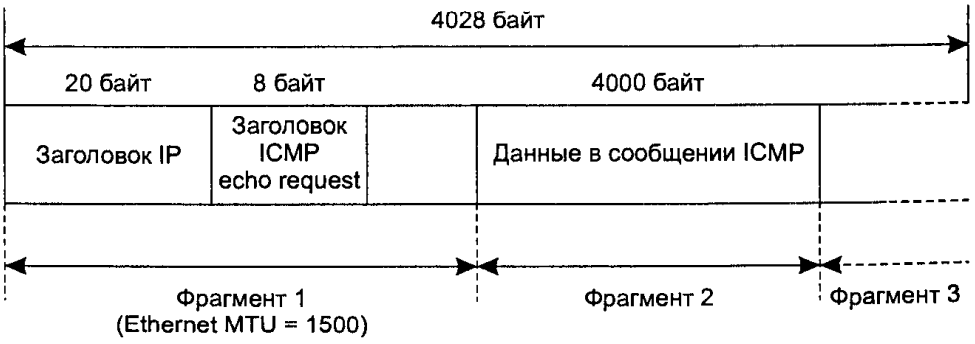


Рис. 4.30. Фрагментация дейтаграммы IP

Следует отметить, что в заголовке любой дейтаграммы IP есть поле с назва нием Identification (идентификатор – главу 2), содержащее номер, уникально идентифицирующий каждую дейтаграмму, посылаемую хостом в сеть. Обычно значение этого поля увеличивается на единицу с каждой следующей дейта граммой. В том случае, когда дейтаграмма делится, все ее фрагменты имеют одинаковый номер, который можно так же назвать идентификатором фрагмен та. Ниже приведен пример перехвата первого из трех фрагмента дейтаграммы (результат выполнения команды ping –l 4028). Все остальные фрагменты дейта граммы будут иметь тот же номер 0x5bc9.

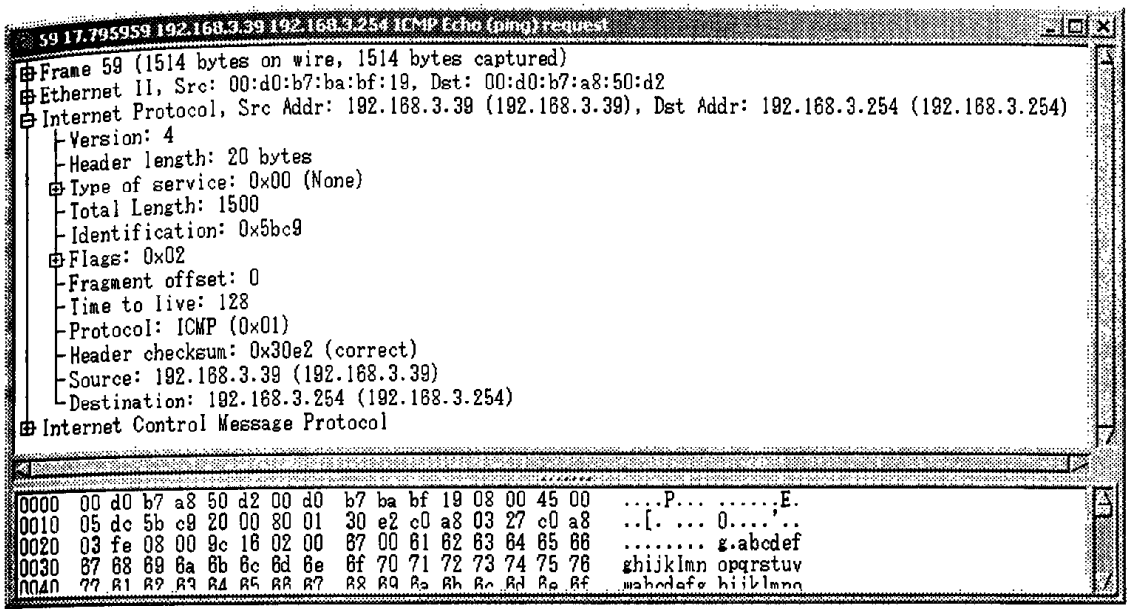


Рис. 4.31. Первый фрагмент дейтаграммы IP

Только первый фрагмент дейтаграммы содержит в себе заголовок протокола ICMP. Как видно из результата перехвата (рис. 4.31), поле, указывающее на смещение фрагмента (Fragment offset), для первого фрагмента установлено в ноль, а поле данных, следующее за 8-байтовым заголовком ICMP, вмещает в себя 1472 байт (1500 – 20 – 8 = 1472). Флаг More fragments («еще фрагменты») установлен, указывая на то, что за этим фрагментом поступят последующие. За первым фрагментом следует второй, который показан на рис. 4.32.

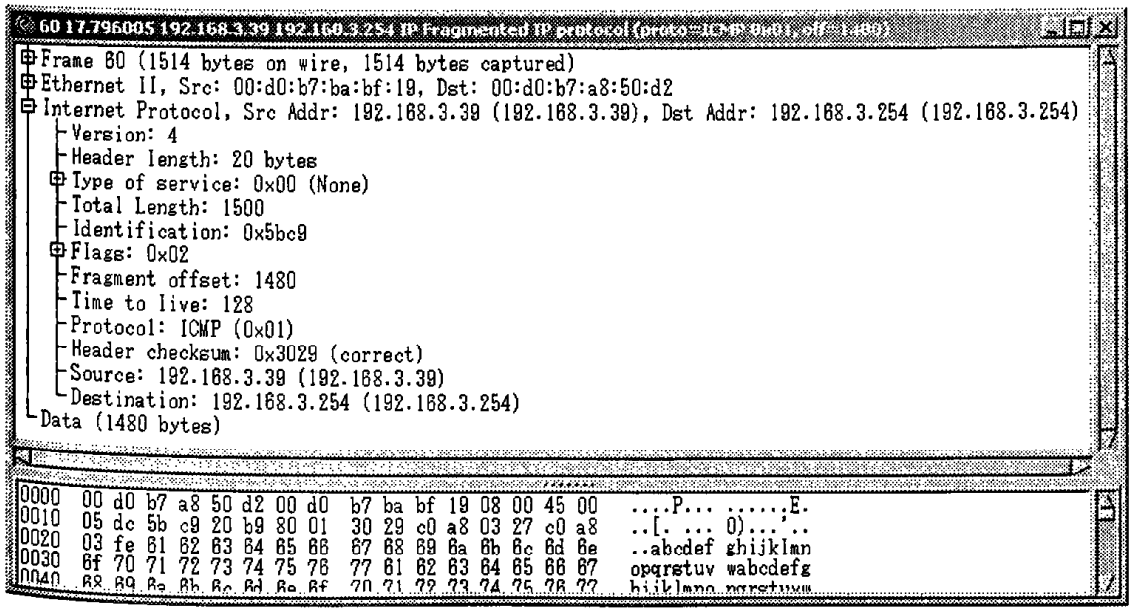


Рис. 4.32. Второй фрагмент дейтаграммы IP

Как уже говорилось, заголовок ICMP будет присутствовать только в перв
фрагменте, тем самым для данных остается на 8 байт больше в последующи
Идентификатор второго фрагмента совпадает со значением того же поля в пе
вом фрагменте. Установлен бит More fragments, указывая получателю на то, ч
он должен ожидать дополнительные пакеты. Другое отличие последующих фра
ментов от первого заключено в поле Fragment offset. Если для первого фрагмен
оно равно нулю, то для остальных оно содержит значение смещения внутри ор
гинальной дейтаграммы. Так как при совмещении фрагментов заголовок втор
го фрагмента нам не нужен, поле смещения содержит значение 1480 (1500
вычетом 20 байт заголовка). Если в сети был перехвачен второй фрагмент б
первого, то невозможно определить тип сообщения протокола ICMP, так к
его заголовок утерян. Рассмотрим формат третьего, последнего фрагмента в н
шем примере (рис. 4.33).

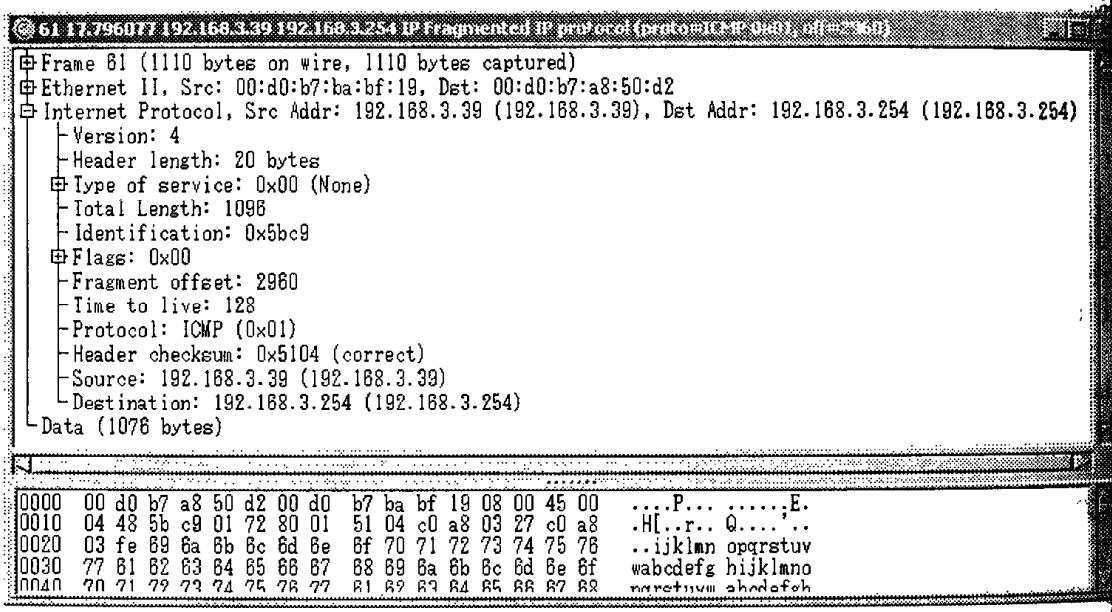


Рис. 4.33. Третий фрагмент дейтаграммы IP

Здесь также имеется заголовок протокола IP со все тем же значением в пол
Identification, однако флаг, указывающий на наличие дополнительных фрагмент
не установлен. Смещение в оригинальной дейтаграмме составляет 2960 бай
что равно сумме двух первых фрагментов по 1480 байт. После получения вс
трех фрагментов хост собирает их вместе в своей памяти, обнаруживает, что эт
запрос ICMP echo request, и посылает в ответ собранную дейтаграмму, но тол
ко с новыми заголовками IP и ICMP. То есть в ответ на три фрагмента пост
пят тоже три фрагмента с сохраненным содержимым поля данных сообщен
ICMP.

Важно вновь отметить, что только первый фрагмент из всей цепочки соде
жит информацию о передаваемом протоколе, например TCP или ICMP. То ес
если в сети используется фильтрующее устройство, например маршрутизато
то только первый фрагмент будет блокироваться. Это при условии, что устро

ство недостаточно «интеллектуально» для проверки полей в каждом фрагменте. В идеальном же случае после получения первого фрагмента нужно ожидать последующие и не пропускать их в сеть. К сожалению, при интенсивном потоке трафика выполнять подобную проверку становится очень сложно по причине большой занятости ресурсов. Если блокирующее устройство не отслеживает все фрагменты, то оно отбросит только первый с информацией о переносимом протоколе, а остальные пропустит в сеть. Ведь часто решение о том, пропускать или не пропускать дейтаграмму, принимается при анализе портов на стороне получателя заголовков протоколов TCP или UDP. Кроме того, так как протокол IP не гарантирует доставку, то в распределенной сети существует вероятность того, что первый фрагмент будет потерян. С учетом подобной возможности задача для блокирующего устройства усложняется.

Если необходимо, то фрагментацию дейтаграммы можно не выполнять, и для этого достаточно установить флаг Don't fragment (DF) в заголовке протокола IP. Если, например, сформировать дейтаграмму, размер которой заведомо больше допустимого MTU в сегменте распределенной сети, то маршрутизатор отбросит подобную дейтаграмму, а в ответ пошлет сообщение ICMP destination unreachable. Это сообщение будет содержать в себе размер MTU сети, которая требует фрагментации. Интересно отметить, что некоторые хосты специально посылают в сеть дейтаграммы с установленным флагом DF с целью определения оптимального размера пакетов в пути до определенного получателя. Это делается для того, чтобы избежать фрагментации, поскольку сам процесс разбиения на фрагменты и их последующая сборка дополнительно нагружают хосты. Более того, они вносят хоть и небольшие, но дополнительные издержки, так как 20-байтовый заголовок протокола IP посылается с каждым фрагментом. Давайте посчитаем — команда `ping -l 4028` четыре раза послала по три фрагмента и четыре раза их приняла. Грубо говоря, накладных расходов было (4×3×20) + (4×3×20) = 480 байт. Из этих 480 байт только малая часть несет полезную информацию, а информация в остальных совпадает. Получается, что при пересылке 4028 байт 240 байт будут служебными, что составляет примерно 6 %. Кроме того, если по каким-либо причинам один фрагмент был потерян, все фрагменты должны посылаться вновь. Если говорить о воздействии на сеть, то атакующий может задействовать бит DF для определения значения MTU на разных участках распределенной сети.

Как уже отмечалось, фрагментация может использоваться злоумышленником. С помощью очень популярной программы nmap (или nmapnt) нетрудно воплотить все вышеизложенные общие рассуждения в практические действия и выполнить сканирование для определения номеров открытых портов на целевом хосте. Если воспользоваться параметром `-f`, то программа будет разбивать заголовок протокола TCP (20 байт) на несколько фрагментов с целью избежать обнаружения сканирования системами IDS. Например, в результате команды `nmapnt -Sf -p 53 host` хосту будут посланы на порт 53 фрагментированные запросы SYN. С точки зрения сетевого уровня нельзя с уверенностью сказать, что происходит сканирование, из-за наличия большого количества маленьких фрагментов. Заголовок дейтаграммы IP повторяется, но заголовок TCP (минимальная длина 20 байт) разбивается на кусочки. В первом фрагменте посылаются 16 байт заголовка TCP, а во втором фрагменте — дополнительные 4 байт.

В зависимости от используемой системы, IDS может и не обнаружить подоб ный тип сканирования.

В заключение рассмотрения протокола ICMP хочется задержаться на мо менте, связанном с блокированием исходящего и входящего трафика ICMP н защитном экране. Если говорить об исходящем трафике, то тут ситуация не сложная — можно блокировать все сообщения ICMP за исключением ICMP echo request — именно этот тип используется сетевыми администраторами как наиболее удобный для проверки работоспособности доступа в сеть Интернет. То есть на защитном экране можно разрешить прохождение запросов ICMP echo — в зависимости от типа защитного экрана, ответы ICMP reply будут прохо дить автоматически, в рамках одного сеанса. Сказанное относится также и к по пулярной утилите tracert, которая при своем запуске в операционной системе Windows начнет посылать дейтаграммы IP с вложенными сообщениями ICMP echo request, которые также не будут задержаны защитным экраном. Однако если для трассировки используются вложенные сообщения, не относящиеся к ICMP, то нужно скорректировать правила фильтрации. Фильтрация всех типов сообщений ICMP, за исключением echo request, не позволит вашим сер верам реагировать на нештатные или ошибочные ситуации, если трафик прихо дит из сети Интернет. Другая проблема имеет место тогда, когда внутренний хост (сервер), взаимодействуя с каким-то устройством в сети Интернет, получа ет сообщение ICMP, на которое он может прореагировать (например, перестать посылать данные в ответ на сообщение ICMP host unreachable). Это сообщение будет отфильтровано защитным экраном. Вы не должны рассчитывать, что за щитный экран будет достаточно «умным», чтобы проанализировать оригиналь ные заголовки в сообщении ICMP и понять, какие из них «свои», и пропустить дейтаграмму во внутреннюю сеть.

5 Настройка протокола TCP

В этой главе дано подробное описание алгоритмов, лежащих в основе протокола TCP, выделены наиболее критичные параметры и определена степень их влияния на производительность TCP. Кроме того, рассмотрена сетевая атака TCP SYN flooding, использующая особенности работы протокола TCP. Освещены методы противодействия этой атаке. В качестве иллюстрации методов настройки протокола приведены примеры настройки для операционных систем Microsoft Windows NT и 2000, причем возможности последней рассматриваются отдельно, так как наиболее широкий список дополнений и расширений протокола TCP компанией Microsoft представлен именно в операционной системе Windows 2000. Эти знания должны помочь администратору оптимально настроить протокол в своей сети с учетом риска потенциальных атак. И конечно, знание процессов, происходящих при транспортировке данных в сетевой среде, является необходимым условием понимания работы в сети в целом.

Введение в протокол TCP

Очень кратко напомним основы протокола TCP (Transmission Control Protocol — протокол управления передачей данных). Описание протокола TCP приводится в документе RFC 793 (в дальнейшем предлагались различные усовершенствования, отраженные во множестве других RFC). Протокол TCP — это основной транспортный протокол в стеке TCP/IP. Он обеспечивает надежную передачу потока данных на базе ненадежного сервиса транспортировки дейтаграмм, предоставляемого протоколом IP. На протокол TCP, в частности, возложена задача управления потоками и перегрузками; этот протокол отвечает за согласованность скорости передачи данных с техническими возможностями получателя и промежуточных устройств на пути прохождения пакетов.

Поступающие данные буферизуются средствами протокола TCP. Для передачи на сетевой уровень из буфера берется определенная непрерывная часть данных, которая называется сегментом. Сегмент является единицей данных протокола TCP. Определение размеров сегментов TCP осуществляет протокол IP. Не все сегменты, посланные через установленное логическое соединение, будут одного и того же размера. Однако два абонента, осуществляющие связь при помощи

логического соединения, должны договориться о максимальном размере сегмента, который они будут использовать. Размер сегмента выбирается таким образом, чтобы при его упаковке в IP-дейтаграмму он поместился в нее целиком.

Информацию на компьютере-получателе принимает модуль протокола TCP. Этот модуль, в свою очередь, помещает данные сегмента в буфер прикладной программы получателя и оповещает его о прибытии данных. С каждым модулем протокола TCP связан модуль протокола IP, который обеспечивает передачу по локальной сети. При этом происходит инкапсуляция сегмента протокола TCP в IP-дейтаграмму. Эта дейтаграмма, в свою очередь, помещается в кадр, регламентированный сетевым окружением.

Надежная передача данных по протоколу TCP осуществляется благодаря подтверждениям, высылаемым получателем после приема данных, и механизму нумерации. Для этого в заголовке сегмента протокола TCP с общим размером 20 байт выделяется специальное поле «Номер [в] последовательности» (Sequence Number — SN). Это поле определяет номер первого байта текущего сегмента в общей очереди (или последовательности) байтов. Исключение составляют случаи, когда выставлен бит (флаг) синхронизации SYN. Тогда в это поле помещается начальный номер последовательности (Initial Sequence Number — ISN), и первый байт данных получает номер ISN + 1.

Так как все сегменты пронумерованы, то каждый из них может быть опознан. Механизм распознавания носит накопительный характер, то есть получение и опознание байта с номером N означает, что все байты с предыдущими номерами (N – 1, N – 2, ...) уже получены и опознаны. Этот механизм позволяет регистрировать появление дублирующих байтов при повторной передаче. Диапазон возможных номеров байтов лежит в пределах от 0 до 4 294 967 295 ($2^{32} - 1$). Так как набор номеров ограничен, то все арифметические операции с номерами последовательностей должны осуществляться по модулю 2^{32}.

Номера не должны исчерпать все значения до 2^{32}, прежде чем связанные с ними данные из отправляемого сегмента получат подтверждение от получателя, а все дубликаты этого сегмента покинут сеть. Иначе двум сегментам могут быть назначены одинаковые номера, что вызовет проблему с идентификацией этих сегментов у получателя. При скорости обмена информацией в 100 Мбит/с все возможные номера перебираются за 5,4 минуты.

Кроме того, сегменты несут номера подтверждений, которые являются номерами для следующих ожидаемых байтов данных, передаваемых в обратном направлении. Для этого в заголовке сегмента выделено поле «Номер подтверждения» (Acknowledgment Number), содержащее номер в последовательности получаемых подтверждений, который отправитель ожидает в ответ на посылаемый сегмент. Поступление подтверждения означает успешную доставку этого сегмента. Признаком подтверждения служит установленный контрольный бит ACK. Подтверждения (или, как часто говорят, ACK) посылаются постоянно с момента установления соединения TCP и до его закрытия.

Для определения назначения большинства полей заголовка TCP служат контрольные биты (флаги): URG, ACK, PSH, RST, SYN и FIN. Соединения протокола TCP переходят из одного состояния в другое в ответ на определенные события — запросы клиента, приход сегментов с флагами SYN, ACK, RST, FIN — или по истечении заданного времени. Условно работу соединения можно разде-

лить на три стадии: установление соединения, рабочее состояние и закрытие соединения. Основное состояние — ESTABLISHED (обе системы готовы к приему и передаче данных приложения). В этом состоянии осуществляется обмен данными между абонентами. При создании соединения происходит так называемая синхронизация номеров.

Рассмотрим, например, процесс взаимодействия сервера и клиента по протоколу TCP. Для установления соединения необходимо выполнить следующие шаги.

1. Клиент посылает серверу сегмент с флагом SYN и своим номером в последовательности (Sequence Number) N.
2. Сервер отправляет клиенту подтверждение (Acknowledgment Number): «Ваш номер в последовательности — N», — и сообщает свой номер в последовательности (обозначим его K).
3. Клиент посылает серверу подтверждение: «Ваш номер в последовательности — K».

Теперь используем реальные номера. Выделим три шага в процессе установления соединения TCP. В этом примере клиент инициирует подключение по протоколу HTTP к серверу с портом назначения 8080. На первом шаге клиент посылает сегмент протокола TCP с установленным флагом SYN и своим номером в последовательности — 1362555259. На рис. 5.1 показаны поля заголовка сегмента TCP на первом этапе установления соединения. Запись получена с помощью анализатора протоколов eEye (полное название eEye Iris Network Traffic Analyzer, пробную версию программы можно получить по адресу www.eeye.com после регистрации).

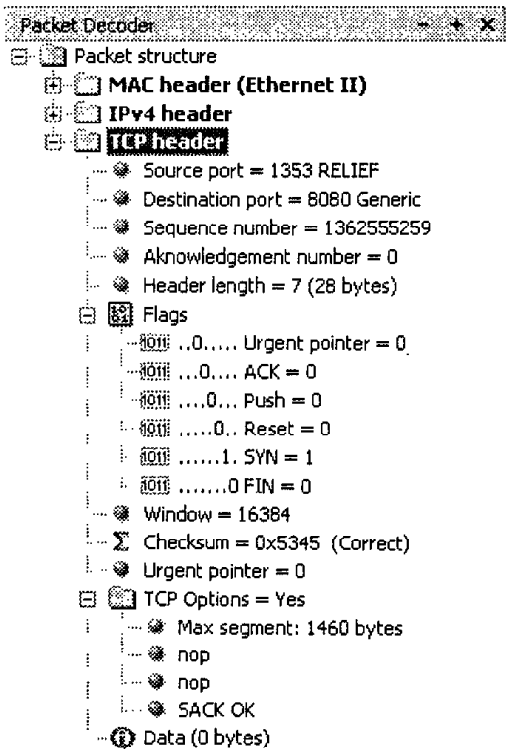


Рис. 5.1. Первый шаг установления соединения

На втором шаге сервер посылает клиенту сегмент, подтверждающий получение запроса (флаг ACK = 1) на установление соединения, с указанием своего номера — 88315884. Кроме того, сервер устанавливает поле Acknowledgment Number в значение, которое на единицу больше полученного номера в последовательности — 1362555260. Тем самым сервер показывает, что он ожидает от клиента следующий сегмент (рис. 5.2).

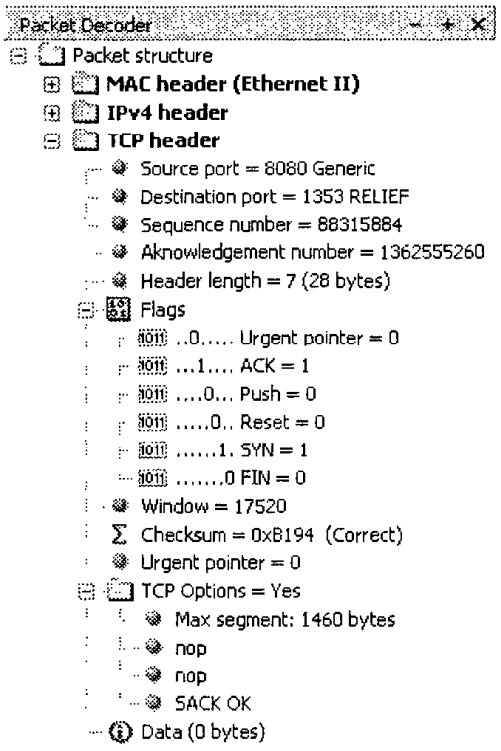


Рис. 5.2. Второй шаг установления соединения

На третьем шаге (рис. 5.3) клиент высылает серверу сегмент, в котором поле номера в последовательности устанавливается равным значению поля Acknowledgment Number, полученного от сервера — 1362555260. Кроме того, клиент подтверждает получение сегмента от сервера, присваивая полю Acknowledgment Number значение, на единицу больше полученного номера в последовательности — 88315885.

Следует обратить внимание на то, что в процессе установления соединения клиент и сервер согласовывают размер скользящего окна (Window = 17520). Данные при установлении соединения не передаются (Data = 0 bytes). На стороне клиента работает порт 1925, на стороне сервера — 8080.

При обмене сегментами в процессе установления соединения полю Maximum Segment Size (MSS) задано значение 1460, что соответствует максимальному размеру сегмента. Как будет показано далее, согласованный размер окна (Window = = 17520) должен быть кратен этому значению, что и подтверждается на практике (17 520 / 1460 = 12).

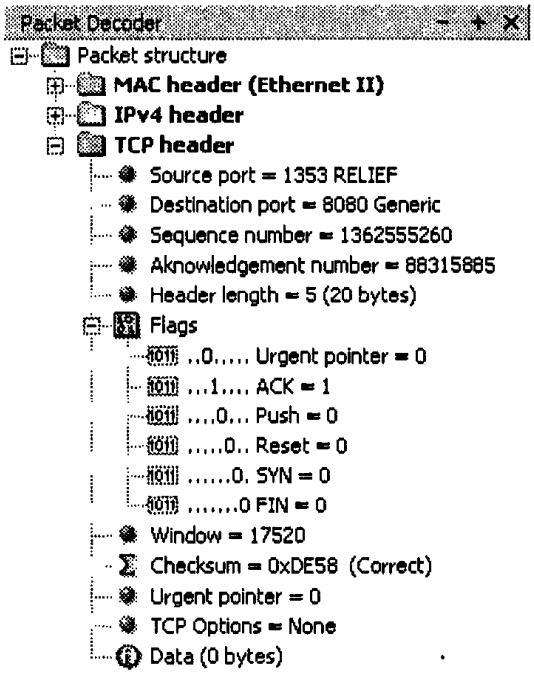


Рис. 5.3. Третий шаг установления соединения

ПРИМЕЧАНИЕ

Администратор может установить временной интервал, в течение которого маршрутизатор будет ожидать возможных попыток установления соединения по протоколу TCP. В ранних версиях операционной системы Cisco IOS этот интервал времени был фиксирован и равнялся 30 секундам. Такое значение мало для сетей, в которых используются медленные соединения, например асинхронные на аналоговых модемах. Значение этого интервала влияет только на трафик, адресованный самому маршрутизатору, но не на проходящий через него. Для того чтобы изменить интервал ожидания, нужно воспользоваться командой ip tcp synwait-time, выполняемой в глобальном режиме с параметром, задающим количество секунд ожидания синхронизации. Изменять значение по умолчанию стоит только в том случае, если администратор испытывает проблемы связи с маршрутизатором, например при работе с удаленным маршрутизатором по медленному каналу связи через сеанс Telnet.

При передаче протоколом TCP сегментов с данными он помещает их копии в очередь повторной передачи и запускает таймер. После прихода подтверждения для очередного сегмента соответствующая копия удаляется из очереди повторной передачи. Если подтверждение не приходит до истечения срока, заложенного в таймер, то сегмент посылается повторно. Повреждения сегментов при передаче фиксируются за счет добавления к каждому передаваемому сегменту контрольной суммы, проверки ее при получении и последующего удаления дефектных сегментов.

Как уже отмечалось, продолжительность жизни дейтаграммы протокола IP не должна превышать двух минут. Следовательно, когда протокол TCP отправляет с дейтаграммой протокола IP некоторый байт данных, порядковый номер этого байта сможет быть повторно использован не ранее, чем через две минуты. В результате 32-разрядного номерного пространства при двухминутном цикле хватает лишь на то, чтобы достичь скорости передачи данных 286 Мбит/с.

Номер в последовательности может рассматриваться как простой 32-разряд
счетчик, который принимает любые значения в интервале от 0 до 4 294 967
Для администратора сети очень важно понимать, каким образом изначально
бирается значение номера в последовательности и как оно изменяется во врем
Когда клиент начинает работу, его ISN устанавливается в единицу, и значе
счетчика затем увеличивается каждую секунду на 128 000. Сброс счетчика бу
производиться каждые 9,32 часа при условии, что соединение устанавливаться
будет. При установлении соединения значение счетчика увеличивается на 64

Следует отметить, что из-за довольно большой популярности атак, основан
на предсказании начального номера в последовательности, компания Micro
внесла определенные изменения в свою реализацию стека протоколов TCP
для операционной системы Windows 2000. Процесс увеличения начальн
номера в последовательности ISN носит непредсказуемый характер. При э
используется основанный на алгоритме RC4 механизм генерации случай
чисел, инициализируемый с 2048-разрядным случайным числом при старте с
темы. К сожалению, более подробной информации о механизме формирова
ISN в операционной системе Windows 2000 автору найти не удалось.

Атака TCP SYN flooding

Рассмотрим механизм атаки TCP SYN flooding («шторм» запросов на устано
ние соединения протокола TCP), которая, пожалуй, является наиболее попу
ной в семействе атак типа DoS. Напомним, что установление соединения ос
вано на обмене информацией между клиентом и сервером. При этом сущест
некий максимально возможный предел (лимит) для количества одновреме
обрабатываемых протоколом TCP запросов SYN. Эта предельная норма назы
ется backlog. По сути, она представляет собой очередь, в которую помеща
входящие незавершенные запросы на соединение. Предельная норма этой очер
объединяет как незавершенные запросы на создание соединения, так и заверш
ные запросы на создание соединений, которые еще не переданы приложению с
мощью соответствующей процедуры. Если произошло превышение лимита
протокол TCP начнет отбрасывать новые входящие запросы SYN. Это будет
должаться до тех пор, пока в очереди не освободится место. И именно такая с
работы протокола TCP создает реальную возможность для проведения атак

ПРИМЕЧАНИЕ

Атаки, организованные по типу «отказ в обслуживании» (Denial of Service — DoS) высветили
одну проблему в защите, причем такую, для которой простого решения не существует. В об
случае атаки этого типа проводятся посредством генерирования большого объема трафика, за
мающего всю полосу пропускания или ресурсы атакуемого компьютера, в результате чего др
пользователи не могут получить к нему доступ. Чаще всего атаки направлены против web-се
ров, хотя базовый принцип позволяет проводить атаки практически против любого доступ
устройства в сети, которое способно отвечать на посылаемые запросы. Например, это может
массовая посылка запросов ping (точнее, ICMP echo) или запросов на установление соедин
протокола TCP.

Стек протоколов TCP/IP, реализованный в операционной системе Windows 2000 для отслеживания запросов к приложениям, взаимодействующим через интерфейс Windows Sockets, использует функцию listen(). Одним из передаваемых параметров при вызове этой функции является параметр backlog, определяющий количество незавершенных запросов, которые приложение хочет поместить в очередь. После того как приложение устанавливает соединение, его запросы удаляются из очереди. В спецификации Windows Socket 1.1 максимальное допустимое значение для backlog равно 5 запросам. Версия операционной системы Windows NT 3.51 могла иметь значение backlog до 100 запросов. Версии NT 4.0/2000 Server допускают до 200 запросов, в то время как NT 4.0 Workstation и Windows 2000 Professional ограничивают это значение числом 5.

Атакующий компьютер посылает огромное количество запросов на установление соединения TCP (или коротко — запросов SYN) компьютеру, который следует вывести из строя. При этом в качестве адреса отправителя указывается любой адрес (может быть адрес 0.0.0.0), так как атака не подразумевает какой-либо обмен между атакуемым и атакующим компьютером. Естественно, если был указан несуществующий адрес, то протокол IP будет информировать TCP о недостижимости получателя. Но протокол TCP будет считать, что эта ошибка должна быть исправлена протоколом IP, например с помощью выбора другого маршрутизатора для отправки, то есть фактически протокол TCP проигнорирует недостижимость. Более того, важна именно сама недостижимость получателя, так как в другом случае атакуемый компьютер будет отправлять сегменты с установленными флагами SYN/ACK (второй шаг в процессе установления соединения) по указанному адресу и в ответ может получить RST, что в силах помешать атаке.

Таким образом, атакующий компьютер, посылая массированные запросы SYN, заполнит очередь протокола TCP на атакуемом компьютере, что приведет к тому, что последний будет игнорировать любые другие запросы на данном порте. Разные реализации протокола TCP имеют разные размеры очереди. Если говорить об операционной системе Windows NT 4.0, то реализация протокола TCP (при конфигурации по умолчанию) посылает в ответ сегменты с флагами SYN/ACK 5 раз, удваивая значение тайм-аута после каждой передачи. Начальное значение таймера устанавливается в 3 секунды, таким образом, повторные запросы будут переданы на 3, 6, 12, 24, и 48-й секундах. После последней повторной передачи должно пройти 96 секунд, по прошествии которых протокол TCP перестанет ожидать ответ и освободит место в очереди. Таким образом, общее время использования ресурсов под открытие несостоявшегося соединения будет составлять 189 секунд. Пользователь, подозревающий, что его компьютер подвергается подобной атаке, в состоянии проверить этот факт, выполняя команду netstat –n –p tcp. В случае организации реальной атаки на экране можно получить результат, показанный в следующем примере. Этот пример взят из статьи Q142641 Microsoft Knowledge Base, которая была опубликована 4 июля 2000 года:

```
Proto Local Address          Foreign Address        State
  TCP   127.0.0.1:1030       127.0.0.1:1032         ESTABLISHED
  TCP   127.0.0.1:1032       127.0.0.1:1030         ESTABLISHED
  TCP   10.57.8.190:21       10.57.14.154:1256      SYN_RECEIVED
  TCP   10.57.8.190:21       10.57.14.154:1257      SYN_RECEIVED
```

```
TCP    10.57.8.190:21         10.57.14.154:1258      SYN_RECEIVED
TCP    10.57.8.190:21         10.57.14.154:1259      SYN_RECEIVED
TCP    10.57.8.190:21         10.57.14.154:1260      SYN_RECEIVED
TCP    10.57.8.190:21         10.57.14.154:1261      SYN_RECEIVED
TCP    10.57.8.190:21         10.57.14.154:1262      SYN_RECEIVED
TCP    10.57.8.190:21         10.57.14.154:1263      SYN_RECEIVED
TCP    10.57.8.190:21         10.57.14.154:1264      SYN_RECEIVED
TCP    10.57.8.190:21         10.57.14.154:1265      SYN_RECEIVED
TCP    10.57.8.190:21         10.57.14.154:1266      SYN_RECEIVED
TCP    10.57.8.190:4801       10.57.14.221:139       TIME_WAIT
```

Анализируя результат выполнения команды, можно увидеть большое количество состояний SYN_RECEIVED (индикация получения запроса SYN), соответственно возникает законное подозрение о проведении атаки на данный компьютер.

Кроме рассмотренного изменения в реализации стека TCP/IP, компания Microsoft реализовала еще ряд расширений. Так, администратору доступен специальный параметр в реестре операционной системы — SynAttackProtect, установка которого избавляет администратора от необходимости проводить корректировку других значений параметров протокола TCP для достижения максимальной защиты (табл. 5.1).

Таблица 5.1. Параметр SynAttackProtect

Название параметра	SynAttackProtect
Ключ в реестре	Tcpip\Parameters
Тип записи	REG_DWORD
Диапазон значений	0, 1, 2
Значение по умолчанию	0

По умолчанию этот параметр реестра 0 (защиты от атаки TCP SYN flooding нет), в то время как рекомендованным является значение 2. При SynAttackProtect = будут уменьшаться интервалы между повторными передачами и будет задерживаться создание записи о маршруте соединения — RCE (route cache entry), если параметры TcpMaxHalfOpen и TcpMaxHalfOpenRetried адекватны угрозе проведения атаки (ниже описание этих параметров в реестре). При SynAttackProtect = 2 в дополнение к действиям, выполняемым при SynAttackProtect = 1, включается индикация о задержке для приложений.

При этом, как видно из описания, система будет уменьшать количество повторных передач сегментов с флагами SYN/ACK. Дополнительно будет сокращено время, в течение которого система поддерживает информацию о соединении в очереди. Это позволит быстрее очищать очередь от незавершенных соединений. Кроме того, выделение записи RCE задерживается до тех пор, пока соединение не будет установлено. Обратите внимание на то, что этот защитный механизм начинает работать только тогда, когда превышены значения параметров TcpMaxHalfOpen и TcpMaxHalfOpenRetried для соединений протокола TCP. Первый параметр (TcpMaxHalfOpen) контролирует количество незавершенных (так называемых half-open) запросов на установление соединения до момента, когда оп

рационная система будет считать, что она подвергается атаке TCP SYN flooding. Если параметр реестра SynAttackProtect равен 1, то администратор сети должен убедиться, что значение рассматриваемого параметра меньше, чем размер очереди backlog (табл. 5.2).

Таблица 5.2. Параметр TcpMaxHalfOpen

Название параметра	TcpMaxHalfOpen
Ключ в реестре	Tcpip\Parameters
Тип записи	REG_DWORD
Диапазон значений	0x64– 0xFFFF
Значение по умолчанию	0x64 (100) для Microsoft Windows Professional, Server и 0x1F4 (500) для Advanced Server

Параметр TcpMaxHalfOpenRetried (табл. 5.3) призван контролировать количество незавершенных соединений, для которых была выполнена по крайне мере одна повторная посылка SYN, до того момента как начнет работать защита от атаки.

Таблица 5.3. Параметр TcpMaxHalfOpenRetried

Название параметра	TcpMaxHalfOpenRetried
Ключ в реестре	Tcpip\Parameters
Тип записи	REG_DWORD
Диапазон значений	0x50–0xFFFF
Значение по умолчанию	0x50 (80) для Microsoft Windows Professional, Server и 0x190 (400) для Advanced Server

Если администратор считает, что установки только этого параметра в реестре недостаточно для защиты от атак TCP SYN flooding, то он может скорректировать другой параметр — TcpMaxConnectResponseRetransmissions, влияющий на количество попыток отправления сегментов с флагами SYN/ACK в ответ на запрос SYN (табл. 5.4).

Таблица 5.4. Параметр TcpMaxConnectResponseRetransmissions

Название параметра	TcpMaxConnectResponseRetransmissions
Ключ в реестре	Tcpip\Parameters
Тип записи	REG_DWORD
Диапазон значений	0–0xFFFFFFFF
Значение по умолчанию	3

Значение по умолчанию для данного параметра равно 3, но в распоряжении администратора есть еще два режима — 1 и 2. В табл. 5.5 показано поведение протокола TCP при различных значениях рассматриваемого параметра реестра.

Таблица 5.5. Влияние параметра TcpMaxConnectResponseRetransmissions на соединение

Значение	Повторные передачи через	Общее время	Сброс незавершенных сеансов через
1	3 секунды	9 секунд	6 секунд
2	3 и 6 секунд	21 секунда	12 секунд
3	3, 6 и 12 секунд	45 секунд	24 секунды

Данный параметр изменяет время по умолчанию, которое требуется для сбр са незаконченных сеансов TCP за период времени между 9-й и 45-й секундам и предоставляет администратору возможность более точного контроля. В том сл чае, если на сервер постоянно происходят атаки SYN flooding (факт можно об наружить, задействуя систему обнаружения атак — Intrusion Detection Syste IDS), администратор вправе установить данный параметр в 1, максимально сни жая тем самым время сброса незавершенных сеансов. Администратор может уст новить параметр и в 0, но в этом случае сегменты с флагами SYN/ACK вооб не будут передаваться повторно и соединение будет завершаться через 3 секун Обратите внимание, что, обеспечивая наиболее сильную защиту от атаки SY flooding, вы вносите вероятность проблем с удаленными клиентами на плохи линиях связи, когда возможна потеря сегментов при передаче и в процессе уст новления соединения.

Приложения, работающие с Windows Sockets, такие как серверы FTP и WWW обрабатывают попытки подключения по протоколу TCP с помощью драйвер afd.sys, который был изменен для поддержки большого количества незавершен ных соединений без отказа в обслуживании. Это достигается включением сис темный реестр параметров, позволяющих администратору сети настраивать ди намическую очередь backlog.

Параметр реестра EnableDynamicBacklog позволяет глобально включать или вы ключать динамическую очередь backlog. По умолчанию этот параметр равен 0, что означает отключение очереди. Однако если система часто подвергается ма сированным атакам, то рекомендуется выставлять значение 1 (табл. 5.6).

Таблица 5.6. Параметр EnableDynamicBacklog

Название параметра	EnableDynamicBacklog
Ключ в реестре	\SYSTEM\CurrentControlSet\Services\AFD\Parameters
Тип записи	REG_DWORD
Диапазон значений	0, 1
Значение по умолчанию	0

Параметр MinimumDynamicBacklog контролирует минимальное количество сво бодных соединений, разрешенных на сервере. Если количество свободных со единений (или, иными словами, место в очереди backlog) снижается на вел чину, меньшую, чем значение этого параметра, то формируется системны запрос на создание дополнительных свободных соединений. Значение парам

ра MinimumDynamicBacklog не должно быть большим, так как это может привести к снижению производительности (табл. 5.7).

Таблица 5.7. Параметр MinimumDynamicBacklog

Название параметра	MinimumDynamicBacklog
Ключ в реестре	\SYSTEM\CurrentControlSet\Services\AFD\Parameters
Тип записи	REG_DWORD
Диапазон значений	0–0xFFFFFFFF
Значение по умолчанию	0

По умолчанию значение этого параметра равно 0, но для серверов, которые часто подвергаются атакам, его можно увеличить до 20.

Параметр MaximumDynamicBacklog контролирует количество так называемых квазисвободных (quasi-free) соединений, разрешенных на сервере. Квазисвободные соединения включают в себя и свободные соединения, и соединения, которые находятся в незавершенном состоянии (SYN_RECEIVED). Сервер не будет пытаться создавать дополнительные свободные соединения, если это приводит к превышению данного параметра. Значение параметра MaximumDynamicBacklog не должно превышать 5000 для каждых 32 Мбайт памяти, установленной на сервере. Это сделано для того, чтобы избежать возможной нехватки памяти при осуществлении атаки (табл. 5.8).

Таблица 5.8. Параметр MaximumDynamicBacklog

Название параметра	MaximumDynamicBacklog
Ключ в реестре	\SYSTEM\CurrentControlSet\Services\AFD\Parameters
Тип записи	REG_DWORD
Границы значений	0–0xFFFFFFFF
Значение по умолчанию	0

Параметр DynamicBacklogGrowthDelta задает количество создаваемых свободных соединений, когда необходимы дополнительные соединения. Администратору сети следует очень аккуратно работать с этим параметром, так как большое его значение может привести к взрывному росту объема памяти, выделяемой под свободные соединения. Рекомендованное значение на случай возможной атаки равно 10 (табл. 5.9).

Таблица 5.9. Параметр DynamicBacklogGrowthDelta

Название параметра	DynamicBacklogGrowthDelta
Ключ в реестре	\SYSTEM\CurrentControlSet\Services\AFD\Parameters
Тип записи	REG_DWORD
Диапазон значений	0–0xFFFFFFFF
Значение по умолчанию	0

Чтобы получить максимум преимуществ от изменений в драйвере afd.sy приложения, работающие с Windows Sockets, при формировании запроса listen должны специально запрашивать значение очереди backlog большее, чем значени параметра MinimumDynamicBacklog. Приложения, разработанные компанией Micr soft, среди которых наиболее популярным объектом для организации атак явл ется Internet Information Server (IIS), позволяют производить настройку свои параметров, причем для IIS размер очереди backlog по умолчанию равен 25.

Мы рассмотрели некоторые варианты защиты от атаки TCP SYN flooding н конечных устройствах сети — серверах. Однако на практике редко встречает ситуация, когда серверы, предоставляющие какие-либо услуги внешним польз вателям, были бы установлены в сегменте сети, напрямую подключенном к Инте нету без обеспечения защиты.

Первый барьер на пути потенциальной атаки можно создать на гранично маршрутизаторе компании Cisco Systems (или любом другом, который поддер живает нужные функции). Естественно, самым простым способом защиты явл ется формирование списка управления доступом на маршрутизаторе, которы будет просто блокировать запросы на установление соединения TCP с опреде ленных адресов в сети. Несмотря на абсолютное отсутствие смысла в такой з щите с точки зрения доступности, например, сервера WWW в Интернете, он может быть неплохим решением, если выявлен постоянный источник ненормаль ного количества SYN-запросов TCP.

К сожалению, атака может быть организована и со случайно выбранными ад ресами отправителя. Начиная с версии 11.3 операционной системы IOS админи стратору сети доступен механизм TCP Intercept. Настройка TCP Intercept (пе рехват) на маршрутизаторе препятствует проведению подобной атаки. Кратко опишем схему работы этого механизма. Граничный маршрутизатор перехваты вает запрос SYN на установление соединения TCP от клиента к серверу. Затем маршрутизатор сам устанавливает соединением с клиентом, действуя как бы вместо сервера с последующим связыванием двух незавершенных соединений вместе. Происходит это абсолютно прозрачно как для клиента, так и для серве ра. Важным моментом является то, что SYN-запрос протокола TCP от несуще ствующего клиента в сети никогда не достигнет сервера, тем самым устраняется одно из условий успешной атаки TCP SYN flooding.

При настройке рассматриваемого механизма на маршрутизаторе администр ратор может указать на необходимость перехвата всех запросов TCP SYN ил только тех, которые приходят из определенных сетей или адресованы конкрет ному серверу. Важной особенностью работы механизма перехвата является то что специфичные параметры протокола TCP (например, масштабирование сколь зящего окна), оговариваемые в процессе установления соединения, не будут приниматься во внимание маршрутизатором, так как последний, естественно, не обладает информацией о соответствующих возможностях сервера.

Для того чтобы включить на маршрутизаторе механизм TCP Intercept, нужно сначала сформировать расширенный список доступа (глава 9), определяющий перечень адресов, для которых необходимо проводить перехват запросов TCP SYN. Затем, после определения списка, следует ввести команду ip tcp intercept lis

с указанием номера созданного списка. Рассмотрим пример, который показывает процесс формирования расширенного списка доступа с номером 101, указывающего на необходимость перехвата SYN-запросов, адресованных всем серверам, принадлежащим сети 192.168.3.0/24.

```
ip tcp intercept list 101
!
access-list 101 permit tcp any 192.168.3.0 0.0.0.255
```

ПРИМЕЧАНИЕ

Для получения информации о детальной настройке механизма TCP Intercept можно обратиться по адресу http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scprt3/scdenial.htm.

АТАКИ LAND И STREAM.C

Атака Land относится к атакам типа DoS, и в ее основе лежит отправка массированных запросов SYN (сама атака является развитием атаки TCP SYN Flooding). Причем адрес и номер порта отправителя устанавливаются **совпадающими с адресом и портом получателя**. Подобная атака может заставить некоторые реализации стека TCP/IP войти в цикл, приводящий к сбою атакуемого компьютера или значительному потреблению ресурсов, что негативно влияет на общую производительность. Обнаружение подобной атаки — довольно простая задача, так как совпадение адресов и портов однозначно указывает на эту атаку. Для противодействия можно порекомендовать настроить граничный маршрутизатор или защитный экран на блокировку входящих пакетов, содержащих адреса организации в качестве отправителей.

Атака stream.c также относится к атакам типа DoS. Она нацелена на выведение уязвимой системы из рабочего состояния путем посылки большого количества ложных сегментов протокола TCP с установленным флагом ACK и указанием **случайных номеров портов** на стороне получателя. Системы обнаружения вторжений могут идентифицировать этот тип атаки, анализируя значения в заголовке сегмента TCP, искомый набор которых возможен, но очень редко встречается в обычных пакетах.

Продолжая разговор о защите от атаки TCP SYN flooding, напомним, что администратор сети в состоянии воспользоваться также возможностями защитного экрана. В настоящее время большинство производителей защитных экранов включают необходимые для этого средства в свои продукты. Например, работая с комплексом NetGuard GuardianPro v5.0x, при создании стратегии защиты (здесь под стратегией понимается набор правил фильтрации трафика) администратор может также включить механизм защиты от атаки TCP SYN flooding. На рис. 5.4. представлен фрагмент окна программного комплекса, которое позволяет регулировать параметры защиты.

Так, администратор может указать промежуток времени в секундах, в течение которого защитный экран будет ожидать завершения трехшагового процесса установления соединения TCP (поле Wait for second TCP ACK for ... seconds) до принудительного сброса соединения.

Среди других доступных настроек — количество попыток сброса установления соединения TCP по тайм-ауту до того, как защитным экраном будет принято решение о том, что проводится атака (поле Detect SYN flood after ... missing TCP ACKs — защитный экран проверяет пропущенные сегменты с флагами ACK).

Тут следует отметить, что при работе пользователей с внутренними серверами через защитный экран по некачественным каналам связи зачастую происходит регулярная потеря пакетов, что приводит к завершению незаконченного соединения TCP по тайм-ауту. Это значит, что только относительно большое значение этого параметра будет служить признаком атаки. При использовании последнего параметра (Reset count every ... seconds) факт атаки подтверждается только в том случае, когда указанное количество пропущенных подтверждений TCP ACK фиксируется в течение заданного промежутка времени.

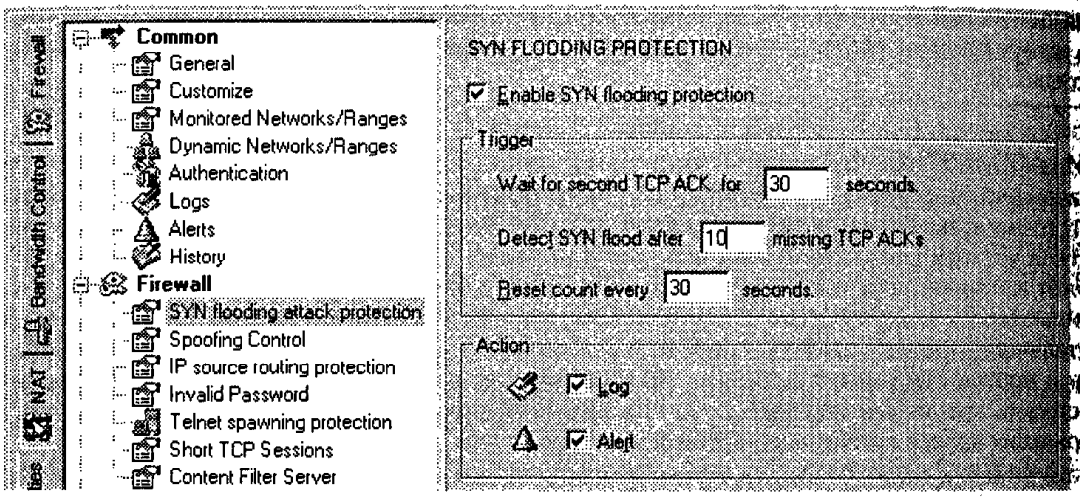


Рис. 5.4. Фрагмент окна настройки защиты от атаки TCP SYN Flooding

В нашем примере защитный экран будет оповещать об атаке, если он в течение 30 секунд обнаружит 10 пропущенных подтверждений TCP ACK от клиента, необходимых для завершения квитирования соединения. Дополнительно защитный экран будет принудительно разрывать соединение, если от клиента не поступает третий, завершающий пакет TCP ACK. Эти значения устанавливаются по умолчанию, когда администратор устанавливает флажок Enable SYN flooding protection. Однако администратору может потребоваться скорректировать данные параметры, если к нему поступает большое количество оповещений об атаках TCP SYN flooding, ввиду того что, скорее всего, это будет «ложная тревога».

Администратор сети может настроить защитный экран на отправку по почте извещений об атаках. В Guardian Pro этот механизм действует по некой накопительной схеме. При этом защитный экран посылает обобщающие сообщения, например, по конкретному адресу электронной почты, не информируя о каждом обнаружении атаки и не нервируя администратора. На рис. 5.5 показано реальное письмо с оповещением об атаке.

К сожалению, без дополнительных средств точно отличить факт проведения атаки от результата работы плохих каналов связи затруднительно. По этой причине, чтобы своими поспешными выводами не обидеть инициаторов этих сообщений, будем считать, что их причина кроется в некорректной настройке параметров для отслеживания атак TCP SYN flooding (кроме того, стоит принимать

во внимание тот факт, что чаще всего при атаке в качестве адресов отправителей указываются либо случайные, либо несуществующие адреса). В этом сообщении также фигурирует адрес атакуемого компьютера (192.168.3.250). Обратите внимание на то, что в сети используется внутренняя адресная схема (192.168.3.0/24), а взаимодействие с внешним миром осуществляется благодаря технологии трансляции адресов NAT. Но логика работы защитного экрана такова, что вначале он преобразует внешний адрес поступающего пакета во внутренний адрес сервера, и только после этого применяет к нему правила фильтрации и проводит анализ на предмет наличия атак. В этом причина того, что в уведомлении указывается внутренний адрес сервера, несмотря на то что инициатором были хосты в Интернете.

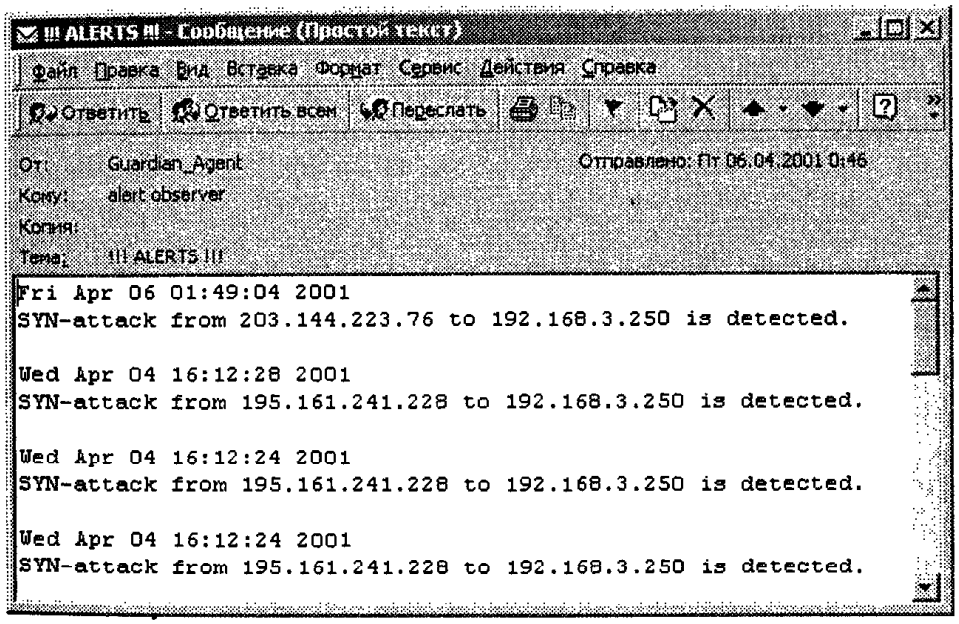


Рис. 5.5. Оповещение об атаке

ИЗ ОПЫТА

Вообще говоря, администратору необходим достаточно богатый опыт для уверенной работы со средствами предотвращения атак, и относиться к этой проблеме он должен спокойно. Но и умалять проблему нельзя. Как вы думаете, сколько подобных сообщений получает автор при работе реальной сети? Около 10 за рабочий день. Причем, учитывая накопительных характер оповещений, можно с уверенностью говорить о том, что в среднем сеть подвергается 20–25 атакам за сутки. Естественно, как уже не раз отмечалось, очень велика вероятность того, что это просто ложные срабатывания. С другой стороны, нельзя не учитывать и достаточную вероятность осуществления реальных атак, что просто подтверждает их популярность и необходимость для администратора сети обращать на них самое пристальное внимание. Автор полностью согласен с экспертами в области компьютерной безопасности (в частности, с авторами третьего издания книги «Атака из Internet»), говорящими о том, что большая часть атак мотивирована элементарным вандализмом, а не преследует цель получения доступа к какой-либо конфиденциальной информации. Иначе довольно трудно объяснить повторяющиеся попытки вывести из строя сервер WWW (а именно этот сервис предоставляет компьютер с адресом 192.168.3.250), так как сам сервер общедоступен.

Блок управления передачей протокола TCP

Одновременное взаимодействие нескольких прикладных программам на одно компьютере с помощью протокола TCP сводится к взаимодействию через сок ты. Для сохранения всей совокупности информации, необходимой для создани и поддержки соединения, каждый раз при установлении соединения создае ся структура данных, называемая блоком управления передачей (Transmissio Control Block — TCB). TCB содержит ряд переменных, определяющих оч редность отправления и получения сегментов. В частности, к ним относятс SND.UNA — посылка не подтверждена; SND.NXT — послать следующий сегмен SND.WND — отправить окно; RCV.NXT — получить следующий сегмент. На рис. 5 показана последовательность этапов отправки и приема данных.

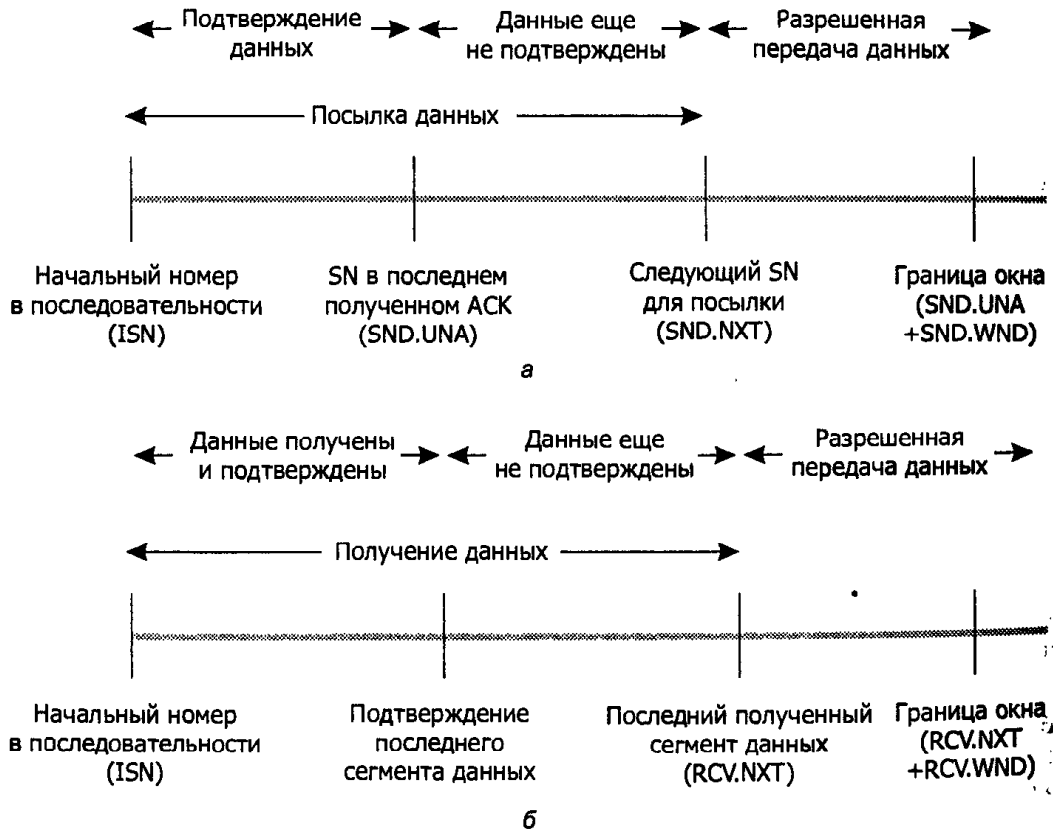


Рис. 5.6. Чередование отправки и приема данных: а — этапы отправки данных, б — этапы приема данных

Рассмотрим пример использования некоторых переменных. Отправитель с по мощью переменной SND.NXT следит за отправкой следующего сегмента в очереди Получатель на основании значения переменной RCV.NXT отслеживает номер сл дующего поступающего сегмента. В переменную SND.UNA отправитель помеща последний номер сегмента, который уже отправлен, но еще не подтвержд Когда отправитель посылает новый сегмент, он инкрементирует значение сво переменной SND.NXT. Получатель при поступлении этого сегмента увеличив

значение своей переменной RCV.NXT и отправляет подтверждение. При доставке подтверждения увеличивается значение переменной SND.UNA отправителя. Разность значений переменных SND.NXT и SND.UNA может служить мерой задержки сегментов в сети. Значения переменных изменяются на величину, равную длине поля данных сегмента. Более подробно формат заголовка, описание полей, флаги, возможные состояния соединения и переменные описаны в RFC 793.

Скользящее окно

Как и большинство протоколов, осуществляющих управление потоком данных (например, HDLC и X.25), протокол TCP использует механизм скользящих (или плавающих) окон. Пожалуй, для TCP можно считать этот механизм основополагающим. Протоколы HDLC и X.25 реализуют его в классическом виде — на каждый отправленный блок данных должно быть получено подтверждение. Протокол TCP несколько отходит от традиционной схемы. Основной недостаток классического варианта заключается в том, что за один сеанс может быть передан только один сегмент. В данном случае под сеансом понимается посылка сегмента и получение подтверждения его успешного приема. Такая схема не позволяет работать с максимальной производительностью. Эффективность может быть значительно повышена, если разрешить передачу множества сегментов за один сеанс с группировкой всех подтверждений для них в одно.

Рассмотрим этот метод на примере работы двух станций — А и Б, которым необходимо обмениваться данными. Станция Б выделяет буферное пространство для приема n сегментов. То есть можно сказать, что в данный момент времени станция Б готова принять n сегментов, а станция А готова послать те же n сегментов без подтверждения их приема. Для идентификации подтверждений о принятых сегментах каждый из них помечается своим номером в последовательности данных. Станция Б удостоверяет получение текущего сегмента посылкой подтверждения, которое включает в себя номер следующего ожидаемого сегмента в последовательности. Это подтверждение также косвенно извещает станцию А о том, что станция Б готова получить следующие n сегментов начиная с указанного номера последовательности. Очевидно, что можно выслать одно подтверждение на несколько сегментов. Например, станция Б получила сегменты 2, 3 и 4, но задержала подтверждения на сегменты 2 и 3 до получения сегмента 4. После посылки подтверждения с номером 5 станция Б информирует о получении сегментов 2, 3 и 4 за один раз. Иными словами, станция А поддерживает список номеров в последовательности, которые ей разрешено посылать, а станция Б — список номеров в последовательности, которые она готова принять. Каждый из этих списков может быть рассмотрен как окно сегментов. Такую схему работы часто называют «управление потоком с помощью скользящего окна».

Так как номер в последовательности занимает в сегменте одно поле, то очевидно, что он имеет границы. Например, если это поле занимает 3 бита, то номер в последовательности может принимать значения от 0 до 7. Соответственно, сегменты нумеруются циклически, по модулю 8, то есть за номером 7 последует номер 0. Таким образом, для поля номера последовательности, состоящего из

k битов, значения номера лежат в пределах от 0 до $(2^k - 1)$, а сегменты нумеруются по модулю 2^k. В процессе передачи и приема сегменты проходят ряд промежуточных состояний (рис. 5.7).

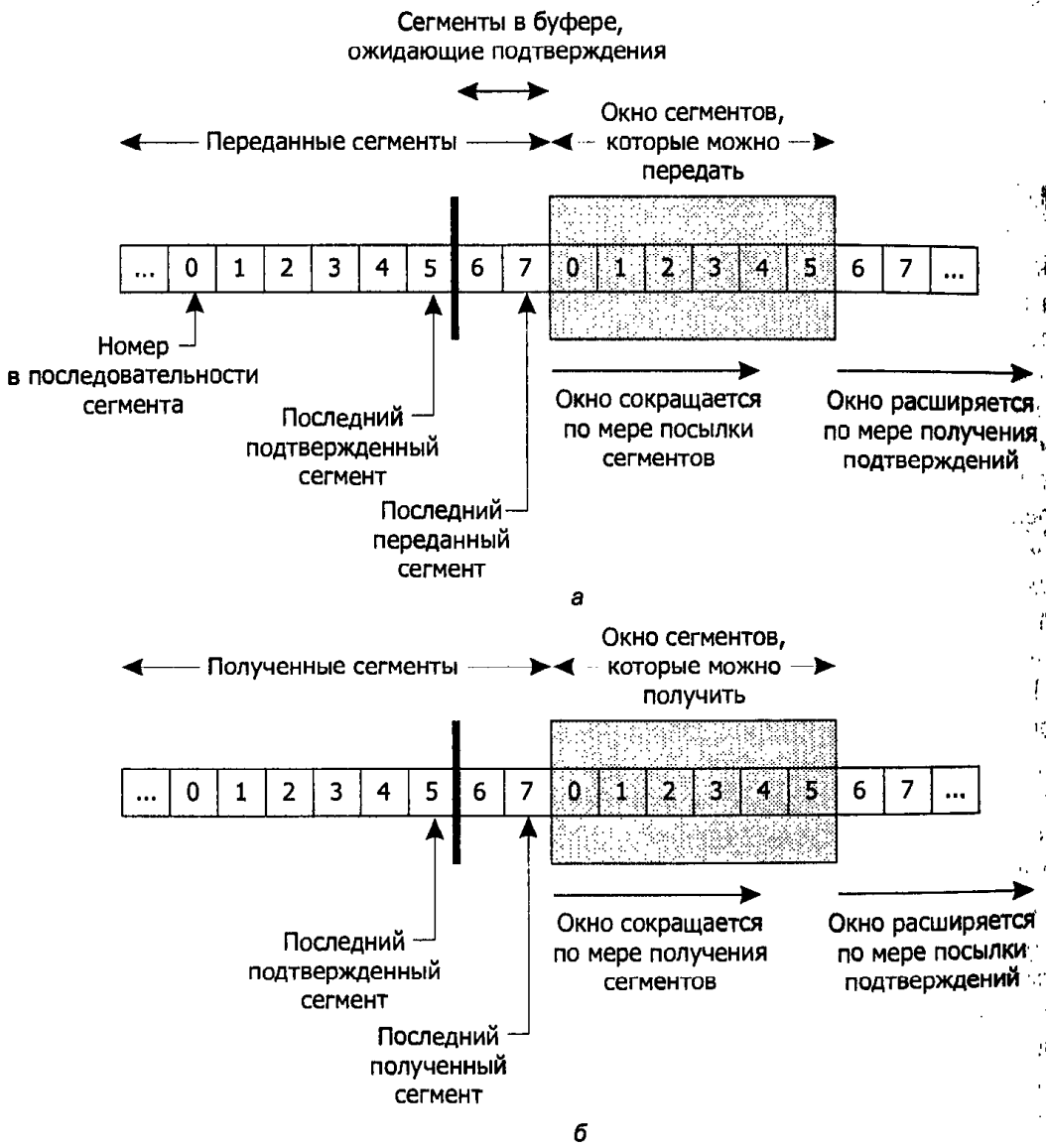


Рис. 5.7. Состояния сегментов при приеме и передаче: а — на стороне отправителя, б — на стороне получателя

На рис. 5.7 показаны направления движения границ окон посылки и приема. В примере для простоты используется 3-разрядное поле «Номер в последовательности». Затененные прямоугольники указывают на сегменты, которые могут быть посланы. В соответствии с рисунком, отправитель может послать 5 сегментов, начиная с сегмента с номером 0. Тогда, когда очередной сегмент отправляется в путь, ширина затененной области уменьшается; каждый раз, когда получено подтверждение, ширина затененной области увеличивается. Сегменты, находящиеся между вертикальной чертой и затененным участком, уже были посланы

но еще не были подтверждены. Отправитель должен хранить копии этих сегментов в своем буфере на тот случай, если потребуется их повторная передача.

Рассмотрим следующий пример (рис. 5.8), который позволяет проследить за обменом информацией.

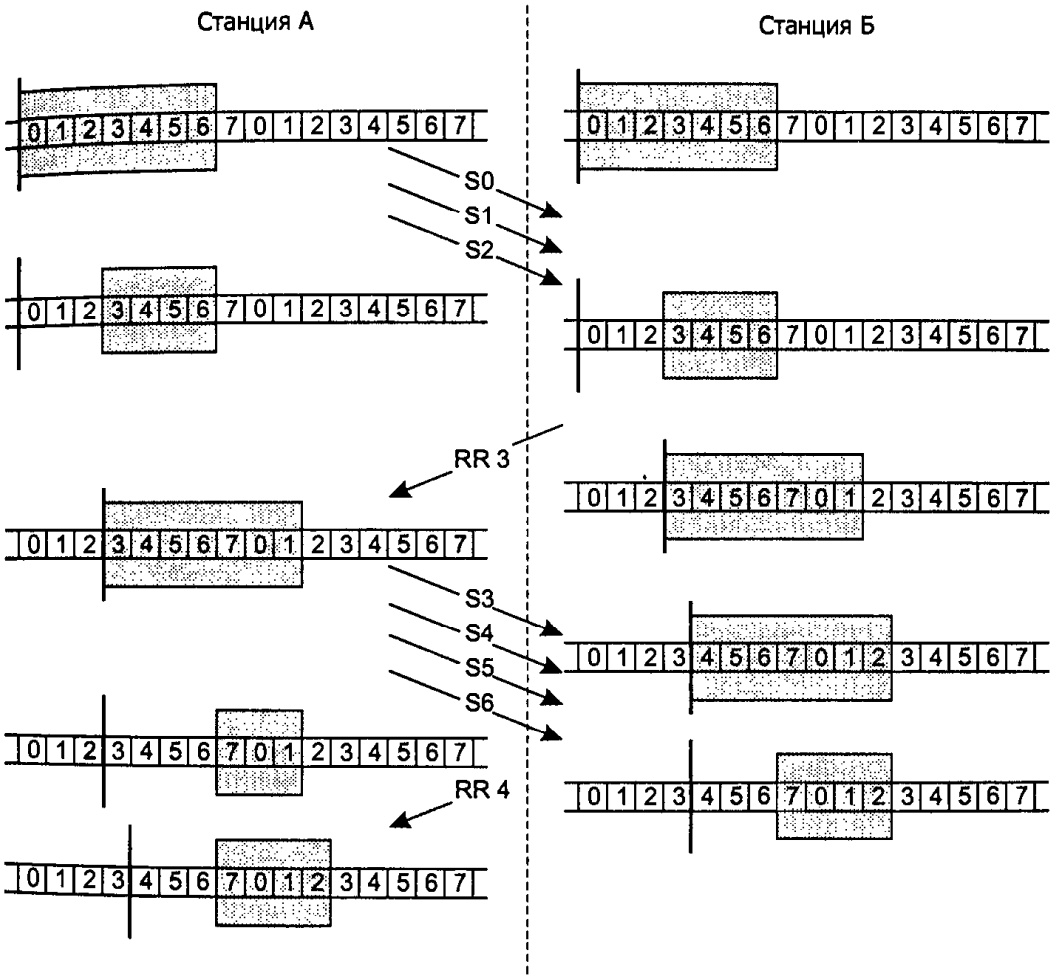


Рис. 5.8. Схема обмена информацией

Здесь также используется 3-разрядное поле номера в последовательности. Максимальный размер окна в этом случае позволяет работать с 7 сегментами.

В начале работы размер окон станций А и Б таков, что станция А может передать 7 сегментов, начиная с сегмента S0, а станция Б — принять такое же количество сегментов. После доставки 3 сегментов (S0, S1 и S2) без подтверждения станция А сокращает размер своего окна посылки до 4 сегментов и сохраняет в буферной памяти копию 3 посланных сегментов. Новый размер окна посылки указывает на то, что станция А готова передать 4 сегмента, начиная с сегмента S3. Станция Б после получения части сегментов отправляет сообщение RR3, из которого следует: «Я получила все сегменты по S2 включительно и готова принять сегмент S3; более того, я готова к приему 7 сегментов, начиная с сегмента S3». Для станции А эта информация означает то, что она получила разрешение на

передачу 7 сегментов, начиная с сегмента S3. Кроме того, станция А вправе, о стить свою буферную память от копий первых 3 сегментов, поскольку они успе но доставлены. Теперь станция А передает сегменты S3, S4, S5 и S6. Станци отправляет в ответ подтверждение RR4 на получение сегмента S3 и разреш отослать сегменты от S4 до S2 (сегмент S2 относится уже к следующей последо тельности из 7 сегментов). Но на момент получения станцией А этого подтвер дения сегменты S4, S5 и S6 уже были посланы и, следовательно, она может р ширить свое окно посылки и отправить 4 сегмента, начиная с сегмента S7.

Узкие места в сети

В протоколах канального уровня управление потоком данных с использовани плавающего окна дает получателю возможность управлять скоростью рабо отправителя. Получатель в этом случае только подтверждает полученные сегме ты и расширяет свое окно приема в соответствии с наличием свободного буфер го пространства. Задавать скорость передачи данных можно и при работе с п токолом TCP. Для простоты рассмотрим классический вариант, когда скоро отправки данных определяется скоростью поступления подтверждения для п дыдущего посланного сегмента.

Покажем, что в этом случае скорость поступления подтверждений зависит наличия так называемых «узких мест» в сети между отправителем и получат лем. Узким местом может быть либо устройство в сети либо часть канала, име щие значительно меньшие параметры быстродействия, чем вся сеть в цело В том числе вносить задержку способен и получатель. Узкое место, которое б вает как логическим, так и физическим, может возникнуть в разных точках эт сети на пути между отправителем и получателем. На рис. 5.9 представлен пр мер образования логических и физических узких мест.

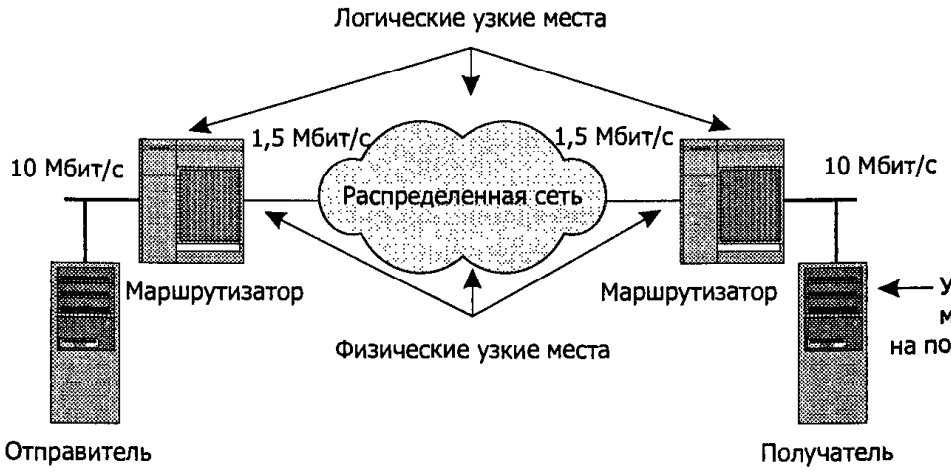


Рис. 5.9. Узкие места в сети

В этом примере отправитель имеет пропускную способность 10 Мбит/с. Но работы протокола TCP канал со скоростью 1,5 Мбит/с между маршрутиза

ми становится узким местом. Это физическое узкое место, и после достижения устойчивого состояния протокол TCP будет эффективно использовать доступную скорость передачи. Однако наиболее часто узкие места являются логическими и образуются из-за очередей на маршрутизаторах, коммутаторах или получателях. Задержки в очередях, как правило, подвержены флуктуациям и усложняют процесс формирования устойчивого потока.

Флуктуация задержек, которая имеет место в распределенных IP-сетях, затрудняет выбор политики отправки данных по протоколу TCP на стороне отправителя. Если поток имеет слишком маленькую скорость, то распределенная сеть будет использоваться неэффективно. Если один или несколько отправителей работают на повышенной скорости, то другие потоки TCP будут «стеснены» потоками этих отправителей. Если же множество отправителей TCP используют чрезмерно высокую скорость, то сегменты будут теряться при передачах, приводя к повторным передачам, или подтверждения будут сильно задерживаться, что также влечет ненужные повторные передачи. Более того, повторные передачи могут иметь эффект положительный обратной связи: чем больше сегментов посылается заново, тем быстрее растет перегрузка, что, в свою очередь, приводит к дальнейшему повышению задержек и к увеличению числа отброшенных сегментов. Все это имеет следствием увеличение числа повторных передач, что в еще большей степени усугубляет перегрузку.

Рассмотрим случай, когда узкое место находится где-то внутри сети. На рис. 5.10 изображен низкоскоростной канал между отправителем и получателем.

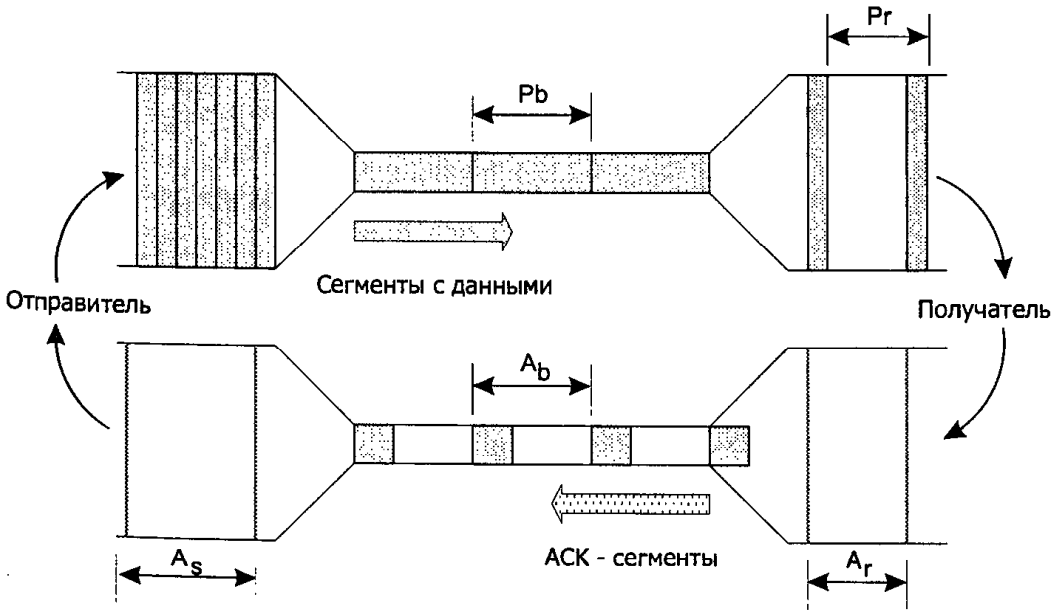


Рис. 5.10. Влияние низкоскоростного канала на передачу данных

Ширина этого канала пропорциональна скорости передачи данных. Отправитель и получатель подключены к высокоскоростным сетям. Канал из-за своей низкой пропускной способности изображен на рисунке тонким. Он как раз и соз-

дает «бутылочное горлышко». Каждый посланный сегмент с данными изоб
жается на этом рисунке прямоугольником, площадь которого пропорциональ
количеству байтов информации в сегменте. Поэтому когда сегмент проходит
тонкому каналу, он как бы сдавливается им и вытягивается в длину и, соотв
ственно, увеличивается время его прохождения по каналу. Время прохожде
сегмента через определенную точку медленного канала обозначим *Pb* — это в
мя между моментами пересечений передней и задней границ сегмента. Предп
ложим, что, продвигаясь по медленному каналу, сегменты идут вплотную др
к другу, то есть задняя граница первого примыкает к передней границе втор
сегмента. Поэтому *Pb* будет определять и время пересечения передней грани
первого сегмента и передней границы второго сегмента через некоторую точ
в канале. При поступлении сегментов в высокоскоростной канал это время ме
передними границами сегментов сохраняется даже с учетом повышения скоро
передачи данных, так как интервал между поступлениями не меняется. А
скольку сегмент на высокоскоростном канале как бы сужается, но увеличива
ся в ширину, то теперь это время состоит из времени прохождения через
сегмент и паузы до передней границы следующего сегмента, то есть *Pr* =
Если получатель подтверждает сегменты в момент поступления, то время ме
ду посылками подтверждающих сегментов (ACK-сегментов) *Ar* на выходе по
чателя равно *Pr*. И, наконец, мы получаем, что *Ab* = *Ar*, а *As* = *Ab*.

После попытки работать с высоким быстродействием система переходит в усто
чивое состояние, в котором скорость отправки сегментов равна скорости по
ступления подтверждений. Очевидно, что скорость отправки сегментов буд
зависеть от пропускной способности самого медленного канала. Можно ск
зать, что протокол TCP автоматически определяет узкое место в сети и рег
лирует свой поток. Этот процесс называется самосинхронизацией (self-clockin
В общем, наилучший способ повышения производительности — устранени
узких мест в сети.

Пропускная способность TCP

Рассмотрим методы определения максимально возможной пропускной спосо
ности в соединении протокола TCP. Пропускная способность зависит от разм
ра окна отправки, задержки и скорости передачи данных. Используем следу
щие обозначения:

W — размер окна отправки в байтах;

R — скорость передачи данных (бит/с) по определенному соединению, до
тупная на стороне отправителя;

D — задержка (в секундах) при передаче данных между отправителем и по
лучателем через определенное соединение.

Предположим, что отправитель начинает передавать последовательность байт
получателю через установленное соединение. Для того чтобы первый байт дос
получателя, потребуется время, равное *D* секундам. Такое же время — *D* секун
понадобится для получения подтверждения. В течение этого промежутка от

витель может передать *2RD* битов, или *RD/4* байтов. Но отправитель ограничен размером окна в *W* байтов и не может сдвигать окно, пока не получит подтверждение. Только при $W \geq RD/4$ на этом соединении достигается максимально возможная пропускная способность. Если $W < RD/4$, то близость пропускной способности к максимальной определяется отношением *W* к *RD/4*. Следовательно, нормированная пропускная способность *S* может быть выражена как:

$$S = \begin{cases} 1, \text{ если } W \geq RD/4 \\ \dfrac{4W}{RD}, \text{ если } W < RD/4. \end{cases}$$

. На рис. 5.11 представлен пример определения максимальной пропускной способности в зависимости от произведения *RD*.

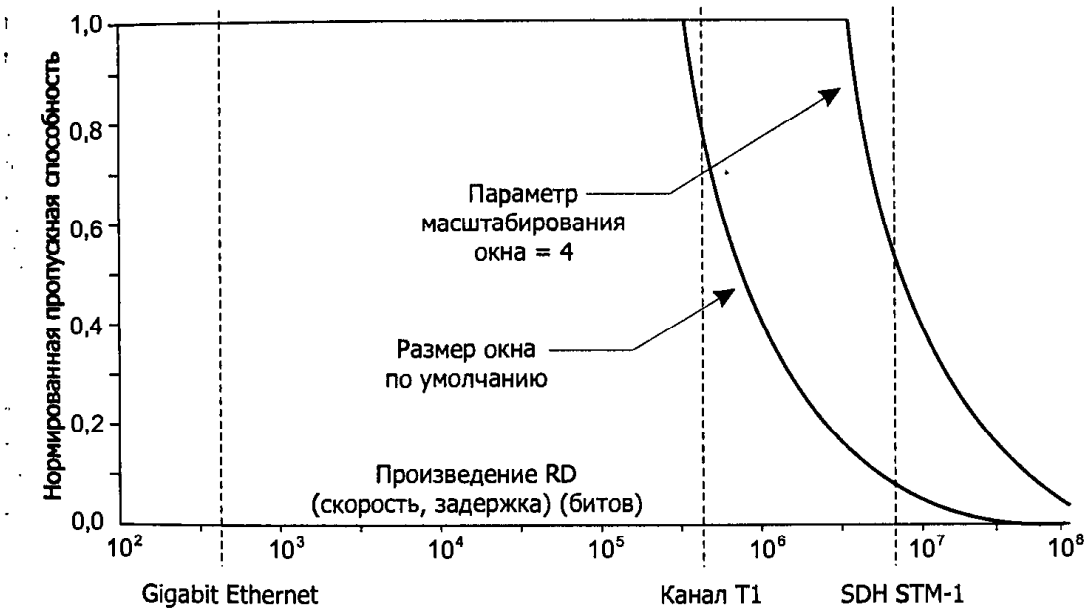


Рис. 5.11. Определение максимальной пропускной способности

Максимальный размер окна может составлять $2^{16} - 1 = 65\,535$ байт (поле «Окно» (Window) имеет длину 16 бит). Такой ширины окна достаточно для большинства приложений. В качестве примера рассмотрим три различные технологии, применяемые для передачи сегментов TCP и использующие означенный размер окна. Для технологии Gigabit Ethernet с длиной магистрали, равной 100 м, произведение *RD* будет меньше, чем 10^3 бит. На больших расстояниях, пусть даже при более низких скоростях, например в случае использования канала T1 (1,5 Мбит/с), произведение *RD* увеличивается, следовательно, эффективность падает. Тем не менее она остается приемлемой — около 0,8. На значительных дистанциях при дальнейшем увеличении скорости (допустим, речь идет о передаче информации с использованием технологии SDH — канал 155 Мбит/с между двумя городами) рассматриваемая технология становится крайне неэффективной — нормированная пропускная способность падает до 0,1. Очевидно, что

в данном случае размер окна слишком мал. Необходимо другое значение раметра масштабирования окна, которое позволило бы эффективно задейс вать пропускную способность канала. Достаточно увеличить этот параметр на 4 единицы, и окно будет значительно увеличено — $2^{20} - 1 \approx 10^6$ байт.

Рассматривая взаимное влияние параметров, можно получить представл о потенциале протокола TCP по повышению производительности. Однако суш вует множество усложняющих факторов, которые следует принять во внима Во-первых, в большинстве случаев некоторое количество соединений TCP мул плексируются через один канал, так что каждое соединение получает только ч доступной пропускной способности канала. Это приводит к снижению скор передачи R и, следовательно, к снижению эффективности работы прото Во-вторых, множество соединений TCP будет проходить через определенное к чество маршрутизаторов. Тогда время D будет равно сумме задержек в ка сети плюс сумма задержек на каждом маршрутизаторе в отдельности. При суммарная задержка на маршрутизаторах часто бывает наибольшей составляю общего времени D, особенно в случаях возникновения перегрузок. В-трет значение скорости R, используемое в приведенной выше формуле, опреде скорость передачи данных, доступную для этого соединения, только на сто отправителя. Если эта скорость передачи данных больше, чем скорость на од из переходов в пути от отправителя до получателя, то попытка передачи на ма мальной скорости приведет к образованию узкого места, что неизбежно увел время D. И, наконец, в-четвертых, если любой сегмент будет потерян, то он жен быть передан вновь, откуда вытекает снижение пропускной способно Влияние потерянных сегментов определяется политикой повторных переда

Для повышения производительности в сетях, для которых характерны б шие задержки, в операционную систему Microsoft Windows 2000 введен а ритм масштабирования окна (TCP Receive Window Size Calculation and Win Scaling). Этот алгоритм описан в разделе 2.2 документа RFC 1323. При испол вании параметра масштабирования окна количество байтов данных, определяем полем «Окно», становится кратным 2^F, где F — значение параметра масшта рования окна. Максимальная величина F, которая используется протоколом T равна 14. Следует отметить, что этот алгоритм включается только на время и циирования соединения.

В операционной системе Microsoft Windows NT размер окна приема мо возрастать на величину, кратную максимальному размеру сегмента (Maxim Segment Size — MSS). Размер MSS определяется во время установления сое нения. По умолчанию окно приема имеет размер 8 Кбайт для Windows NT и 16 Кбайт для Windows 2000. Ширина окна устанавливается в реестре опе ционной системы параметром TcpWindowSize (табл. 5.10).

Таблица 5.10. Параметр реестра TcpWindowSize

Название параметра	TcpWindowSize
Ключ в реестре	Tcpip\Parameters, Tcpip\Parameters\Interface\
Тип записи	REG_DWORD
Диапазон значений	0–0x3FFFFFFF (1 073 741 823)

В сетях Ethernet размер окна обычно равен 8760 байт (8192 байт, размещенные в 6 сегментах по 1460 байт каждый) для Microsoft Windows NT 4.0 и 17 520 байт (16 Кбайт, размещенные в 12 сегментах по 1460 байт) для Microsoft Windows 2000.

Для операционной системы Microsoft Windows 2000 ширина окна рассчитывается следующим образом: первый запрос на установление соединения, посылаемый удаленному абоненту, регламентирует значение 16 Кбайт (16 384 байт) данных. После установления соединения размер окна приема округляется до объема данных, кратного максимальному размеру сегмента (MSS), который был оговорен в процессе установления соединения. Если размер окна приема определяет объем данных, близкий к четырехкратному значению MSS, то окно выравнивается до значения 4×MSS. Этот размер окна сохраняется до тех пор, пока не будет включен алгоритм масштабирования окна.

Анализируя результаты перехвата информации в процессе установления соединения протокола TCP при включенной поддержке алгоритма масштабирования окна, следует иметь в виду, что рекламируемые размеры окон должны быть кратны параметру масштабирования. Этот параметр появляется только в первых двух сегментах при установлении соединения. Например, для размера окна, равного 65 535, при параметре масштабирования 3 актуальный размер окна будет составлять 524 280 (или $2^3 \times 65\,535$). В приведенном ниже примере показан фрагмент перехвата сегмента, который содержит параметры, относящиеся к работе алгоритма масштабирования окна.

```
...
+ TCP: Flags = 0x02 : ....S.
         TCP: Window = 65535 (0xFFFF)
         TCP: Checksum = 0x8565
         TCP: Urgent Pointer = 0 (0x0)
         TCP: Options
         +  TCP: Maximum Segment Size Option
            TCP: Option Nop = 1 (0x1)
            TCP: Window Scale Option
                  TCP: Option Type = Window Scale
                  TCP: Option Length = 3 (0x3)
                  TCP: Window Scale = 5 (0x5)
            TCP: Option Nop = 1 (0x1)
            TCP: Option Nop = 1 (0x1)
         +  TCP: Timestamps Option
            TCP: Option Nop = 1 (0x1)
            TCP: Option Nop = 1 (0x1)
         +  TCP: SACK Permitted Option
...
```

В операционной системе Microsoft Windows 2000 окно масштабируется автоматически, если параметр реестра TcpWindowSize установлен в значение, превышающее 64 Кбайт. Масштабирование окна можно запретить вручную параметром Tcp1323Opts (табл. 5.11).

Таблица 5.11. Параметр реестра Tcp1323Opts

Название параметра	Tcp1323Opts
Ключ в реестре	Tcpip\Parameters
Тип записи	REG_DWORD
Возможные значения	0, 1, 2, 3
Значение по умолчанию	3

Работать с окном, превышающим 64 Кбайт, можно только с абонентом, ко рый поддерживает эту опцию. Значение по умолчанию устанавливается как н меньшее из следующих: 0xFFFF, значение дополнительного параметра реес GlobalMaxTcpWindowSize, максимум из 4×MSS и числа 16 384, выравненного значения, кратного размеру поля данных протокола TCP. Значение по умолча для сети Ethernet составляет 17 520 байт (в реализации TCP для операционно системы Microsoft Windows 2000), но может быть немного сокращено, если единение установлено с абонентом, который поддерживает алгоритмы SA и временные отметки, так как они увеличивают размер заголовка протокола T сверх обычного (20 байт), оставляя меньше места для данных.

Следует отметить, что данный параметр является одновременно и глоба ным параметром. Он устанавливается на каждом интерфейсе в отдельности в висимости от того, где расположен ключ реестра. Если это значение для опред ленного интерфейса, то оно перекрывает значение для всей системы.

Параметр Tcp1323Opts может иметь следующие значения:

- 0 — запретить использование опций RFC 1323,
- 1 — разрешено только масштабирование окна;
- 2 — разрешено использовать только временные штампы;
- 3 — разрешено использовать обе опции.

Изменение размера окна на маршрутизаторе

По умолчанию размер окна протокола TCP составляет 2144 байт. В докумен тации на операционную систему Cisco IOS дана рекомендация не менять эт значение, до тех пор пока не будет выяснено, что маршрутизатор посылает пак ты данных, размер которых превышает 536 байт. Для изменения размера ок следует воспользоваться командой ip tcp window-size, выполняемой в глобальн режиме, а в качестве параметра указать новый размер окна в байтах.

Управление потоком

Управление потоком основано на механизме плавающего окна, но кроме это использует более гибкую схему приема и передачи данных и отправки подтвер дений успешного их приема. Управление потоком осуществляется в соответств со схемой, устанавливающей предельную норму выделения готового к отпра

массива данных. Согласно этой схеме, каждый индивидуальный передаваемый байт имеет свой собственный номер в последовательности. Когда протокол TCP посылает сегмент, он выставляет в поле номера номер первого байта в последовательности байтов, размещаемых в поле данных этого сегмента. На стороне получателя поступивший сегмент подтверждается сообщением, имеющим форму ($A = i$, $W = j$). Такая запись обозначает следующее: подтверждается получение всех байтов, вплоть до номера ($i - 1$); следующие ожидаемые байты для приема имеют номер в последовательности i. Кроме того, выдается разрешение на посылку дополнительного окна W (Window — окно) из j данных, где байты данных соответствуют номерам в последовательности от i вплоть до $i + j - 1$. Рисунок 5.12 иллюстрирует работу этого механизма.

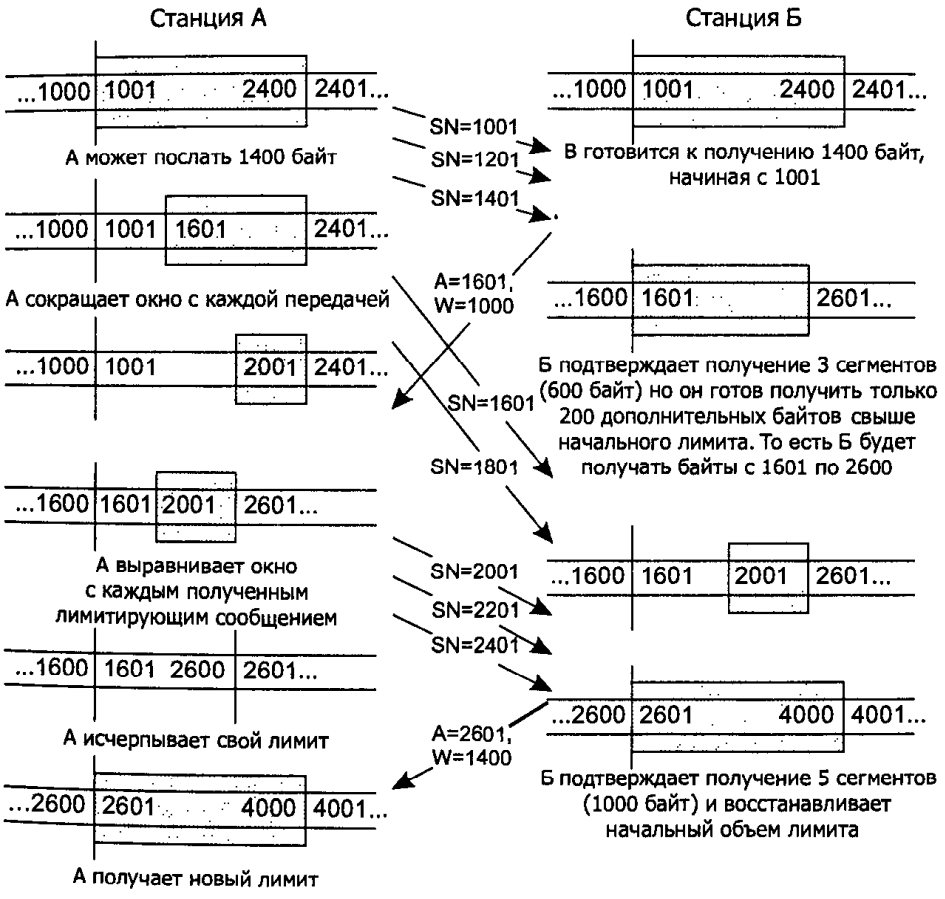


Рис. 5.12. Схема управления потоком

Для большей наглядности покажем поток данных, идущий только в одном направлении, и предположим, что в каждом сегменте посылается 200 байт данных. Во время установления соединения номера в последовательностях отправителя и получателя синхронизированы и станция А работает, исходя из начального лимита, равного 1400 байт, начиная с номера байта 1001. После посылки 600 байт в трех сегментах станция А уменьшает свое окно отправки до размера 800 байт (номера от 1601 до 2400). После получения этого сегмента станция Б подтвер-

ждает доставку всех байтов вплоть до 1601-го байта и формирует свое со венное окно приема на 1000 байт. Это означает, что станция А может посыл байты начиная с номера 1601 по номер 2600, то есть в сумме 5 сегментов. Од ко к моменту времени, когда сообщение от станции Б доходит до станции станция А уже успевает выслать два сегмента, содержащие байты с номера от 1601 до 2000, что было разрешено во время начального назначения п дельной нормы. Следовательно, оставшийся резерв станции А на этот мом составляет всего 400 байт или два сегмента. В процессе обмена станция А, двигает конечную (левую) границу своего окна каждый раз, когда перед сегмент, и смещает начальную (правую) границу окна только тогда, когда по чает новую предельную норму.

На практике обе стороны одновременно задействуют режимы передачи и пр ема, так как данные могут передаваться в обоих направлениях (режим полно плексной передачи). Механизм выделения лимита достаточно гибок. Наприм рассмотрим ситуацию, когда заключительное сообщение, посланное станцией было: $A = i$, $W = j$. Последним байтом данных, полученным станцией Б, с байт с номером $i - 1$. В результате для увеличения лимита до значения k, е выполняется условие $k > j$ и еще не поступили дополнительные данные, ста ция Б формирует сообщение ($A = i$, $W = k$), а для подтверждения входящ сегмента, содержащего m байтов данных ($m < j$), без получения дополнительн го лимита станция Б формирует сообщение ($A = i + m$, $W = j - m$).

Следует отметить, что от получателя не требуется немедленного подтверж ния приходящих сегментов — он может ожидать какое-то время, формируя об щенное подтверждение для некоторого итогового их количества. Получате должен подчиняться определенной политике, регулирующей объем входны данных, который он сам позволяет передавать отправителю. Можно выдели две схемы поведения получателя: консервативную и оптимистическую. *Консе вативная* схема управления потоком основана на том, что предельная норма в деляется исходя из фактического значения свободного буферного пространст Если это правило применить к ситуации, показанной на рис. 5.12, то первое л митирующее сообщение говорит о том, что станция Б готова разместить в свое буфере 1000 байт, а второе — о том, что станция Б в состоянии размести 1400 байт. Консервативная схема управления потоком может ограничить проп скную способность логического соединения в ситуациях, когда в сети возни ют длительные задержки.

Получатель может увеличивать пропускную способность с помощью *опт мистического* выделения лимита на свободное буферное пространство, которо он на данный момент не имеет. Например, если буфер получателя заполнен, он ожидает, что сумеет освободить 1000 байт буферного пространства за вре прохождения информации между конечными точками соединения, то он жет немедленно выделить лимит в 1000 байт. Если получатель в силах п держивать скорость, заданную отправителем, то такая схема способна повыси пропускную способность и не принесет вреда. Если же отправитель рабо быстрее получателя, то некоторые сегменты будут отбрасываться из-за за тости буфера, что повлечет за собой повторную передачу. А так как пов ная передача является необходимым атрибутом надежного сетевого сер

то оптимистическое управление потоком при таком развитии событий может повысить загрузку сети.

Получатель всегда старается увеличить окно приема, если имеет свободное буферное пространство. Отправитель также стремится расширить окно отправки при малейшей возможности и на любую величину. В такой ситуации говорить о стабильном потоке сегментов не приходится. Для того чтобы уберечь отправителя и получателя от соблазна «втиснуть» в канал лишний сегмент, используется алгоритм предотвращения «синдрома глупого окна» (Silly Window Syndrome — SWS). Этот алгоритм описан в RFC 1122. Большой объем данных не отправляется до тех пор, пока получатель не объявит размер окна, достаточный для посылки полного сегмента, или не будет выполнена настройка, запрещающая увеличить окно приема меньше чем на сегмент.

Стратегии отправления и приема данных

При отсутствии данных, помеченных флагом PSH, протокол TCP на стороне отправителя имеет возможность самостоятельно назначать время передачи. Когда данные передаются модулю протокола TCP пользовательским приложением, они записываются в буфер для передачи. Протокол TCP может создавать сегмент для каждой группы поступивших от приложения данных или ожидать накопления большого объема данных, и только после этого формировать и отправлять сегмент. Актуальность политики отправления всецело зависит от требований к производительности. Если передачи сегментов происходят редко, но при этом переносятся большие объемы данных, то такая схема вносит небольшие накладные расходы при создании сегмента и его обработке. Если же передачи происходят часто, а объем передаваемых данных невелик, то такая система обеспечивает быструю реакцию на изменения состояния сети.

При отсутствии данных, помеченных флагом PSH, протокол TCP на стороне получателя также может самостоятельно определять время доставки данных пользовательскому приложению. Так, он может доставлять данные последовательно при получении каждого сегмента или заносить данные из нескольких сегментов в буфер приема. Как и в предыдущем варианте, реальная политика доставки будет зависеть от требований к производительности. Если данные приходят редко и имеют большие объемы, то приложение не получит их сразу. Если данные поступают часто и малыми порциями, следует ожидать избыточной обработки информации в протоколе TCP и в приложении.

Когда сегменты поступают в порядке их отправки по установленному соединению, протокол TCP помещает данные в буфер получения для доставки пользовательскому приложению. Вполне возможна ситуация, когда определенный сегмент поступает не в порядке отправления. В этом случае протокол TCP на стороне получателя может либо принимать только сегменты, которые приходят в порядке отправления (другие сегменты просто отбрасываются), либо принимать все сегменты, номера которых зафиксированы в окне приема (вне зависимости от порядка получения).

Первый вариант действий упрощает реализацию протокола, но при возникают излишние накладные расходы на сеть, так как отправитель б ожидать некоторое время, а затем повторно передаст отброшенные сегме которые были успешно получены, но затем были отброшены из-за некоррек го порядка поступления. Более того, если один сегмент был потерян при пер че, все последующие сегменты необходимо посылать повторно. Во втором в анте может быть снижено количество повторных передач, но при этом требу более сложный алгоритм приема и более запутанная схема сохранения дан для буферизации и отслеживания порядка принятия данных.

Протокол TCP поддерживает очередь сегментов, которые были посланы еще не были подтверждены. Спецификация протокола говорит о том, что будет повторно передавать сегмент, на который не поступило подтвержде в определенный период времени. Реализация протокола TCP поддерживает стратегии повторной передачи.

- **Только первый**. Используется один таймер повторной передачи для очереди. Если получено подтверждение, удаляется первый сегмент из оч ди повторной передачи и сбрасывается таймер. Если время таймера исте повторно передается сегмент из начала очереди и таймер сбрасывается.
- **Пакетная**. Поддерживается один таймер повторной передачи для всей оч ди. Если получено подтверждение, удаляются все сегменты из очереди вторной передачи и сбрасывается таймер. Если время таймера истекает вторно передаются все сегменты из очереди и таймер сбрасывается.
- **Индивидуальная**. Поддерживается один таймер для каждого сегмента в реди. Если получено подтверждение, удаляется первый сегмент из оче повторной передачи и обнуляется соответствующий таймер. Если подо к концу время любого таймера, то повторно индивидуально передается с ветствующий сегмент и его таймер сбрасывается.

Первая стратегия повышает эффективность передачи трафика, так как то ко потерянный сегмент (или сегменты, подтверждения которых были пот ны) передаются повторно. Но из-за того что таймер для второго сегмента в реди не устанавливается до тех пор, пока не подтвержден первый сегмент, м возникать некоторые задержки. Индивидуальная стратегия решает эти блемы за счет более сложной реализации. Пакетный режим также снижает роятность длительных задержек, но может привести к ненужным повтор передачам.

Реальная эффективность политики повторной передачи зависит от по тики приема сегментов на стороне получателя. Если получатель реализует литику приема, в соответствии с которой принимаются только сегменты дующие в порядке отправления, то он будет отбрасывать сегменты, получе после потерянного сегмента. Такая схема работы хорошо подходит к на ной стратегии. Если получатель принимает все сегменты несмотря на п док их прибытия, то оптимальными являются стратегии «только первый индивидуальная.

На стороне получателя при поступлении сегмента протокол TCP может выбирать время отправки подтверждения двумя способами.

- **Немедленно**. Как только данные с определенным номером в последовательности приняты, немедленно передается пустой (без данных) сегмент, содержащий соответствующий номер подтверждения.
- **С накоплением**. Когда данные успешно приняты, отмечается необходимость в их подтверждении, но ожидается очередной выходящий сегмент с данными, в котором установлен флаг ACK. Во избежание длительных задержек устанавливается таймер окна. Если время таймера истекает до момента отправки очередного сегмента с подтверждением, то посылается пустой сегмент, содержащий соответствующий номер подтверждения.

Первый вариант прост в реализации и обеспечивает полную информированность протокола TCP на отправляющей стороне, что ограничивает ненужные повторные передачи. Однако такой подход приводит к дополнительным передачам пустых сегментов, использующихся только для подтверждения. Более того, он повышает загрузку сети. Вполне возможна, например, ситуация, когда протокол TCP на принимающей стороне получает сегмент и немедленно посылает подтверждение. А само приложение вдруг расширяет свое окно приема, вызывая тем самым посылку еще одного пустого сегмента для предоставления дополнительного лимита отправляющей стороне. Поскольку эта стратегия сопровождается накладными расходами, обычно используется режим работы с накоплением. Следует отметить, что последний приводит к увеличению загрузки на стороне получателя и усложняет задачу отправителя по оценке времени обращения.

Рассмотрим два расширения протокола TCP, реализованные в Window 2000.

Выборочное подтверждение SACK

Это расширение сравнительно недавно утвердила организация IETF (Internet Engineering Task Force — http://www.ietf.org) (RFC 2018). Оно позволяет подтверждать прием данных не в порядке их поступления, как это было раньше, а выборочно. Данный подход имеет два главных преимущества. Во-первых, повышается эффективность повторной передачи сегментов TCP благодаря сокращению времени на саму процедуру. Протокол TCP использует алгоритм повторной передачи, опираясь на информацию, извлекаемую из упорядоченных подтверждений. Такой подход вполне приемлем, однако в случае его применения на восстановление каждого потерянного сегмента уходит примерно один цикл обращения. SACK (Selective Acknowledgments — выборочное подтверждение) позволяет осуществлять повторную передачу сразу нескольких потерянных сегментов в одном цикле. Во-вторых, благодаря выборочным подтверждениям протокол TCP точнее оценивает доступную ширину полосы пропускания в условиях нескольких последовательных потерь сегментов и способен обойтись без алгоритма медленного старта. SACK играет важную роль для соединений, использующих большой размер окна TCP. Когда работает SACK (а этот режим включен по умолчанию), при потере сегмента или серии сегментов получатель имеет возможность

точно информировать отправителя о том, какие данные были получены успе
но, и указывает на «прореху» в потоке сегментов. Отправитель может теп
выборочно повторить передачу потерянных сегментов и не передавать весь бл
данных. Ниже показан пример работы механизма SACK.

```
...
        TCP: Source Port = 0x04DA
        TCP: Destination Port = NETBIOS Session Service
        TCP: Sequence Number = 925104 (0xE1DB0)
        TCP: Acknowledgement Number = 54857341 (0x3450E7D)
{Подтверждение получения данных до номера в последовательности 54857341}
...
     TCP: Options
          TCP: Option Nop = 1 (0x1)
          TCP: Option Nop = 1 (0x1)
        + TCP: Timestamps Option
          TCP: Option Nop = 1 (0x1)
          TCP: Option Nop = 1 (0x1)
          TCP: SACK Option
             TCP: Option Type = 0x05
             TCP: Option Length = 10 (0xA)
             TCP: Left Edge of Block  = 54858789 (0x3451425)
     TCP: Right Edge of Block = 54861685 (0x3451F75)
{Указание на успешно принятые номера в последовательности данных
   54858789-54861685}
...
```

В примере показан фрагмент перехвата сетевого монитора (Microsoft Netw
Monitor). Видно, что подтверждено получение данных до номера в последо
тельности 54857341, плюс получение данных с номерами в последователь
сти от 54858789 до 54861685. Данные с номерами от 54857341 до 54858789
подтверждены. Таким образом, SACK указывает на то, что данные с номера
от 54857341 до 54858789 были пропущены и отправителю следует повторить
передачу. В реестре Windows 2000 существует параметр SackOpts (табл. 5.1
с помощью которого можно активизировать работу выборочного подтвержден

Таблица 5.12. Параметр реестра SackOpts

Название параметра	SackOpts
Ключ в реестре	Tcpip\Parameters
Тип записи	REG_DWORD
Диапазон значений	0, 1 (False, True)
Значение по умолчанию	1 (True)

Операционная система Cisco IOS поддерживает механизм выборочного по
тверждения, который включается командой ip tcp selective-ack, выполняемой в гл
бальном режиме. Другой возможностью выборочного подтверждения являет

поддержка временных отметок TCP. Однако следует отметить, что при задействовании механизма сжатия заголовков TCP последние должны быть отключены. Это также касается и алгоритма SACK. Для включения механизма временных отметок следует воспользоваться командой ip tcp timestamp, выполняемой в глобальном режиме. Временные отметки и SACK на маршрутизаторах компании Cisco Systems по умолчанию неактивны (отключены).

Задержанное подтверждение

В соответствии с этим алгоритмом, подтверждения высылаются, если не посылалось подтверждение на предыдущий принятый сегмент и если после принятия сегмента в течение 200 мс не зарегистрирован следующий. Значение таймера отложенных подтверждений можно устанавливать в реестре посредством параметра TcpDelAckTicks (табл. 5.13), который был впервые введен в пакете обновлений 4 (Service Pack 4) для операционной системы Microsoft Windows NT 4.0.

Таблица 5.13. Параметр реестра TcpDelAckTicks

Название параметра	TcpDelAckTicks
Ключ в реестре	Tcpip\Parameters\Interfaces\<interface>
Тип записи	REG_DWORD
Диапазон значений	0–6
Значение по умолчанию	2 (200 миллисекунд)

Значение данного параметра может меняться от 0 до 6 с шагом 100 мс, то есть максимальное время таймера — 600 мс. Установка параметра в 0 запрещает задерживать подтверждения, что приводит к немедленной отправке подтверждений для каждого полученного сегмента.

Таймер повторной передачи

Протокол TCP не формирует явных негативных подтверждений, то есть подтверждений, явно указывающих на произошедшие нарушения. Вместо этого протокол полагается исключительно на положительные подтверждения и на повторную передачу в случаях, когда подтверждение не поступает в течение определенного интервала времени. К повторной передаче сегмента могут привести два события.

1. Сегмент испорчен при передаче, но тем не менее доставлен получателю. Контрольная сумма, включенная в заголовок, позволяет получателю обнаружить ошибку и отбросить такой сегмент.
2. Сегмент просто не поступает.

И в том и в другом случае посылающая сторона просто не знает о том, что посылка сегмента была неуспешной.

Если на принимающей стороне обнаруживаются ошибки при приеме мента либо он не принимается вовсе, то при этом не формируется и не п лается ACK (подтверждение). В этом случае потребуется повторная пере сегмента. Для принятия решения о времени повторной передачи исполь ся таймер, работающий с каждым посланным сегментом. Если время тай истекает до момента получения ACK для этого сегмента, отправитель дол выполнить повторную передачу. Важным параметром в протоколе TCP являе регулируемое время этого таймера. Если значение тайм-аута будет слиш малым, то часто будут осуществляться ненужные повторные передачи, уме шающие полосу пропускания сети. Если это значение будет чересчур бо шим, то работа протокола станет очень инертной при потере сегмента. Ну задать значение времени, немного большее, чем время обращения (RTT, Rou Trip Time).

Существуют два способа определения интервала повторной передачи. В п вом случае используется фиксированное значение, основанное на статисти ских данных, характерных для обычного поведения распределенной сети, есть значение, ненамного превышающее среднее значение RTT. Такая страте не способна адекватно и гибко реагировать на резкие изменения в условиях боты сети. Как уже отмечалось, если значение таймера слишком велико, раб протокола будет инертной, а если это значение мало, то может сложиться сит ция, при которой перегрузка в сети приведет к повторным передачам, что больше увеличит перегрузку.

Второй способ — адаптивная схема работы, которая также имеет достоинс и недостатки. Предположим, что протокол TCP постоянно отслеживает вре получения подтверждений на посланные сегменты и устанавливает свой тайм основываясь на наблюдаемой задержке. Это значение не может полностью уд летворять всю совокупность требований к транспортному протоколу по следу щим причинам: посылка сегмента не обязательно подтверждается немедлен если сегмент был послан повторно; отправителю неоткуда знать, какой сегм подтверждается полученным ACK — отправленный впервые или еще раз; ус вия в сети могут внезапно измениться. Эта проблема не имеет полного реше Всегда будет существовать некоторая неуверенность в оптимальности устан ки таймера повторной передачи.

Если происходят какие-либо изменения в поведении сети, велика веро ность того, что статический таймер повторной передачи неадекватно отреа рует на эти изменения: его значение будет либо слишком мало, либо сли ком велико. Поэтому все реализации протокола TCP проводят оценку текущ времени передачи, которая осуществляется в процессе наблюдения за хар тером изменения задержки последних посланных сегментов. Затем устана ливается значение, которое немного превышает полученную оценку време передачи.

Рассмотрим один из подходов, который учитывает среднее наблюдаемое в мя для некоторого количества посланных сегментов. Если это среднее вр достаточно точно предсказывает будущие задержки, то соответствующее з

ние таймера повторной передачи приведет к очень хорошей производительности. Усредненное время можно определить, используя следующее выражение:

$$ARTT(K+1) = \frac{1}{K+1}\sum_{i=1}^{K+1} RTT(i),$$

где $RTT(i)$ — время обращения, наблюдаемое для i-го переданного сегмента, а $ARTT(K)$ — среднее время обращения для первых K сегментов. Это выражение можно представить в другом виде:

$$ARTT(K+1) = \frac{K}{K+1} ARTT(K) + \frac{1}{K+1} RTT(K+1)$$

Следует учесть, что здесь все параметры имеют равный вес, то есть каждый параметр увеличивается на константу $1/(K+1)$. Как правило, больший вес, или большая степень доверия, дается более новым, последним результатам измерений, так как они наиболее точно отражают будущее поведение сети. Общий метод предсказания следующего усредненного значения времени обращения на основе предыдущей серии измерений времени приведен в документе RFC 793. В основе метода лежит формула:

$$SRTT(K+1) = \alpha \times SRTT(K) + (1-\alpha) \times RTT(K+1),$$

где $SRTT(K)$ (Smoothed Round-Trip Time) — так называемая усредненная оценка времени обращения. Можно сравнить это выражение с предыдущим. Используя константу α ($0 < \alpha < 1$), не зависящую от числа последних наблюдений, получаем условия, при которых все последние значения наблюдений учитываются, но более ранние наблюдения имеют меньший вес. Эту константу называют фактором сглаживания.

Наименьшее значение константы α дает наибольший вес самым последним наблюдениям. При $\alpha = 0{,}5$ наиболее весомыми становятся 4 или 5 последних наблюдений, в то время как при $\alpha = 0{,}875$ средний вес распределяется по выборке из 10 и более последних наблюдений. Достоинством именно небольшого значения константы α является то, что при этом отражаются быстрые изменения в наблюдаемых величинах. Недостаток заключается в том, что если происходит короткое волнообразное изменение в наблюдаемых значениях с последующим переходом к усредненному значению, использование небольшого значения константы α приведет к резким изменениям вычисляемого усредненного времени обращения.

Как уже упоминалось, таймер повторной передачи должен быть установлен в значение, большее чем оцененное время обращения. Для вычисления этого превышения можно использовать формулу:

$$RTO(K+1) = SRTT(K+1) + \Delta,$$

где RTO (Retransmission Time Ort) — контрольное время повторной передачи (его иногда называют тайм-аутом повторной передачи), а Δ — константа. Недостаток данного способа вычисления превышения состоит в том, что значение Δ не пропорционально значению $SRTT$. Для больших значений $SRTT$ константа Δ

относительно невелика. В этом случае флуктуации фактического значения S будут приводить к ненужным повторным передачам. Для небольших же зна ний *SRTT* величина константы Δ относительно велика и может вызвать нен ные задержки при повторных передачах потерянных сегментов. В связи с эт документ RFC 793 определяет таймер, пропорциональный *SRTT*, со следующ ми ограничениями:

$$RTO(K+1) = \min(UBOUND, \max(LBOUND, \beta \times SRTT(K+1))),$$

где *UBOUND* и *LBOUND* — фиксированные верхние и нижние границы зна ния таймера, а β — константа. Ее называют фактором изменения задержки. кумент RFC 793 не рекомендует применение фиксированных значений, но п водит диапазон изменений параметров: 0,8–0,9 для α и 1,3–2,0 для β.

Временные отметки

Для соединений, работающих с окнами большого размера, используется алгори временных отметок (TCP Timestamps), описанный в документе RFC 1323. Эт алгоритм помогает проводить точное измерение времени кругового обращен (RTT). Такие замеры необходимы для корректировки таймера повторной пере чи. Алгоритм временных отметок определяет два дополнительных поля: Timesta Value и Timestamp Echo Reply. Протокол TCP может включить поле Timestar Value в любой отправляемый сегмент. Когда подтверждается успешное получ ние этого сегмента, то получатель включает в ответный сегмент поле Timesta Echo Reply с тем же значением, что было в поле Timestamp Value. Это позвол ет протоколу TCP выполнять постоянный мониторинг времени обращения.

В операционных системах компании Microsoft вплоть до Windows 2000 пр токол TCP определял значение RTT только для одного сегмента в окне, чтоб корректировать значение RTO. При этом для вычисления RTT протокол отм чал время отправки сегмента и время получения подтверждения об успешно получении этого сегмента. Например, если размер окна составлял 8760 ба (то есть размер 6 сегментов), то для вычисления значения RTT использовал только один сегмент. Однако при введении поддержки алгоритма масштабир вания окна одного сегмента для всего окна становится недостаточно. Наприме при максимальном размере окна и масштабировании до 1 Гбайт в сети Ether будет задействоваться только одно измерение для каждых 735 440 сегментов.

В следующем примере показан фрагмент перехвата сетевого монитора с по лями временных отметок. В операционной системе Microsoft Windows 2000 ис пользование временных отметок разрешено по умолчанию.

```
...
TCP: Timestamps Option
          TCP: Option Type = Timestamps
          TCP: Option Length = 10 (0xA)
          TCP: Timestamp = 2525186 (0x268802)
          TCP: Reply Timestamp = 1823192 (0x1BD1D8)
...
```

Для определения количества повторных передач в операционной системе Microsoft Windows 2000 применяется алгоритм, регламентирующий поведение протокола TCP. Протокол TCP запускает таймер повторной передачи (retransmission timer) в момент передачи сегмента протоколу IP. Для новых соединений таймер повторной передачи инициируется со значением 3 с. Показания таймера увеличиваются каждый раз при повторной посылке сегмента на значение, указанное в параметре реестра TcpMaxConnectRetransmissions (значение по умолчанию для операционной системы Windows NT 2000 равно 2, а для Windows NT 4.0 — 3). Для действующих соединений количество попыток повторных передач управляется в реестре параметром TcpMaxDataRetransmissions (значение по умолчанию равно 5).

Для оптимизации параметров соединения таймер повторной передачи может корректироваться «на лету» с использованием SRTT. Используя этот алгоритм, протокол TCP сам настраивается с учетом задержек на данном соединении и других факторов работы сети. Соединения TCP в каналах, имеющих большую задержку, требуют большего интервала повторной передачи, чем в быстрых каналах связи.

Ниже приведен пример работы сетевого монитора с алгоритмом повторной передачи в сети Ethernet.

```
время отправ.     получат.     прот. Флаги    описание
0.000 10.57.10.32 10.57.9.138 TCP   .A..., len: 1460, seq: 8043781, ack:
   8153124, win: 8760
0.521 10.57.10.32 10.57.9.138 TCP   .A..., len: 1460, seq: 8043781, ack:
   8153124, win: 8760
1.001 10.57.10.32 10.57.9.138 TCP   .A..., len: 1460, seq: 8043781, ack:
   8153124, win: 8760
2.003 10.57.10.32 10.57.9.138 TCP   .A..., len: 1460, seq: 8043781, ack:
   8153124, win: 8760
4.007 10.57.10.32 10.57.9.138 TCP   .A..., len: 1460, seq: 8043781, ack:
   8153124, win: 8760
8.130 10.57.10.32 10.57.9.138 TCP   .A..., len: 1460, seq: 8043781, ack:
   8153124, win: 8760
```

В процессе передачи большого файла по протоколу FTP получатель был отключен от сети. Так как значение SRTT для данного соединения было очень мало (соединение между хостами высокоскоростное по сети Ethernet), то первая повторная передача произойдет примерно через 0,5 с. После этого значение таймера будет удваиваться после каждой повторной передачи. Если после пяти повторных передач не поступает подтверждений, то соединение закрывается.

Контрольное время повторной передачи RTO, которое используется отправителем для повторной передачи потерянного сегмента, на практике может быть больше, чем реальное время обращения RTT. На величину этой разности могут влиять следующие причины.

1. RTO вычисляется на основе предсказания RTT, которое было спрогнозировано исходя из значения предыдущего RTT. Но если задержки в сети флуктуируют, то спрогнозированное RTT может быть значительно больше, чем реальное RTT.

2. Если задержки на стороне получателя изменяются, то и в этом случае по ем крайне неточную оценку RTT.
3. Получатель не подтверждает каждый сегмент, а формирует подтвержде для множества сегментов и посылает его с ответными данными. При та поведении получателя невозможно точно определить RTT без привлече специальных методов.

Понятно, что при завышенном RTO в случае потери сегментов прото TCP становится инертным при проведении повторной передачи. Если прото на стороне получателя использует правило приема «в порядке отправления может быть потеряно множество сегментов. Даже в более благоприятном чае, когда получатель принимает все указанные в окне отправления сегме медленная повторная передача также не исключает проблем.

Покажем это на примере. Предположим, что станция А передала последо тельность сегментов, и в этой последовательности первый сегмент был поте До тех пор пока окно отправки станции А не станет нулевым и не истечет вре определяемое RTO, станция А и дальше будет продолжать передавать сегме без подтверждения на ранее отосланные. Станция Б получила все сегменты исключением первого. Но станция Б должна отправлять в буфер все новые в дящие сегменты, пока не получит копию потерянного сегмента: до поступле этой копии она не может отправить данные приложению и тем самым очист свой буфер. Если повторная передача потерянного сегмента будет по каким причинам задержана, то станция Б после заполнения буфера начнет отбра вать новые входящие сегменты.

Быстрая повторная передача

Для того чтобы уменьшить негативные последствия описанной выше ситуац был предложен алгоритм быстрой повторной передачи (Fast Retransmit), ко рый повышает производительность при неоптимальном выборе RTO. Быстр повторная передача определяет следующие правила работы TCP. Если на п нимающую сторону приходит сегмент не в порядке отправления, то немедлен формируются подтверждения для последнего сегмента, поступившего в поряд отправления, и для всех предыдущих. При этом подтверждения для кажд сегмента будут посылаться до тех пор, пока не поступит копия пропущенно сегмента. После этого протокол переходит в режим отправки накопленных по тверждений для всех сегментов, полученных в порядке отправления.

Когда отправитель получает двойное подтверждение на один из сегмент (а это последний сегмент, поступивший в порядке отправления), то он мож интерпретировать данное событие двояко: либо нарушена последовательнос получения (по разным причинам), либо произошла потеря сегмента. В перв случае сегмент в конечном счете может поступить и, следовательно, не обя тельно спешить с повторной передачей. Но с другой стороны, получение дво ного подтверждения отправителем может быть расценено как раннее предуп ждение, говорящее о том, что сегмент потерян и необходимо посылать коп Для окончательного принятия решения о посылке копии отправитель дол

получить еще три подтверждения на один и тот же сегмент (то есть всего четыре подтверждения). В этом случае, не дожидаясь срабатывания таймера повторной передачи, отправитель посылает копию потерянного сегмента. Рисунок 5.13 демонстрирует работу алгоритма быстрой повторной передачи.

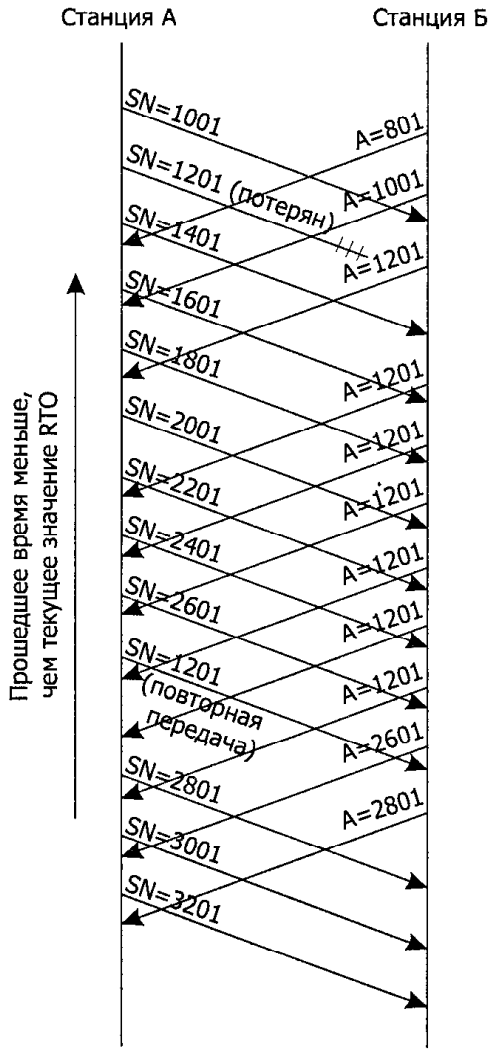


Рис. 5.13. Быстрая повторная передача

Станция А посылает последовательность сегментов, каждый из которых включает в себя поле данных размером 200 байт. Предположим, что сегмент 1201 был потерян, но станция А не будет реагировать на это событие до тех пор, пока не пройдет время RTO. Она продолжит посылать новые сегменты, пока не закроется ее окно отправки. Станция Б получает сегмент 1001 (с байтами от 1001 до 1200) и высылает подтверждение с номером 1201. Затем она получает сегмент 1401 (байты от 1401 до 1600). А так как этот сегмент нарушает последовательность отправки, станция Б повторит посылку подтверждения с номером 1201 и будет высылать подтверждение с этим номером при поступлении каждого нового сегмента. Так будет продолжаться до получения копии сегмента 1201.

За время, прошедшее между отправкой сегмента 1201 и получением четвер подтверждения с этим же номером, станция А успевает отослать 7 сегмен После получения четвертого подтверждения с номером 1201 станция А не ленно повторит последовательность сегментов, начиная с 1201.

По умолчанию в Windows 2000 сегмент посылается повторно, если пр маются три подтверждения с одним и тем же номером в последовательно и этот номер отстает от текущего. Управление осуществляется с помощью п метра TcpMaxDupAcks в реестре операционной системы (табл. 5.14).

Таблица 5.14. Параметр реестра TcpMaxDupAcks

Название параметра	TcpMaxDupAcks
Ключ в реестре	Tcpip\Parameters
Тип записи	REG_DWORD
Диапазон значений	1–3
Значение по умолчанию	2

Этот параметр определяет число двойных подтверждений с одним номе в последовательности, которые должны быть получены, прежде чем начнет ботать алгоритм быстрой повторной передачи.

Контроль над перегрузками

Контроль над перегрузками в сетях IP достаточно сложно реализовать по це му ряду причин.

- Протокол IP не ориентирован на установление соединения. Он не обеспе вает обнаружение состояния перегрузки и по этой причине не может б использован для контроля над ним.
- Хотя протокол TCP и осуществляет управление потоком, он способен ли по косвенным признакам определить перегрузку в сети. Более того, так задержки в распределенных сетях постоянно изменяются и могут быть оч большими, то информация, полученная на основании косвенных призна не является достоверной.
- Не существует распределенного алгоритма для связывания вместе различ протоколов TCP. Протоколы на разных компьютерах не могут взаимодей вовать друг с другом для поддержания определенного уровня общего пот Более того, на практике они ведут себя «эгоистично» по отношению к бодным ресурсам канала.

Сообщение «Подавление источника» (Source Quench) протокола ICMP дует рассматривать в качестве грубого инструмента сдерживания трафика отправителя, но его нельзя причислять к эффективным методам контроля перегрузками. Администратор сети может возложить задачу контроля над п грузками на протокол RSVP (Resource Reservation Protocol — протокол ре вирования ресурсов).

Протокол RSVP, который описан в документе RFC 2205, является протоколом сигнализации для предоставления гарантированного качества обслуживания в распределенной сети. Клиенты используют этот протокол для обеспечения оговоренного уровня обслуживания, того, что в англоязычной литературе называется резервированием (reservations). При отправке запроса целью применения протокола является получение информации о необходимом качестве. Функциональность клиентов этого протокола может быть встроена в маршрутизатор вместо использования ее на обычных рабочих станциях. Например, если маршрутизатор отвечает за доставку голосового потока от местной телефонной станции через распределенную IP-сеть, то он может использовать протокол RSVP для резервирования необходимых ресурсов, гарантирующих успешную доставку трафика реального времени через сеть. В свою очередь, резервирование ресурсов гарантирует необходимый уровень качества обслуживания для потока информации приложений, называемого сеансом. Сеанс является эквивалентом обычного потока информации по протоколу TCP/UDP, и он может быть единичной передачей дейтаграмм протокола IP (unicast) или многоадресной передачей (групповой — multicast). При резервировании маршрутизатор гарантирует, что сеанс получит необходимую пропускную способность и приоритет в очередях, и он будет защищать эти ресурсы от «посягательства» любых других, не обслуживаемых сеансов. Если маршрутизатор не в силах обеспечить требуемое для работы сеанса качество обслуживания, он посылает клиенту сообщение об отклонении запроса. Этот клиент может повторить запрос позже, изменить желаемые параметры качества или послать свои данные без предварительного резервирования.

Настраивать протокол RSVP в распределенной сети очень просто. Более того, чтобы его настроить, даже не требуется глубокого понимания принципов его работы. Данный протокол устанавливается на интерфейсе маршрутизатора, а не на маршрутизаторе в целом, что дает определенную гибкость. Для того чтобы настроить протокол, следует выполнить на требуемом интерфейсе команды bandwidth и ip rsvp bandwidth.

Протокол TCP может влиять на загрузку сети, управляя потоком данных с помощью рассмотренных выше методов. Но без принятия специальных мер протоколу сложно поддерживать загрузку каналов связи на оптимальном уровне. Для этого разработаны и используются алгоритмы, предотвращающие возникновение перегрузки в сети. Эти алгоритмы применяются практически во всех современных реализациях протокола. К ним относятся: алгоритмы медленного старта и предотвращения перегрузки (Slow Start Algorithm и Congestion Avoidance, RFC 1122), динамическое изменение окна при перегрузке, быстрая повторная передача и быстрое восстановление.

Медленный старт

Казалось бы, чем больший размер окна имеет отправитель, тем больше сегментов он может отправить до момента получения очередного подтверждения. Но на практике все происходит совсем иначе. В устоявшийся период передачи дан-

ных скорость отправки сегментов регулируется за счет самосинхронизации
токола. Гораздо сложнее установить начальную скорость передачи данных с
же после установления соединения, правильно выбрать алгоритм ее повы
ния, с тем чтобы единовременно не заполнить своими сегментами канал св
Решить эту проблему можно двояко. Первый способ состоит в том, что отп
витель начинает посылать данные, имея относительно большой размер о
отправки, с постепенным увеличением его размеров до такого значения, кото
оптимально для стационарного режима этого соединения. При таком подх
всегда существует определенный риск заполнения распределенной сети мно
ством сегментов еще до того момента времени, когда отправитель поймет,
он посылает данные со слишком большой скоростью. Очевидно, что основн
недостаток этого способа — возможность выбрать слишком широкое нача
ное окно отправки. Поэтому второй путь заключается в том, что в начальн
момент функционирования соединения окно отправки открывается с ми
мальным размером, который ни при каких обстоятельствах не вызовет пе
грузку. «Изюминка» — в способе наращивания размера окна отсылки. Име
для этого используется медленный старт. Здесь окно называется окном перег
ки, которое измеряется не в байтах, а в количестве сегментов. В любой мом
времени скорость передачи данных по протоколу TCP определяется следующ
соотношением:

$$AWND = MIN[CREDIT, CWND],$$

где *AWND* — разрешенный размер окна отправки в сегментах. Иначе говоря,
количество сегментов, которое разрешено послать по протоколу TCP на данн
момент без получения подтверждений успешного приема этих сегментов;

CREDIT — имеющийся на данный момент резерв отправки сегментов, ко
рый был выдан в самых последних подтверждениях;

CWND — размер окна перегрузки в сегментах. Это окно используется про
колом TCP в начальный период работы и позволяет снизить скорость переда
данных при наступлении перегрузки.

При создании нового соединения TCP устанавливает размер окна перегру
ки (CWND) в единицу. Это означает, что отправитель может послать толь
один сегмент, а затем должен ожидать подтверждения его успешного прие
и только после его получения вправе послать следующий сегмент. Кажд
раз, когда поступает подтверждение, окно CWND расширяется на едини
Это происходит до тех пор, пока размер этого окна не достигнет максимальн
значения.

В результате медленный старт гарантирует, что не будет послано слиш
много сегментов в распределенную сеть, даже если последняя работает на пр
деле своих возможностей. Ведь только по мере прибытия подтверждений отпр
витель расширяет свое окно отправки. Таким образом, можно сказать, что TC
принимает в расчет загруженность канала связи.

Выше изложен теоретический ход событий. Правильнее было бы назва
этот процесс экспоненциальным стартом, так как размер окна CWND на
мом деле нарастает по экспоненте. Когда прибывает первое подтвержде
протокол TCP увеличивает размер окна CWND в два раза и посылает ср

два сегмента. После этого окно CWND инкрементируется для каждого приходящего подтверждения. Следовательно, после получения подтверждения на два последних отосланных сегмента отправитель может послать уже 4 сегмента. После того как на эти 4 сегмента поступят подтверждения, протокол может послать 8 сегментов и т. д. На рис. 5.14 показан пример работы алгоритма медленного старта.

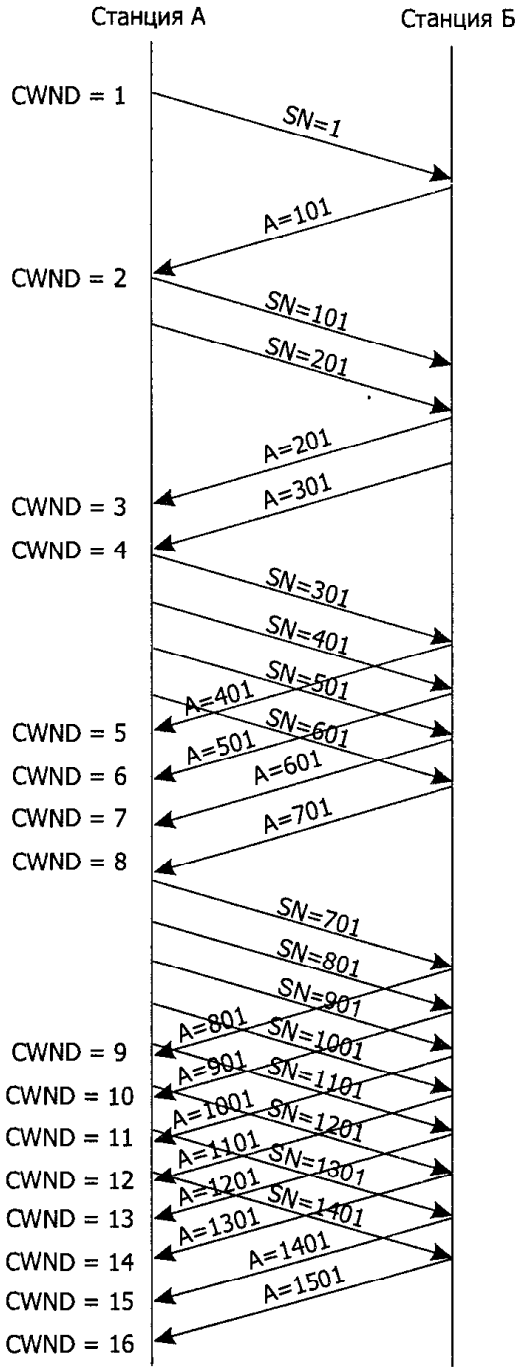


Рис. 5.14. Медленный старт

В этом примере станция А посылает сегмент длиной 100 байт и по истечен времени, приблизительно равного четырехкратному периоду кругового обращ ния (RTT), станция А получает возможность заполнить канал связи непрерыв ным потоком своих сегментов.

Медленный старт работает достаточно эффективно при создании соедин ния, позволяя протоколу TCP на стороне отправителя быстро определять опт мальный размер окна для данного соединения. А если несмотря на все мер предосторожности перегрузка все-таки наступила? Рассмотрим поведение этог алгоритма в такой ситуации. Предположим, что устанавливается соединени с медленным стартом, и до того момента, как размер окна перегрузки CWND достигает размера предельной нормы, выделенного отправителю другой сторо ной, происходит потеря сегмента. Этот факт может свидетельствовать о том, что сеть перегружена. Но не существует никаких свидетельств, позволяющих оценить серьезность возникшей ситуации. А раз так, то логичнее и безопаснее всего посту пить следующим образом: сбросить размер окна перегрузки CWND до исходно го единичного значения и начать медленный старт заново.

Такая схема может показаться вполне приемлемой, но только на первый взгляд. Давно получил подтверждение тот факт, что сеть легко перегрузить, но сложно восстановить ее нормальную работу. Иными словами, если случается перегрузка, может потребоваться достаточно длительное время и большие усилия администратора и участников сетевого обмена, чтобы восстановить рабочие параметры сети. Следовательно, при экспоненциальном росте размер окна перегрузки CWND способен быстро достичь максимума, но к этому времени сеть еще не будет восстановлена. Поэтому такой медленный старт не только не облегчит ситуацию, а наоборот, за счет слишком агрессивного поведения усложнит процесс восстановления. Для решения этой проблемы был предложен алгоритм медленного старта с линейным ростом размера окна перегрузки CWND. Теперь при наступлении перегрузки в сети выполняются следующие действия.

1. Переменной SSTHREST (граница медленного старта) присваивается значение, равное половине текущего размера окна перегрузки, то есть SSTHREST = = CWND / 2.
2. Устанавливается размер окна перегрузки CWND = 1 и выполняется медленный старт, до тех пор пока размер окна CWND не сравняется с заданным значением переменной SSTHREST (шаг 1). В этой фазе размер окна CWND будет увеличиваться на единицу после получения подтверждения на каждый посланный сегмент.
3. Если размер окна CWND превысил пороговое значение SSTHREST, то CWND необходимо увеличивать на единицу каждый раз по истечении времени обращения RTT.

Рисунок 5.15 иллюстрирует описанный вариант использования медленного старта.

При этом рис. 5.15, *а* иллюстрирует динамику роста размера окна перегрузки, аналогичную процессам, показанным на предыдущем рисунке, за исключением того, что последний сегмент потерян.

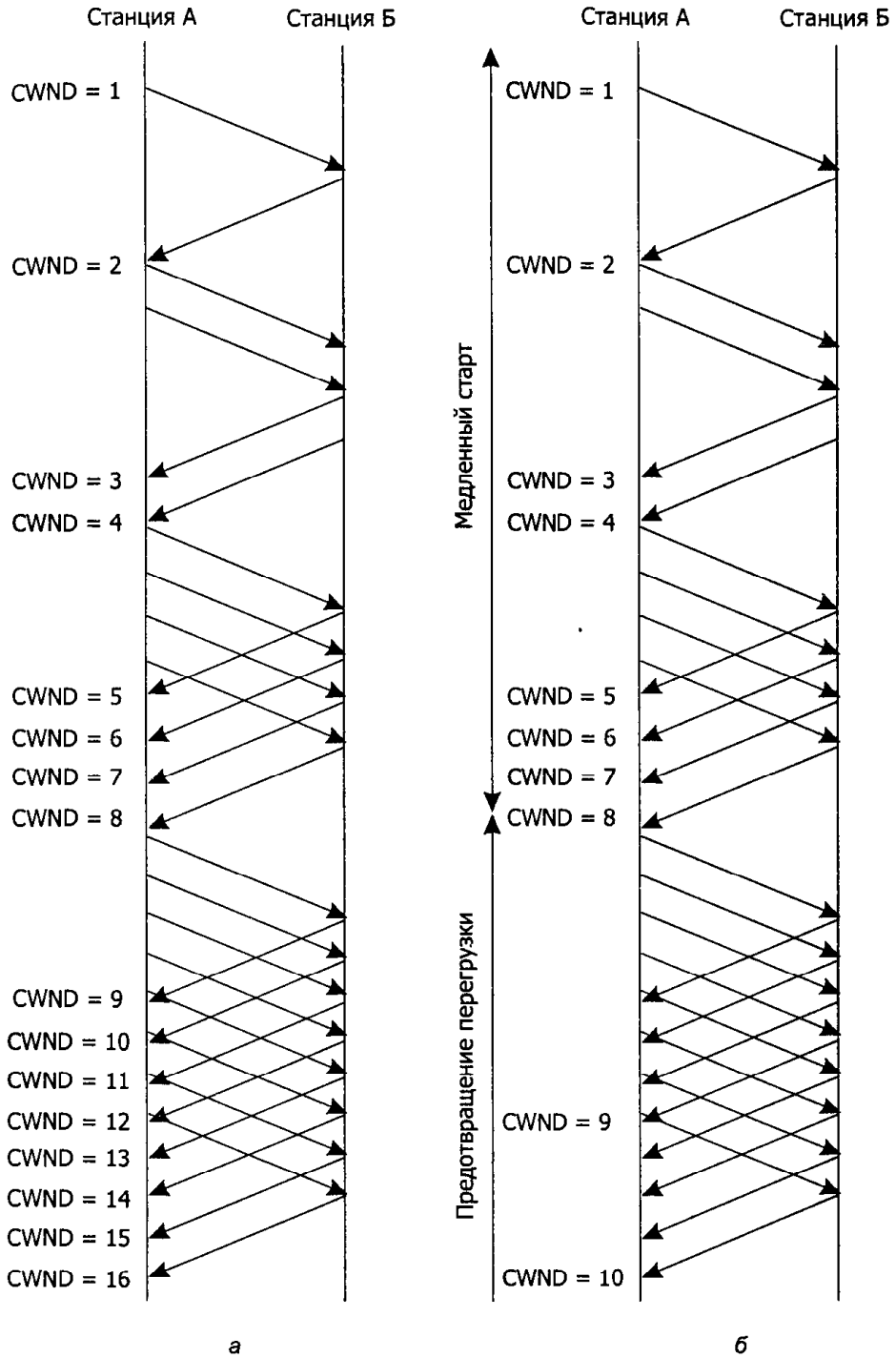


Рис. 5.15. Использование медленного старта: а — медленный старт, который заканчивается потерей сегмента, б — предотвращение перегрузки

Рисунок 5.15, *б* показывает реакцию на потерю сегмента. Значение переменной SSTHREST равно 8. Пока размер окна перегрузки не достигнет этой величины, окно растет по экспоненте. После пересечения этого рубежа размер окна CWND увеличивается линейно. Такое поведение демонстрирует рис. 5.16.

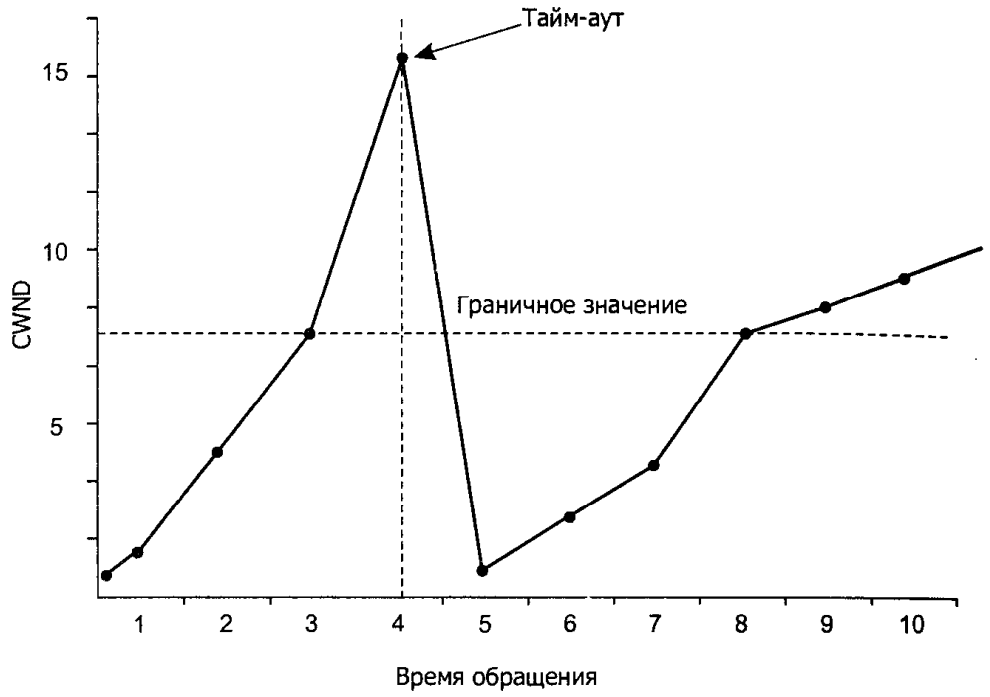


Рис. 5.16. Увеличение размера окна

На рис 5.16 окно CWND размера 8 показано как граничное значение. И ес до наступления тайм-аута прошло время, равное 4×RTT, то для повторного в жденния в этот режим уже потребуется время 11×RTT.

Механизм TCP Path MTU

Маршрутизаторы компании Cisco Systems поддерживают так называемый м ханизм TCP Path MTU Discovery, который позволяет повысить производ тельность передачи информации между двумя абонентами соединения TCP. П умолчанию этот механизм на маршрутизаторах отключен. Для его активац следует выполнить команду ip tcp path-mtu-discovery в режиме конфигурирован интерфейсов. Формат команды предполагает параметры, которые можно опу тить при ее выполнении.

Наибольший эффект от применения этой технологии будет получен, если с единения TCP в распределенной сети используются для передачи больших об емов данных. Следует отметить, что выполнение указанной команды включ ет работу рассматриваемого механизма для соединений, которые инициир маршрутизатор. Полный формат команды:

```
ip tcp path-mtu-discovery <age-timer {minutes | infinite}>
```

В качестве параметра можно указать временной интервал (age-timer), опре ляющий, как часто протокол TCP должен оценивать параметр Path MTU с н большим значением MSS. По умолчанию интервал составляет 10 минут, а м

симальное значение — 30 минут. Если указать значение infinite, то таймер будет отключен.

Значение MTU обычно устанавливается в сочетании с MSS и RWIN (TCP Receive Windows — размер окна приема). Напомню, что MSS — это наибольший сегмент данных протокола TCP, передача которого возможна в соединении. MSS должен быть меньше MTU по крайней мере на 40 байт. RWIN определяет, как много данных получатель готов принять. Если размер окна приема слишком велик, это приведет к большим потерям при передаче в случае утери информации. Если же это значение совсем мало, например 1×MSS, передача будет слишком медленной. Обычно значение RWIN равно 4×, 6× или 8×MSS.

Когда в распределенной сети пользователь работает с удаленным сервером или приложением, его производительность может снижаться за счет выполнения фрагментации передаваемой информации (точнее, фрагментации дейтаграмм протокола IP). Например, если по крайней мере один маршрутизатор в пути имеет значение MTU, равное 576 байт (что зависит от физической среды передачи), а пользователь работает в операционной системе Microsoft Windows 95, для которой значение MTU по умолчанию составляет 1500 байт, это приведет к тому, что будет выполняться фрагментация.

При передаче информации по протоколу TCP последний может устанавливать бит DF (Don't Fragment) в заголовке дейтаграммы IP. Любой маршрутизатор в пути следования сегментов может иметь значение MTU, отличающееся от того, которое используют абоненты в разных частях распределенной сети. Если значение MTU в пути следования сегмента меньше, чем длина маршрутизируемой дейтаграммы протокола IP, маршрутизатор будет пытаться выполнить фрагментацию. Однако она запрещена установкой соответствующего бита, и с этого момента маршрутизатору следует информировать отправителя о том, что дейтаграмма не может быть передана далее без выполнения фрагментации. Данная функция возложена на протокол ICMP.

Большинство маршрутизаторов также могут указывать значение MTU для следующего перехода, помещая информацию в незадействованные поля. После получения сообщения ICMP протокол TCP выравнивает значение MSS до указанного MTU с учетом байтов заголовков TCP и IP. Таким образом, остальные пакеты, посылаемые по этому соединению, не будут превышать MTU в пути следования. Однако некоторые маршрутизаторы могут просто отбрасывать большие дейтаграммы, которые они не умеют фрагментировать. Если это происходит, следует выполнить настройку маршрутизатора на поддержку механизма PMTU (глава 1).

В реестре Microsoft Windows 2000 есть два параметра, которые помогают операционной системе работать с подобными маршрутизаторами (табл. 5.15 и 5.16).

Таблица 5.15. Параметр EnablePMTUBHDetect

Название параметра	EnablePMTUBHDetect
Ключ в реестре	Tcpip\Parameters
Тип записи	REG_DWORD
Диапазон значений	0, 1 (False, True)
Значение по умолчанию	0 (False)

Равенство параметра EnablePMTUBHDetect единице (true) заставляет прото TCP в процессе выполнения операции PMTU discovery (обнаружение PM делать попытки выявлять так называемые «черные дыры» в виде маршрутиза ров. Это маршрутизаторы, которые не возвращают сообщения ICMP Destinat Unreachable, если необходимо выполнять фрагментацию дейтаграммы, но устан лен бит DF. При включении параметра EnablePMTUBHDetect протокол пытае послать сегменты без установленного бита DF, если несколько повторных пе дач одного сегмента не были подтверждены. Если этот сегмент подтверждае то MSS снижается и бит DF устанавливается для следующих пакетов. Акти ция механизма обнаружения «черных дыр» повышает максимальное количе повторных передач, которые выполняются для данного сегмента.

Таблица 5.16. Параметр EnablePMTUDiscovery

Название параметра	EnablePMTUDiscovery
Ключ в реестре	Tcpip\Parameters
Тип записи	REG_DWORD — Boolean
Диапазон значений	0, 1 (False, True)
Значение по умолчанию	1 (True)

Когда параметр EnablePMTUDiscovery (табл. 5.16) равен 1 (true), протокол T пытается обнаружить MTU в пути следования до удаленного абонента. Этот па метр включен по умолчанию, поскольку его установка помогает повысить про водительность. Сброс параметра в 0 заставляет предположить, что в распредел ной сети используется значение MTU, равное 576 байт, что и будет учитыват для всех соединений с удаленными сетями.

Администратор имеет возможность вывести значение MTU между дву устройствами вручную, используя популярную команду ping. Формат этой манды в операционной системе Microsoft Windows NT для обнаружения M следующий:

ping –f –n <количество попыток> –l <размер> <адрес получателя>.

Как показано далее, изменяя значение параметра <размер> , можно выясн соответствующий MTU. В качестве размера указывается объем посылаемых д ных без включения заголовка. При этом заголовок протокола ICMP составл 8 байт, а заголовок протокола IP — 20 байт. В приведенном ниже примере вы няется, что максимальный посылаемый объем данных составляет 1472 байт. Е к этому значению прибавить 28 байт, то получим максимальное значение M в пути до удаленного компьютера, которое равно 1500 байт:

```
C:\>ping -f -n 2 -l 1472 192.168.7.254
Обмен пакетами с 192.168.7.254 по 1472 байт:

Ответ от 192.168.7.254: число байтов=1472 время=591мс TTL=254
Ответ от 192.168.7.254: число байтов=1472 время=591мс TTL=254
```

```
C:\>ping -f -n 2 -l 1473 192.168.7.254
Обмен пакетами с 192.168.7.254 по 1473 байт:

Требуется фрагментация пакета, но установлен запрещающий флаг.
Требуется фрагментация пакета, но установлен запрещающий флаг.
C:\>
```

Далее, в примере работы анализатора пакетов протокола IP возвращает сообщение об ошибке Destination unreachable протокола ICMP. Если одна из «черных дыр» в пути до получателя только притворяется маршрутизатором, то команда ping не получит ответа, если размер данных превышает MTU, которое может обработать этот маршрутизатор. Приведенный перехват был получен при выполнении команды ping с указанием размера буфера 1000 байт. При этом в сети находится маршрутизатор, который поддерживает MTU, равный 576 байт. В ответном сообщении об ошибке указывается, что размер наибольшей дейтаграммы, которую можно передать, составляет 576 байт (0×240).

```
Src Addr Dst Addr Protocol Description
192.168.3.20 192.168.8.1
10.99.99.10 10.99.99.9 ICMP Destination Unreachable: 192.168.3.20
See frame 3
+ FRAME: Base frame properties
+ ETHERNET: ETYPE = 0x0800 : Protocol = IP: DOD Internet Protocol
+ IP: ID = 0x4401; Proto = ICMP; Len: 56
ICMP: Destination Unreachable: 192.168.3.20 See frame 3
ICMP: Packet Type = Destination Unreachable
ICMP: Unreachable Code = Fragmentation Needed, DF Flag Set
ICMP: Checksum = 0xA05B
ICMP: Next Hop MTU = 576 (0x240)
ICMP: Data: Number of data bytes remaining = 28 (0x001C)
ICMP: Description of original IP frame
...
```

6 Трансляция адресов и настройка очередей

В этой главе будут рассмотрены две технологии, которые хотя и нельзя наз базовыми, они тем не менее являются достаточно важными при построении р пределенной сети. Первая рассматриваемая технология — трансляция адресо призвана значительно облегчить задачу администратора сети по подключен организации к Интернету. Вторая — технология очередей — становится все лее актуальной при наличии интерактивного трафика, который нужно каким образом выделять при передаче для обеспечения требуемого качества.

Трансляция адресов

При применении технологии трансляции сетевых адресов (Network Addr Translation, NAT) администратор сети в состоянии расширить доступное адресное пространство путем определения так называемых внутренних (частны адресов (private addresses) и последующего преобразования их в открытые (pu addresses) адреса. Технология NAT становится очень полезной в случае, ко организации, имеющей крупную распределенную сеть, выделяют ограничен блок зарегистрированных адресов, например одну сеть класса С. При этом ад нистратор может провести дробление одной сети класса С на множество небо ших, например при помощи технологии VLSM. Другой путь — попытаться по чить новые открытые адреса от интернет-провайдера, что не всегда получает

Внутренние адреса — это блок общего адресного пространства протокола которое было выделено интернет-сообществом для использования в сетях, не им ших связи с самой сетью Интернет. Эти блоки определены документом RFC 19

- 10.0.0.0–10.255.255.255/8
- 172.16.0.0–172.31.255.255/12
- 192.168.0.0–192.168.255.255/16

Любая организация вправе без ограничений использовать эти адреса без предварительной регистрации. Очевидно, что при таком подходе множество ганизаций смогут их задействовать в своих частных сетях. Но эти организа должны понимать, что узловые маршрутизаторы в Интернете не будут мар тизировать их трафик, связанный с такими адресами. Для работы с указан

адресными блоками при выделении подсетей, применении технологии VLSM и т. п. все оговоренные правила и подходы сохраняются. На практике ввиду абсолютной доступности этих блоков администратору сети редко приходится выделять подсети — гораздо проще задействовать еще одну сеть целиком, несмотря на большую избыточность.

Остальные адреса, не входящие в это адресное пространство, управляются и координируются специальными организациями. Эти адреса маршрутизируемы в Интернете. Очевидно, что каждый открытый адрес протокола IP является уникальным. Если организация владеет достаточным количеством открытых адресов, то внедрение частных адресных блоков в сеть не является для нее прямой необходимостью.

Следует отметить, что адресная схема протокола IP имеет ограниченный набор возможных уникальных адресов, и в настоящее время становится все труднее получить большое количество открытых адресов ввиду ужесточения контроля над их распределением. На помощь приходит технология NAT, позволяющая при необходимости производить трансляцию частных адресов в открытые и обратно.

Разработано множество аппаратных и программных решений, обеспечивающих технологию NAT. Эту технологию поддерживают известные программные комплексы Checkpoint Firewall-1, операционная система Windows 2000 с приложением RRAS (Routing and Remote Access Service), маршрутизаторы Cisco Systems. Установленные на границе сети маршрутизаторы Cisco Systems могут выполнять динамическую трансляцию частных адресов в открытые, тем самым разрешая устройствам частной сети полноценно взаимодействовать с Интернетом. При этом для внутренних устройств подобная трансляция остается полностью прозрачной.

Администратор сети может настроить маршрутизатор для управления пулом открытых адресов, в котором адресов меньше, чем внутренних устройств сети, имеющих соединение с Интернетом. Маршрутизатор с настроенным пулом адресов будет динамически, по мере необходимости, выделять адреса для сеансов, а затем освобожденные адреса возвращать обратно в пул доступных адресов. При этом наиболее интересной особенностью с точки зрения безопасности является то, что устройства в Интернете работают с открытыми адресами из пула маршрутизатора и не имеют представления о том, что взаимодействие с ними осуществляется с частных адресов. Рассмотрим на примере общую схему работы технологии NAT (рис. 6.1).

В приведенной на рис. 6.1 сети хосту с частным адресом 192.168.3.1 необходимо взаимодействие с Интернетом, в частности с сервером. Администратор сети предварительно настроил технологию NAT на маршрутизаторе. В результате схема трансляции будет работать следующим образом.

1. Хост передает трафик, адресованный находящемуся в Интернете серверу. Так как хост не знает маршрута к серверу, трафик посылается маршрутизатору по умолчанию (он один в этом примере). На данном этапе адрес отправителя трафика — 192.168.3.1, и он является частным адресом в блоке 192.168.3.0 /24. Маршрутизатор, получая пакеты от хоста, обнаруживает, что последний принадлежит частной сети и, следовательно, необходима трансляция адресов. Для этого маршрутизатор просматривает настроенный пул открытых адресов и выбирает доступный адрес (193.125.203.1), который заменит собой адрес отправителя.

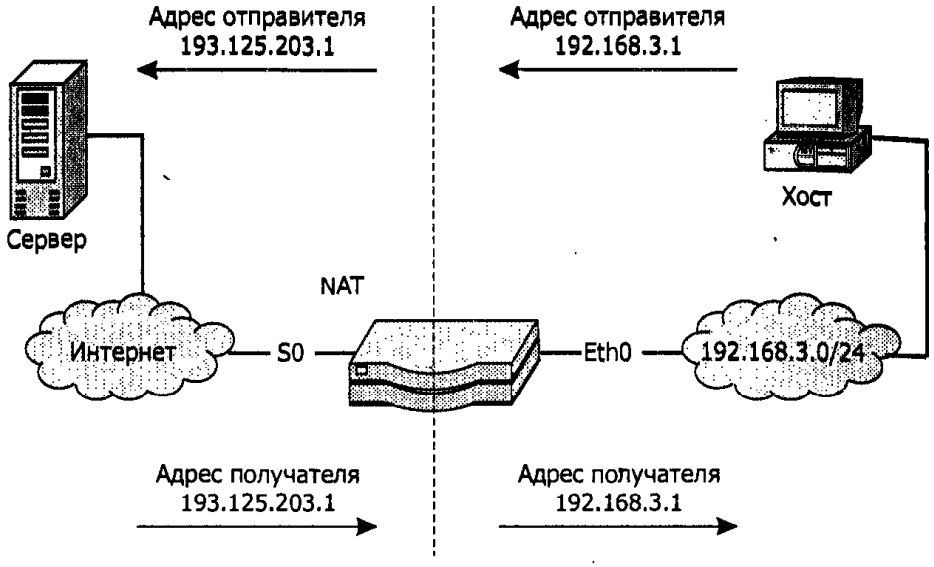


Рис. 6.1. Общая схема работы технологии NAT

2. После замены адресов маршрутизатор передает пакеты в Интернет. Эти кеты маршрутизируются узловыми маршрутизаторами до получения паке сервером.
3. Сервер после обработки пакетов отвечает по адресу 193.125.203.1, не заду ваясь о том, что действительный адрес хоста, работающего с ним, совершен другой. Ответы сервера маршрутизируются в Интернете и в финале посту ют на граничный маршрутизатор, который хранит в своей памяти инфор цию о выполненном преобразовании.
4. Граничный маршрутизатор после получения пакетов с адресом получате 193.125.203.1 вновь выполняет трансляцию адресов, но теперь уже меняет рес получателя на 192.168.3.1, и передает пакеты далее во внутреннюю с Следует отметить, что хотя в рассматриваемом примере сеть 192.168.3.0 напрямую подключена к маршрутизатору (считаем, что по протоколу Ether на практике между граничным маршрутизатором и хостом могут присутство и другие маршрутизаторы, формирующие распределенную внутреннюю организации. Из-за того что граничный маршрутизатор сначала выпол трансляцию и только после этого приступил к выполнению функции марш тизации пакета, для промежуточных маршрутизаторов трансляция адресо будет заметна.

Граничный маршрутизатор содержит таймер неактивности (idle timer), торый в течение определенного времени отслеживает момент, после котор хост перестает посылать пакеты в Интернет. Тогда маршрутизатор вернет ад 193.125.203.1 в пул свободных адресов. При необходимости администратор жет настраивать значение этого таймера.

Для рассмотренного примера настройка маршрутизатора может быть дующей:

```
ip nat pool name 193.125.203.1 193.125.203.254 prefix-length 24
ip nat inside source 2 pool name overload
```

```
!
interface serial0
ip nat outside
!
interface ethernet0
ip nat inside
!
access-list 2 permit 192.168.3.0 0.0.0.255
```

Первая команда (ip nat pool name 193.125.203.1 193.125.203.254 prefix-length 24) создает адресный пул для реализации технологии NAT — так называемые внутренние глобальные адреса (inside global addresses). Адресный пул содержит 254 адреса, начиная со 193.125.203.1 и заканчивая 193.125.203.254. Эти адреса являются зарегистрированными открытыми адресами в Интернете. Именно этими 254 адресами маршрутизатор будет впоследствии заменять внутренние адреса частной сети (так называемые inside local addresses). Данный пул именованный, его имя — name.

Следующая команда (ip nat inside source 2 pool name overload) настраивает маршрутизатор на выполнение трансляции внутренних частных адресов, которые соответствуют списку доступа с номером 2 (более подробно о списках доступа — в главе 9). При этом используется предварительно созданный пул с именем name. Трафик, который не соответствует списку доступа с номером 2, не подвергается трансляции адресов, а маршрутизируется без всяких изменений.

СОВЕТ

Если в рассмотренном примере не выполнять трансляцию адресов, то маршрутизатор будет передавать трафик с адресами отправителей из частной сети 192.168.3.0/24 в Интернет, и трафик дойдет до получателя. Важно только указать корректный адрес получателя. Но обратно пакеты уже не поступят — узловые маршрутизаторы будут игнорировать маршруты в частные сети. На этом факте часто строятся атаки типа DoS, когда атакующий должен указать только корректный адрес получателя, абсолютно не заботясь об адресе отправителя.

Во второй команде ключевое слово overload говорит маршрутизатору, что он может использовать один открытый адрес для представления множества хостов с частными адресами. Такая возможность рассматривается как разновидность мультиплексирования. Подобные средства могут потребоваться в случае, когда пул открытых адресов исчерпается из-за большого количества активных сеансов. При мультиплексировании маршрутизатор будет задействовать уникальные номера портов протоколов TCP/UDP для выделения множества внутренних хостов. Учитывая, что для каждого адреса протокола IP доступно более 64 000 портов этих протоколов, теоретически можно поддерживать до 10 000 внутренних хостов на один публичный IP-адрес, хотя практический лимит будет достигнут быстрее.

СОВЕТ

Если задействовать ключевое слово overload, можно создать пул адресов, состоящий только из одного открытого адреса. Этот адрес может быть также назначен одному из интерфейсов маршрутизатора.

Команда ip nat outside, вводимая в режиме конфигурирования для интерфе са Serial0, указывает маршрутизатору, что данный интерфейс подключен к гл бальной сети (чаще всего это сеть Интернет). Соответственно, команда ip nat ins для интерфейса Ethernet0 указывает маршрутизатору на внутреннюю сеть — ил иными словами, на расположение хостов, для которых предполагается трансл ция адресов.

Последняя команда (access-list 2 permit 192.168.3.0 0.0.0.255) создает списо доступа с номером 2, определяющий устройства, для которых требуется выпо нять трансляцию. Маршрутизатор идентифицирует пакеты, покидающие вну реннюю сеть и входящие в нее, посредством сравнения адресов отправителей списком доступа. Если найдено совпадение (то есть найден адрес отправите в сети 192.168.3.0 с маской 255.255.255.0), то маршрутизатор выделяет откры адрес из пула адресов с именем name, транслирует адреса и передает пакет в И тернет.

Если в пуле нет свободных адресов, маршрутизатор не может выполни трансляцию. В данной ситуации он будет отбрасывать все пакеты и посыла устройству во внутренней сети сообщение протокола ICMP Host Unreachab Для решения этой проблемы можно увеличить размер пула или уменьшить зн чение таймера так, чтобы адреса возвращались в пул быстрее.

ОВЕТ

При использовании динамической маршрутизации во внутренней сети следует настроить филь маршрутов таким образом, чтобы избежать рассылки внутренних маршрутов в Интернет.

У администратора сети может возникнуть потребность в статической, а динамической трансляции адресов. Подобное решение будет особенно востре ванным для тех устройств, которые должны быть достижимы из Интернета фиксированным адресам (например, серверов WWW). Поставленная цель стигается следующей командой, выполняемой в глобальном режиме:

```
Router(config)#ip nat inside source static 192.168.3.251 193.125.203.111
```

Эта команда формирует постоянное соответствие внутреннего адреса 1 168.3.251 устройства, находящегося во внутренней сети, с открытым адрес 193.125.203.111.

РИМЕЧАНИЕ

Технология NAT позволяет повысить защищенность внутренней сети, однако для организа полнофункционального защитного экрана ее нужно использовать вместе с другими решени и организационными мероприятиями, направленными на обеспечение безопасности сети.

Выше рассматривался вариант, в котором граничный маршрутизатор доступ администратору сети для управления. Однако бывают ситуации (и, на взгл автора, довольно часто встречающиеся), когда интернет-провайдер сам настр вает маршрутизатор и управляет им. С точки зрения потребителя, подоб подход предоставляет массу преимуществ, снимая все заботы по техническо обслуживанию граничного маршрутизатора. Вместе с тем решение вопроса тра ляции адресов может потребовать другого подхода.

Выбор технического решения зависит от множества факторов. Предположим, что интернет-провайдер выделил организации сеть класса C 193.125.203.0/24 и по договору самостоятельно выполняет управление маршрутизатором (это могут быть маршрутизаторы производства таких компаний, как 3Com или Nortel). Кроме того, учитывая возрастающий интерес к семейству технологий xDSL, вероятно применение устройства, совмещающего в себе функциональные возможности маршрутизатора и модема, предназначенного для работы по медным проводам выделенных линий. Но и здесь сохраняется необходимость трансляции адресов, и одним из решений может быть установка выделенного защитного экрана, который помимо фильтрации трафика поддерживает технологию NAT.

В качестве примера рассмотрим программный продукт GuardianPro v5 компании NetGuard, работающий под управлением операционной системы Windows NT 4.0. На рис. 6.2 показана функциональная схема фрагмента сети с использованием этого продукта.

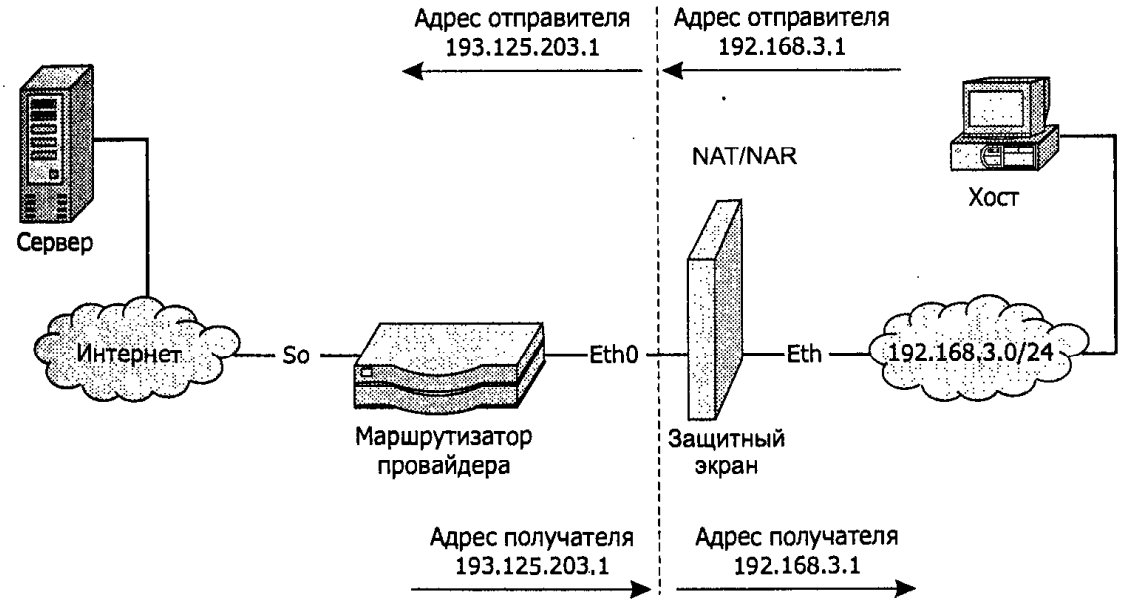


Рис. 6.2. Использование защитного экрана с поддержкой технологии NAT

В рассматриваемом примере за маршрутизатором провайдера установлен защитный экран, представляющий собой сервер на базе платформы Intel с операционной системой Microsoft Windows NT 4.0. Сервер содержит две сетевые платы, которые подключены ко внутренней сети и к сети, непосредственно связанной с Интернетом. Так как во внутренней сети используются частные адреса, то на защитном экране необходимо настроить технологию NAT. Следует отметить, что начальная настройка технологии NAT имеет много общего с настройкой маршрутизатора, и при этом большинство программных защитных экранов предоставляют удобный графический интерфейс, в то время как при настройке маршрутизатора Cisco Systems администратор ограничен командной строкой. Рисунок 6.3 показывает окно управляющей программы защитного экрана NetGuard GuardianPro, с помощью которого осуществляется в том числе и настройка трансляции адресов.

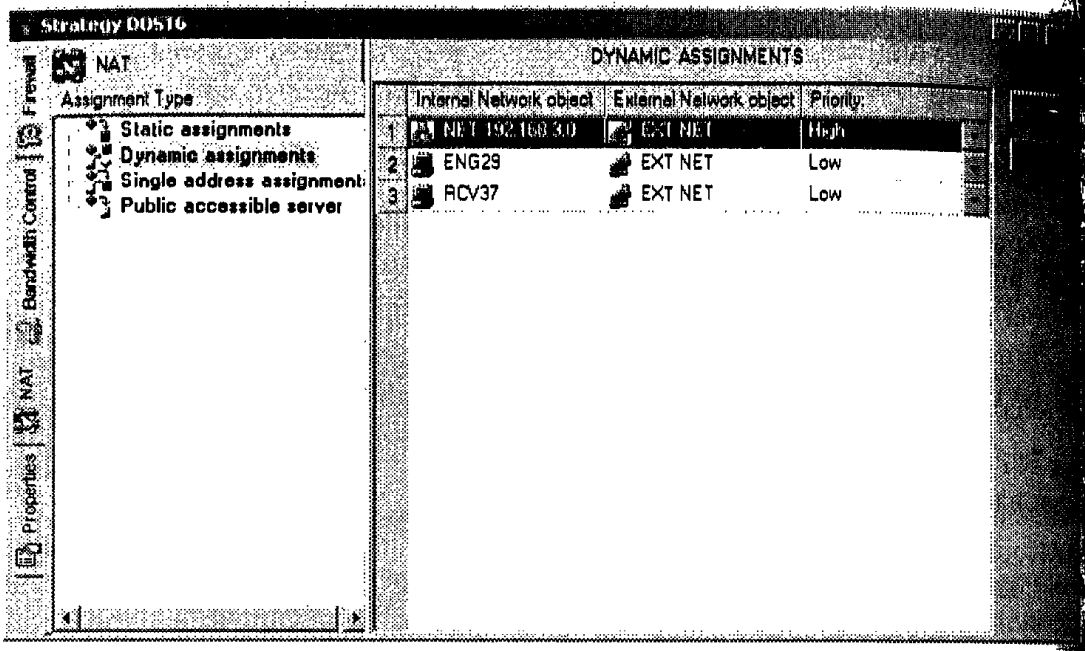


Рис. 6.3. Настройка технологии NAT на защитном экране

Первоначально администратор настраивает так называемые сетевые (NET) об
екты — сети, подсети, хосты и т. п. Каждый объект имеет символьное наимен
вание. При этом рекомендуется наэывать объекты таким образом, чтобы пото
было удобно ориентироваться в их названиях. В рассматриваемом случае объе
NET 192.168.3.0 соответствует сети класса С — 192.168.3.0/24, а объект EXT NE
как можно догадаться из названия, охватывает открытые адреса (193.125.203.10
193.125.203.200/24). Рисунок 6.4 показывает окно для создания этого объект
представляющего собой внешний пул из 100 открытых адресов.

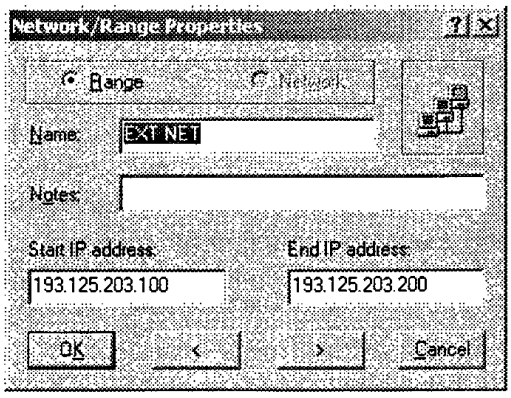


Рис. 6.4. Создание сетевого объекта

Когда объекты определены, администратор переходит в меню настройки NA
и определяет правила трансляции с указанием приоритета. Приоритет (Hig
Medium, Low) позволяет установить порядок использования пула открытых
ресов в случае их нехватки. Как и в случае с маршрутизатором компании Cis

Systems, администратор может устанавливать соответствие как для сетей, так и для отдельных устройств. Так, в рассматриваемом примере серверу ENG29 сопоставляется пул внешних адресов, благодаря чему защитный экран получает возможность автоматически подбирать свободный адрес в пуле. Этот сервер расположен в отличной от 192.168.3.0/24 сети, где только ему требуется доступ в Интернет. В таком случае нет необходимости в создании объекта, охватывающего всю сеть, к которой принадлежит сервер, а достаточно определить выполнение трансляции только для этого сервера.

Администратор сети вправе также создать объекты, соответствующие внутренним маршрутизаторам, и предписать защитному экрану выполнять трансляцию и для них. Это может быть оправдано в случае настройки на маршрутизаторах, например, протокола SNTP для синхронизации времени со внешним источником.

ИЗ ОПЫТА

Если администратор обслуживает распределенную сеть, состоящую из нескольких удаленных объектов, с использованием маршрутизации трафика, то при выборе программного обеспечения защитного экрана следует обратить внимание на возможность динамической маршрутизации (иными словами, на поддержку протоколов RIP v2, IGRP и т. п.). Если защитный экран устанавливается на компьютере, оснащенном несколькими сетевыми платами, например предназначенном для разграничения доступа в Интернет, зону DMZ (DeMilitarized Zone, демилитаризованная зона) и частную сеть, то при отсутствии динамической маршрутизации маршруты придется вводить вручную. А этот способ требует очень пристального внимания при изменении адресной схемы сети.

Продолжая разговор о защитном экране компании NetGuard, следует отметить одну из его дополнительных возможностей, которая называется NAR (Network Address Retention). Этот механизм дает возможность защитному экрану работать как proxy ARP (посредник для протокола ARP). Тогда защитный экран отвечает на все запросы протокола ARP, посылаемые маршрутизатором для разрешения адресов устройств, находящихся в защищаемой внутренней сети. Технология NAR может настраиваться администратором как дополнительная и чаще всего задействуется в случаях, когда по каким-либо соображениям нельзя изменять конфигурацию маршрутизатора или к нему нет доступа. Маршрутизатор поддерживает таблицу протокола ARP, устанавливающую соответствие между MAC-адресами устройств и их IP-адресами. После настройки технологии NAR на защитном экране таблица протокола ARP на маршрутизаторе будет выглядеть следующим образом:

```
Router#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  193.125.203.119      90     00d0.b7bd.1290  ARPA   Ethernet0
Internet  193.125.203.118      90     00d0.b7bd.1290  ARPA   Ethernet0
Internet  193.125.203.117      90     00d0.b7bd.1290  ARPA   Ethernet0
Internet  193.125.203.116      90     00d0.b7bd.1290  ARPA   Ethernet0
Internet  193.125.203.115      90     00d0.b7bd.1290  ARPA   Ethernet0
...
```

Отсюда видно, что нескольким адресам соответствует только один MAC-адрес (столбец Hardware Addr), который как раз и принадлежит внешнему сетевому

адаптеру защитного экрана. В примере показаны первые несколько строк резул тата выполнения команды show arp. На практике же выдается гораздо больш информации — по записи для каждого адреса из диапазона от 193.125.203.1 до 193.125.203.200.

Совместное использование технологии NAR и технологии NAT предоставл ет администратору дополнительные возможности: при обеспечении трансляци внутренних адресов в открытые ему не требуется выполнять никаких измен ний в настройке маршрутизатора.

Важной особенностью технологии NAT является то, что при изменении а ресов отправителя и получателя в заголовке дейтаграммы IP не проверяет и не модифицируется поле данных этой дейтаграммы. В результате приложени передающие адреса отправителей или получателей в поле данных дейтаграмм могут испытывать трудности при трансляции адресов. Это связано с тем, что п умолчанию поле данных не будет проверяться и изменяться ни защитным экра ном, ни маршрутизатором с поддержкой NAT. Учитывая такое ограничени компания Cisco Systems постоянно, от версии к версии, увеличивает возмож ности NAT по проверке содержимого поля данных для некоторых приложе ний. В настоящее время поддержку обеспечивают такие приложения, как H.32 RealAudio, VDOLive, NetBIOS поверх TCP/IP и т. д. (для получения полно перечня поддерживаемых приложений следует посетить сайт www.cisco.com).

СОВЕТ

Следует обратить особое внимание на ресурсы маршрутизатора, требуемые для выполнения тран ляции адресов. Если наблюдения показывают значительную загруженность процессора, то можно посоветовать остановиться на других решениях, например на специализированном защитном экр не Cisco PIX Firewall.

Настройка очередей на маршрутизаторе

Очереди на маршрутизаторе используются для хранения исходящих пакетов если канал связи находится в состоянии перегрузки. Когда маршрутизатор при нимает пакеты и затем обрабатывает их быстрее, чем их можно передать чере исходящий канал связи, он временно помещает их в область памяти, называе мую очередью. Эта очередь назначена исходящему интерфейсу. По выходу и состояния перегрузки канала маршрутизатор удаляет пакеты из очереди и по сылает их в линию связи. Существует достаточно много способов организации очередей. Рассмотрим некоторые из них.

Очередь типа FIFO

Если маршрутизатор получает пакеты быстрее, чем он может отправить их чере какой-либо порт, он помещает пакеты в очередь. Затем, в простейшем случае они отправляются дальше в порядке поступления, то есть реализуется принци «первым пришел, первым ушел» — FIFO (First In, First Out). Алгоритм FIFO является основополагающим в технологии очередей.

Этот алгоритм довольно эффективен, но опыт управления сетями показывает, что он далеко не оптимален. Будучи самым простым в реализации, он создает тем не менее серьезные проблемы. Основной недостаток такой стратегии в том, что она не позволяет ввести несколько уровней качества обслуживания.

При организации очереди в порядке поступления для каждого порта используется только одна выходная очередь, поэтому все потоки трафика оказываются смешаны. В связи с этим возникают две главные проблемы. Первоочередной интерактивный трафик может быть легко блокирован объемной передачей данных, а переполнение очереди чревато потерей пакетов во всем трафике, а не только в том, который переполнил очередь. Типичным является следующий пример: из-за того что трафик транзакций замедлен происходящей одновременно передачей файла по протоколу FTP, возможен разрыв сеанса связи, что приведет к неработоспособности приложения, отвечающего за транзакции. Это наиболее очевидный недостаток очереди FIFO.

Вторая проблема опасна не меньшими трудностями. Когда происходит переполнение очереди, могут быть потеряны пакеты от всех смешанных потоков трафика. Это заставляет все проходящие потоки протокола TCP (напомним, что так как сегмент TCP инкапсулируется в дейтаграмму IP, можно говорить о потоках TCP) выполнять повторную передачу, что будет сопровождаться последующими всплесками трафика. В этих условиях алгоритмы протокола TCP «Медленный старт» и «Предотвращение перегрузки» могут не помочь. Такие короткие синхронизированные волны трафика усугубляют переполнение на маршрутизаторах.

Попытки упредить переполнение очереди за счет увеличения размера самой очереди вызывают другие проблемы. Большая разница между задержкой при пустой очереди и задержкой при полной очереди увеличивает флуктуации трафика в сети. Повышенная флуктуация требует наличия большего буфера для своего «гашения» на принимающей стороне. По этой причине при использовании больших промежуточных очередей почти невозможно выполнить требования трафика реального времени. Когда маршрутизатор удаляет старые пакеты из очереди, то сначала он удаляет пакеты, поступившие раньше других («старые» пакеты). Такая стратегия характерна для FIFO. Общая схема работы с очередями показана на рис. 6.5.

Рисунок 6.5, *а* иллюстрирует ситуацию, когда исходящий канал связи находится в состоянии перегрузки. При этом пакет 1 не покинул маршрутизатор полностью, а другой пакет (пакет 2) только поступает в маршрутизатор. Маршрутизатор за оставшееся время до полного выхода первого пакета может обработать второй пакет, но медленный канал задерживает первый пакет. Так как исходящий интерфейс перегружен, то маршрутизатор, руководствуясь заданной стратегией очередей, вместо удаления (dropping) пакета (в данном случае второго) помещает этот пакет в очередь. Теперь он может быть отправлен только после того, как пакет 1 полностью пройдет через исходящий интерфейс. Время, которое необходимо пакету, для того чтобы покинуть маршрутизатор, называется задержкой сериализации (serialization delay).

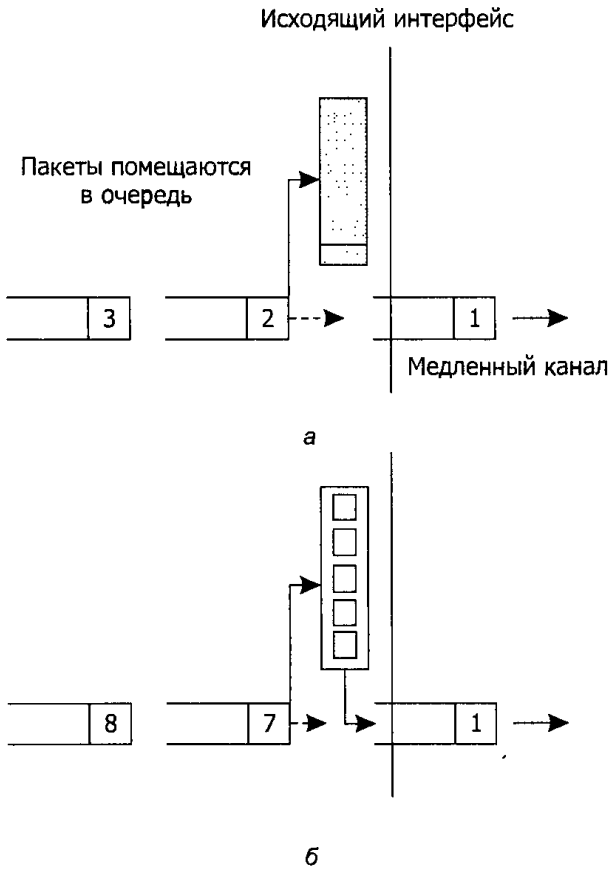


Рис. 6.5. Очередь FIFO: а — исходящий канал в состоянии перегрузки, б — освобождение очереди снизу

При «взрыве» трафика в сети (traffic burst), означающем превышение с вместной скорости поступления пакетов и скорости их обработки маршрутиз тором над пропускной способностью (bandwidth) исходящего канала связи, пр исходит перегрузка этого канала связи и пакеты поступают в очередь. Так в приведенном примере используется стратегия организации очередей FIF пакеты покидают очередь в том порядке, в котором они туда поступили. Следу отметить, что если количество появившихся за короткий промежуток време пакетов слишком велико, то соответственно повышается и скорость запол ния очереди до максимального размера, то есть очередь быстрее переполняет В этом случае маршрутизатор будет отбрасывать следующие поступающие п кеты, до тех пор пока в очереди не освободится место для новых. Такое повед ние маршрутизатора называется «сбросом хвоста» (tail drop condition). Так к сам факт удаления пакетов не является положительным явлением (он приводи к еще большему «наплыву» пакетов из-за повторных передач информации), в хорошо продуманных и разработанных сетях необходимо приложить мак мум усилий для снижения вероятности появления описанной ситуации. Одн если подобная ситуация возникает, то администратору сети следует по возм ности оценить активность и тип трафика, проходящего через маршрутизатор. необходимо для выявления и контроля источника «взрывоопасного» трафи

Возможно, придется увеличить пропускную способность исходящего канала связи, чтобы пакеты могли покидать очередь с большей скоростью. Рассмотрим пример гипотетической сети, в которой маршрутизатор передает пакеты из двух локальных сетей в глобальную сеть (рис. 6.6).

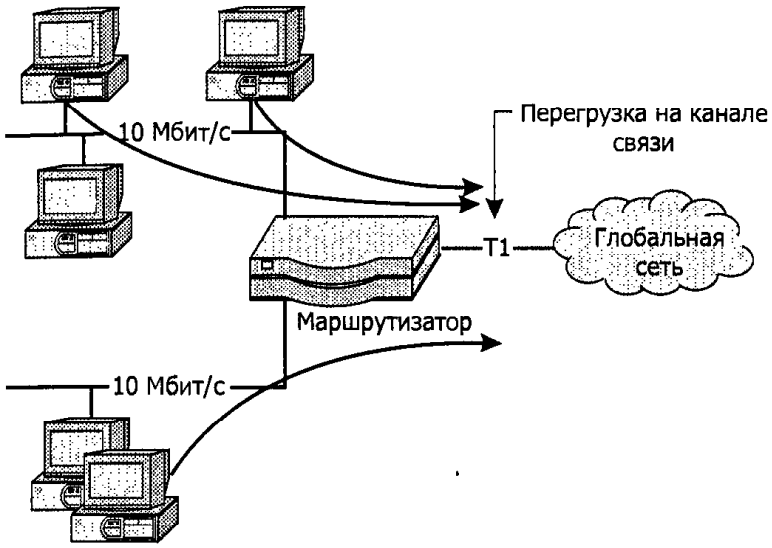


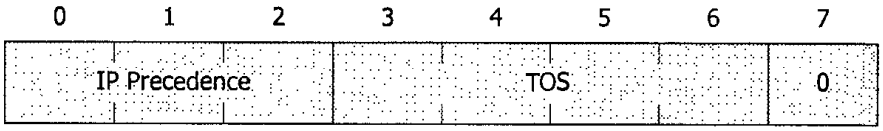
Рис. 6.6. Пример сети с перегрузкой

В этой сети несколько клиентов подключены к интерфейсам маршрутизатора. Как видно из рисунка, пропускная способность исходящего канала связи равна T1 (1,544 Мбит/с), что значительно меньше, чем скорость на интерфейсах локальной сети (10 Мбит/с). В случае, если клиенты локальной сети начинают генерировать значительный объем трафика, направленного в глобальную сеть через маршрутизатор (что, в общем, является типичной ситуацией для большинства удаленных офисов), то канал связи с глобальной сетью быстро достигнет состояния перегрузки. Это заставит маршрутизатор отправлять пакеты в очередь и, естественно, задерживать их доставку получателям. Если интерфейс глобальной сети маршрутизатора использует очередь FIFO, то в результате пакеты от одного клиента (даже если они короче по размеру и их общее количество меньше) будут вынуждены ожидать ухода пакетов от других клиентов, которые начали работать несколько раньше. Если предположить, что этот клиент работает с сеансом Telnet (например, это администратор сети, управляющий удаленным маршрутизатором), то слишком большой интервал между обменами информацией может привести к тому, что администратор не сможет работать в предполагаемом интерактивном режиме.

Качество обслуживания в IP-сетях

Перед рассмотрением других алгоритмов очередей необходимо остановиться на возможностях протокола IP. В заголовке каждой дейтаграммы присутствует поле «Тип сервиса» (Type of Service), состоящее из двух подполей. Первое, содержащие три бита (с 0 по 2), называется IP Precedence и служит руководством

маршрутизатору при выделении ресурсов для обрабатываемой дейтаграммы. С дующие четыре бита (с 3 по 6) определяют способ выбора следующего перех для дейтаграммы. В англоязычной технической литературе эти четыре бита зывают подполем TOS (Type Of Service — здесь может быть некая путаница, как часто все поле в заголовке обозначают как TOS). Бит 7 нулевой (рис. 6.7



111 — Network Control
110 — Internetwork Control
101 — Critical
100 — Flash override
011 — Flash
010 — Immediate
001 — Priority
000 — Routine

1000 — Minimize delay
0100 — Maximize throughput
0010 — Maximize reliability
0001 — Minimize monetary cost
0000 — Normal service

Рис. 6.7. Поле «Тип сервиса»

Остановимся более подробно на подполе TOS. Обычно это подполе устана ливается отправителем дейтаграммы для определения требуемого качества о служивания, которое по возможности должно быть обеспечено. На практи маршрутизаторы часто игнорируют его содержимое. Однако если проверка держимого этого подполя на маршрутизаторе выполняется, то предполагают три способа реагирования.

1. Принимается решение о маршрутизации. Например, любую дейтаграмму с прошенной минимальной задержкой (значение 1000) не следует передава по маршруту, включающему спутниковый канал связи.
2. При маршрутизации дейтаграммы маршрутизатор вправе запросить треб мые параметры качества обслуживания у распределенной сети, возможно которой наилучшим образом соответствуют значению в этом подполе. Не торые технологии распределенных сетей, например ATM, поддерживают боры совпадающих параметров при предоставлении качества обслуживани
3. Маршрутизатор может учитывать значения в этом подполе для принятия реш ния о способе обработки дейтаграммы с использованием технологии оче дей. Например, маршрутизатор может обеспечить предпочтительную обрабо в очереди для дейтаграммы, запрашивающей минимальную задержку, или оборот, попытаться избежать отбрасывания дейтаграммы, для которой за шивается максимальная надежность (значение 0010).

Два документа — RFC 1349, определяющий интерпретацию поля «Тип с виса» в заголовке дейтаграммы, и документ RFC 1812, содержащий требова к маршрутизаторам, — указывают на первый из трех вариантов реагирован а именно на выбор маршрута. Однако способ, с помощью которого маршрути тор определяет, какие маршруты в распределенной сети будут поддерживать прашиваемые параметры, выходит за рамки этих документов. В основном, е две альтернативы. Во-первых, администратор сети может предварительно ас

циировать разные маршруты с разными параметрами подполя TOS. Во втором случае протокол маршрутизации в состоянии динамически отслеживать подполе в сочетании с другими параметрами, такими как задержка, пропускная способность каналов связи и количество отброшенных дейтаграмм (примером такого протокола маршрутизации служит OSPF).

Подполе IP Precedence определяет степень срочности, или приоритет, присвоенный дейтаграмме, и этот приоритет ограничивается наибольшим значением Network Control и наименьшим Routine. Первый предполагается использовать только внутри распределенной сети для передачи служебной информации. Предшествующий наивысшему уровень Internetwork Control также ориентирован на передачу служебных данных, но сфера его применения ограничена информацией, относящейся к задачам маршрутизации, например сообщениями маршрутизации. Оставшиеся уровни не имеют строгого целевого определения.

Отметим вновь, что на практике маршрутизаторы могут игнорировать подполе IP Precedence. Как и в случае подполя TOS, если маршрутизатор поддерживает IP Precedence, он вправе предпринять одно из трех ответных действий: выбор маршрута, организацию очереди и запрос параметров качества обслуживания от нижележащей сети. Например, среди нескольких маршрутов с одинаковой метрикой может быть выбран тот маршрут, для которого меньше очередь на маршрутизаторе. Однако наибольший эффект от использования этого подполя достигается совместно с правильным выбором стратегии для очереди.

Если вернуться к документу RFC 1812, то там можно найти следующие рекомендации (следует обратить внимание на то, что практически все высказывания носят именно рекомендательный характер, оставляя широкое поле деятельности конкретным производителям).

Маршрутизатору **рекомендуется** реализовывать стратегию очереди с учетом подполя IP Precedence. В этом случае при выборе следующего пакета для отправки через интерфейс с настроенной очередью выбирается пакет с наибольшим значением приоритета срочности. Кроме того, любой маршрутизатор **может** реализовывать другие процедуры управления пропускной способностью, опираясь на значения этого подполя, однако он **должен** предоставлять команды для отключения таких процедур. В том случае, если маршрутизатор получает пакет, который уже нельзя поместить в очередь ввиду ее заполненности, он должен отбросить этот или другие пакеты. Причем маршрутизатор **может** отбросить пакет, даже если он только что поступил, и это является самой простой, но не самой лучшей политикой. В идеале, маршрутизатор должен выбирать пакет из того потока, который наиболее загружает исходящий канал связи.

Рекомендованная стратегия в распределенной сети, построенной на базе протокола IP и очередей FIFO на маршрутизаторах, следующая: маршрутизатор отбрасывает случайно выбранный пакет из очереди. Если же используется «справедливая» очередь, то пакет отбрасывается из наиболее заполненной очереди. Маршрутизатор **может** использовать вышесказанное для выбора отбрасываемого пакета. Если реализуется стратегия очереди, основанная на подполе IP Precedence, то маршрутизатор **не должен** отбрасывать пакет, который имеет большее значение приоритета. Иными словами, следует проводить некую селекцию перед отбрасыванием пакета. Кроме того, маршрутизатор **может** защищать от отбрасывания

пакеты, в которых заголовок протокола IP запрашивает максимальную над ность (0010, maximize reliability), за исключением ситуаций, когда это прави вступает в противоречие с предыдущим. Далее, маршрутизатор **может** защища фрагменты дейтаграммы IP, опираясь на тот факт, что отбрасывание одного фр мента опасно дальнейшей перегрузкой в сети, так как потребуется повтори передача всех фрагментов. И последнее правило указывает на то, что маршрут затор **может** защищать пакеты со служебной информацией для предотвращен нежелательных эффектов, например образования петель маршрутизации.

Простая арифметика показывает, что три бита подполя IP Precedence п доставляют 8 значений приоритета (от 0 до 7, или 000 до 111 в двоичном ви Стратегии очередей WFQ, CAR и RED, которые будут подробно рассмотре ниже, используют эти значения для предоставления определенным пакет (точнее, дейтаграммам) лучшего качества обслуживания, чем другим. Напомн что такая функциональность маршрутизатора, как policy routing, позволяет, п мимо прочего, устанавливать значения битов приоритета в дейтаграммах.

Установка перечисленных битов является удобным способом деления т фика на 8 классов. Однажды установленное значение для пакета остается в по мере его маршрутизации по распределенной сети. На каждом из перехо (hop) в пути следования пакета до получателя маршрутизаторы могут про рить значения данного поля в пакете и осуществить предопределенное дейст (например, выполнить высокоприоритетное обслуживание, отбросить пак установить новое значение данного поля и т. п.). В том случае, если маршрут затор не поддерживает предоставление обслуживания на основании содержим го этого поля, пакеты просто передаются далее по маршруту вне зависимости класса. Можно сказать, что средства, предоставляемые данным полем, являют базой для внедрения качества обслуживания в сети.

Чувствительные к качеству обслуживания приложения продолжают развива ся, и вполне вероятно, что использование приоритетов в дейтаграммах и ана гичные стратегии будут оставаться определяющими в достижении требуем качества обслуживания в пакетных сетях. Дополнительно, с помощью сохра ния относящейся к качеству обслуживания информации в самом пакете вме использования списков критериев, настраиваемых непосредственно на кажд из маршрутизаторов в сети, становится возможным расширить предоставлен качества обслуживания (Quality of Service — QoS) по всей распределенной се

Очередь RED

Случайное раннее обнаружение (Random Early Detection — RED) представля собой альтернативу очередям FIFO. Этот метод позволяет смягчить эффе потери пакетов даже при очень больших нагрузках. В основе такой очере по-прежнему лежит принцип FIFO, но пакеты отсеиваются случайным обра (вместо того чтобы отбрасывать сообщения из конца очереди), когда средн длина очереди за данный промежуток времени превосходит установленное з чение. Этим достигается оптимизация заполнения очереди. Данный алгори был изначально создан для протокола TCP, но он может быть применен к т фику любого протокола, когда сеть не гарантирует доставки.

Алгоритм контролирует размер исходящей очереди и отбрасывает пакеты в некоторых потоках трафика до того, как произойдет переполнение очереди. При такой «потере» пакетов срабатывает механизм определения перегрузки протокола TCP и скорость потока временно уменьшается. Таким образом, в жертву приносятся только несколько потоков, которые вынуждены производить повторную передачу и включать алгоритм медленного старта (slow start). Тем самым удается избежать возникновения волн глобальной синхронизации (global synchronization), когда множество потоков начинают одновременную ретрансляцию, вызванную потерей пакетов при переполнении буфера. Ясно, что работа алгоритма RED зависит от алгоритмов протокола TCP «Медленный старт» и «Предотвращение перегрузки». Использование очередей, к которым вынуждены прибегать маршрутизаторы и граничные устройства, не имеющие возможности непосредственно управлять непрерывными потоками трафика, в свою очередь порождает некоторые проблемы, которые решены только частично. Есть проблемы, которые не могут быть решены никакими методами организации очереди, включая взвешенные справедливые очереди WFQ и другие недавние разработки.

Технология RED на маршрутизаторе старается избегать состояния перегрузки за счет использования стратегии потоков для протокола TCP. То есть можно сказать, что RED — это механизм не управления перегрузками, а их предотвращения.

По мере увеличения трафика на исходящем интерфейсе маршрутизатора очередь на нем заполняется пакетами. Как уже говорилось, заполнение очереди происходит из-за того, что низкая пропускная способность исходящего канала связи не позволяет ему успешно справляться с возрастающим объемом поступающего трафика. И если объем трафика продолжает нарастать со скоростью, превышающей скорость исходящего канала, то вполне возможна ситуация, когда в системе не останется ни одной незаполненной очереди. Когда это происходит, маршрутизатор не имеет альтернативы, кроме удаления всех новых поступающих пакетов. Такое вынужденное поведение маршрутизатора называется «сбросом хвоста». Следует отметить, что это состояние может наступить как для очереди FIFO, так и для остальных стратегий очередей. Процесс сброса пакетов приводит к нежелательному результату, поскольку пакеты из всех потоков сбрасываются маршрутизатором в одно время и продолжают сбрасываться им до тех пор, пока система очередей частично не освободится. Потоки, которые используют протоколы, подобные TCP, будут вновь и вновь передавать те пакеты, которые были отбракованы маршрутизатором. Другие потоки, например потоки протокола UDP, напротив, никогда не передадут снова непринятые пакеты, так как эта функция возлагается на протоколы более высокого уровня. Другим потокам может вообще не потребоваться повторная отправка информации, например приложениям, передающим аудио- и видеоинформацию в реальном времени.

Для распределенной сети, построенной на базе протокола IP, нормально наличие нескольких потоков протокола TCP. И если эти потоки подвержены влиянию «сброса хвоста» маршрутизатором, то все отправители информации, переносимой этими потоками, начинают повторную передачу информации (по истечении заданного промежутка времени), причем часто это происходит в одно и то же время. Такие повторные передачи и приводят к крайне нежелательному явлению

в сети — эффекту глобальной синхронизации. Важной особенностью протоко
TCP является его уникальная способность выполнять медленный старт.
означает, что конечные станции в начале сеанса или повторной передачи высы
ют небольшие пакеты, а затем, по мере того как пакеты поступают получате
без сбросов на промежуточных узлах сети, постепенно наращивают скорость
правки информации. Более подробно механизм медленного старта описывае
в главе 5, а сейчас рассмотрим увязку его работы со стратегией RED.

Сочетание глобальной синхронизации и медленного старта протокола T
может привести к еще более разрушительным результатам. Когда у множе
потоков TCP сброшены маршрутизатором один или несколько пакетов, все
правители этих потоков начинают процедуру медленного старта в одно и то
время. Это означает, что все такие потоки одновременно снижают свою с
рость, что, естественно, приводит к резкому спаду интенсивности трафика в се
(в данном случае подразумевается, что в сети «господствует» трафик TCP).
тем отправители начинают постепенно наращивать скорость отправки и дей
вовать также одновременно, что будет продолжаться до тех пор, пока скоро
потоков не достигнут максимума, что вновь приведет к перегрузке и сбросам
кетов, и т. д.

Алгоритм RED случайным образом удаляет пакеты, основываясь на их ко
честве в очереди на интерфейсе. По мере того как объем заполняемой очере
близится к своему максимуму, RED начинает сбрасывать пакеты более агр
сивно, чтобы уйти от процедуры «сброса хвоста». Таким образом, вместо отс
всех новых пакетов при заполнении очереди происходит управление глуби
очереди с помощью случайного сброса некоторых пакетов. При пропаже пак
приложение, которому этот пакет принадлежал, снижает скорость потока T
и начинает процедуру медленного старта. Такое взаимодействие помогает сг
дить скачкообразные взрывы трафика и повышает общую производительно
системы. Следует отметить, что на низкоскоростных каналах связи страте
RED может привести к нежелательным результатам. Это связано с тем, что, к
правило, при работе на низких скоростях приложению желательно полнос
утилизировать доступную полосу пропускания. В такой ситуации RED мож
своей активностью препятствовать этому.

Полностью поддерживая стратегию предоставления качества обслуживан
RED в первую очередь сбрасывает низкоприоритетные пакеты. Выявление пр
оритетов пакетов осуществляется на основании содержимого поля TOS заголов
дейтаграммы IP. Таким образом, приложения, генерирующие высокоприорите
ный трафик, будут меньше «страдать» от утечки пакетов. Естественно, знач
тельно сократится число медленных стартов. Достижение требуемого качес
обслуживания выполняется механизмом RED посредством варьирования п
бин очередей, причем для каждого уровня приоритета используется индиви
альная граница глубины очереди. По мере заполнения очереди и превыше
границы ее глубины уровень приоритета, ассоциированный с этой границей, на
нает приближаться к уровню, при котором возможен случайный сброс паке
Соответственно, высокий уровень приоритета сочетается с более высоким п
ничным значением. Такая комбинация механизма случайного раннего обна
жения с механизмом назначения приоритета дейтаграммам IP называется WR

(Weighted RED) — взвешенным случайным обнаружением. Технология WRED рассчитана на работу на высоких скоростях и, как правило, реализуется на маршрутизаторах средних и верхних уровней.

Настройка очередей RED и WRED

Для того чтобы настроить WRED на маршрутизаторе, нужно выполнить команду random-detect по отношению к выбранному интерфейсу. После этого маршрутизатор автоматически настроит границы для очередей разных уровней приоритета.

```
Cisco1601#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco1601(config)#interface serial1
Cisco1601(config-if)#random-detect
```

Следует отметить, что различий между технологиями RED и WRED немного. Последняя просто добавляет возможность использования поля приоритета дейтаграмм IP. Когда весь трафик в распределенной сети имеет один и тот же уровень приоритета (имеется в виду трафик протокола IP), работает стратегия RED, иначе уровень приоритета сигнализирует WRED о наличии трафика разной важности. После этого пакеты низкоприоритетного трафика сбрасываются ранее, чем высокоприоритетного, по мере заполнения очереди.

В большинстве случаев администратору сети не потребуется изменять настройки стратегии WRED, принятые по умолчанию. Однако если все же имеется необходимость произвести некоторые настройки в сети, то можно выполнить команду random-detect precedence, как показано ниже.

```
Cisco1601(config)#interface serial1
Cisco1601(config-if)#random-detect precedence 0 15 35 10
```

В последней команде значение 0 указывает нулевой уровень приоритета. Трафик с таким приоритетом и является объектом данной команды. Приемлемое значение находится в пределах между 0 и 7 (ограничения поля TOS). Следующие значения —15 и 35 — определяют минимальную и максимальную границы глубины очереди, измеренные в количестве пакетов. Стратегия WRED управляет статистикой средней глубины очереди, оценивая количество пакетов на интерфейсе. Когда средняя глубина не превышает минимальной границы (в нашем примере 15 пакетов), пакеты не отбрасываются маршрутизатором. В том случае, если глубина превышает минимальное значение, маршрутизатор начинает отбрасывать пакеты. По мере увеличения среднего значения глубины очереди из-за усиления состояния перегрузки маршрутизатор начинает удалять пакеты более агрессивно. Свыше максимальной границы (35 пакетов) сбрасываются все пакеты, относящиеся к указанному уровню приоритета. Последний параметр в данной команде указывает, как быстро пакеты начинают сбрасываться, когда средняя глубина очереди достигнет максимальной границы. Значение в примере говорит о том, что здесь отсеивается один из каждых 10 пакетов.

После того как стратегия WRED настроена, проверить сделанные настройки можно с помощью команды show interface, как показано ниже. В качестве примера команда выполнялась на интерфейсе, настроенном на работу с сетью Frame

Relay при помощи команды **random-detect**. Следует отметить, что формат вы димых этой командой данных может отличаться в зависимости от используем версии Cisco IOS.

```
Cisco1601#show interfaces serial0
Serial0 is up, line protocol is up
   Hardware is QUICC Serial
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
...
   Input queue: 0/75/0 (size/max/drops); Total output drops: 63
   Queueing strategy: random early detection(RED)
      mean queue depth: 0
      drops by precedence:  class drops    threshold
                            0     63       20
                            1     0        22
                            2     0        24
                            3     0        26
                            4     0        28
                            5     0        31
                            6     0        33
                            7     0        35
...
```

Очереди с приоритетами

Технология очередей с приоритетами (priority queuing) была разработана дл того, чтобы решить одну из главных проблем очередей FIFO: она предоставля отдельные выходные очереди для различных классов трафика на одном и то же выходном порте. Очереди с приоритетами требуют, чтобы первоочередны потоки трафика были специально обозначены, например с использованием по TOS, в заголовке дейтаграммы. Кроме того, администратор сети вправе назн чить высокий приоритет для трафика, направленного по определенным сетевы адресам или к портам. Пакет из обычной очереди не может быть передан, ес есть пакеты в более приоритетных очередях. Недостаток такой организац очереди состоит в том, что менее приоритетные очереди могут нуждаться в пр пускной способности как раз в тот момент, когда сеть перегружена. Тем сам они способны усугубить ситуацию с перегрузкой, пытаясь инициализирова повторные передачи для своих низкоприоритетных потоков трафика.

Очевидное усовершенствование этого метода состоит в том, чтобы всем очер дям выделить надлежащую долю полосы пропускания. Такой метод называет справедливой организацией очереди (Fair Queuing — FQ). Справедливая очере гарантирует, например, что трафик транзакций будет проходить через марш тизаторы своевременно, причем без полной блокировки канала связи для др гих пользователей.

Очередь с приоритетами — это подход, при котором несколько очере FIFO или RED образуют одну очередь. Трафик распределяется между этими о редями в соответствии с заданными критериями. При этом трафик отправля

ся в порядке строгой очередности: сначала — трафик с высоким приоритетом, затем — со средним и т. д.

Применение на практике очередей с приоритетами вместо очередей FIFO позволяет настраивать маршрутизатор на отправку пакетов, которым на основании гибкой системы критериев присвоены различные приоритеты. Маршрутизатор отправляет высокоприоритетные пакеты быстрее, чем пакеты с более низким приоритетом. Следует отметить, что эта схема приводится в действие только в том случае, если исходящий интерфейс маршрутизатора входит в состояние перегрузки и пакеты начинают посылаться в очередь. Когда перегрузки нет, маршрутизатор просто пересылает пакеты с той скоростью, с которой он в состоянии их обработать, не обращая внимания на уровень приоритета. Такая стратегия ведения очередей (как и стратегия настраиваемых очередей — custom queuing, рассматриваемая далее) разработана для назначения приоритета трафика только в низкоскоростных каналах глобальной сети, что делает ее особенно привлекательной для нашей страны, когда большинство фирм использует именно низкоскоростные соединения со своими удаленными объектами по причине большой стоимости высокоскоростных. Основное объяснение такому разделению заключается в том, что стратегии очередей с приоритетами и настраиваемых очередей создают дополнительные высокие накладные расходы, которые также ложатся бременем на центральный процессор маршрутизатора. В зависимости от модели маршрутизатора каналы связи со скоростью менее 512 Кбит/с — хорошие претенденты на предоставление этого уровня качества обслуживания.

При использовании стратегии очередей с приоритетами выделяется четыре уровня очередей: с высоким приоритетом, средним, нормальным и низким. Помимо этого на маршрутизаторе реализуется так называемый классификатор (classifier), который отвечает за разделение поступающих пакетов по перечисленным четырем уровням, основываясь на заданных критериях. После того как маршрутизатор определил требуемое качество обслуживания для пакета, он помещает его в соответствующую очередь на исходящем интерфейсе. В случае если данная очередь достигает своего максимального объема, маршрутизатор удаляет пакет. По умолчанию действует нормальный тип очереди — если маршрутизатор не смог классифицировать пакеты, то он помещает их в очередь данного типа. При настройке маршрутизатора на использование стратегии очередей с приоритетами можно указать маршрутизатору классифицировать пакеты на основе различных критериев: размера пакета в байтах, номеров портов в протоколах TCP/UDP, номеров интерфейсов, на которые поступили пакеты, соответствующих правил, заданных в стандартных или расширенных списках доступа (standard or extended access list) и т. п.

Стратегия освобождения очереди и отправки пакетов через исходящий интерфейс следующая: всегда происходит первоочередное освобождение высокоприоритетной очереди по отношению к очереди со средним приоритетом; эта очередь освобождается раньше, чем очередь с нормальным приоритетом; а очередь с нормальным приоритетом обслуживается раньше очереди с низким приоритетом. Это простая стратегия означает, что пока существует пакет в очереди с высоким приоритетом, любые другие пакеты из очередей с более низким при-

оритетом не обслуживаются. Иными словами, очередь со средним приорите сможет начать освобождаться от ожидающих отправки пакетов только при ус вии пустой высокоприоритетной очереди. Если новый поступающий пакет кла сифицируется для поступления в высокоприоритетную очередь, обслуживан очереди со средним приоритетом приостанавливается. Однако следует учитыва что при большом объеме высокоприоритетного трафика приложения, траф которых классифицирован как низкоприоритетный, просто не смогут норма но функционировать из-за значительных задержек. Подобная ситуация (обс живание очереди с низким приоритетом приостанавливается на длительное в мя) называется «голоданием очереди» (queuing starvation). Как следствие п внедрении в распределенной сети стратегии очередей с приоритетами появля ся необходимость контроля обслуживания низкоприоритетного порождаем приложениями трафика.

Настройка очередей с приоритетами для маршрутизаторов Cisco

Настройка очередей с приоритетами начинается с определения списка прио тетов (priority list), в котором указывается уровень приоритета для кажд типа пакета, который необходимо классифицировать. Допустимые форматы манды priority-list приводятся в табл. 6.1.

Таблица 6.1. Формат команд для настройки очередей с приоритетами

Команда	Результат
Priority-list <номер списка> protocol <имя протокола> {high | medium | normal | low} <ключевое слово очереди>	Назначение приоритетов на основании типа протокола
Priority-list <номер списка> interface <тип интерфейса> <номер интерфейса> {high | medium | normal | low}	Настройка приоритетов для пакетов, поступающи с указанного интерфейса
Priority-list <номер списка> default {high | medium | normal | low}	Указание приоритетной очереди для пакетов, не соответствующих другим правилам в списке приоритетов

В качестве примера можно привести следующие команды, вводимые в бальном режиме маршрутизатора (табл. 6.2).

Таблица 6.2. Примеры команд

Команда	Результат
priority-list 1 protocol ip medium list 101	Требуем от маршрутизатора классифицировать пакеты, соответствующие списку доступа под номером 101, как трафик со средним приорите
priority-list 1 protocol ip high tcp telnet	Указываем маршрутизатору классифицировать пакеты сеанса Telnet протокола TCP как высокоприоритетный трафик

Команда	Результат
priority-list 1 interface Ethernet0 low	Требуем от маршрутизатора классифицировать пакеты, поступающие на его интерфейс Ethernet0, как трафик с низким приоритетом
priority-list 1 default normal	Сообщаем маршрутизатору, что все другие пакеты, которые не соответствуют ни одному из перечисленных выше правил, следует классифицировать как трафик с нормальным приоритетом. Данную команду можно не вводить, так как это действие по умолчанию

При добавлении нового классифицирующего правила в список приоритетов оно помещается в конец списка. В процессе анализа трафика маршрутизатор сравнивает поступающие на него пакеты с заданным списком начиная с первого правила. При нахождении соответствия пакет помещается в надлежащую очередь. В том случае, если пакет не удовлетворяет первому правилу, он сравнивается со вторым правилом в списке и т. д. Как уже отмечалось, последняя из приведенных команд является командой по умолчанию, и она не отобразится при просмотре настроек маршрутизатора по команде show running-config. После того как список приоритетов был определен, его следует присвоить соответствующему интерфейсу для активации очередей с приоритетами. Например, для того чтобы присвоить список с номером 1 интерфейсу Serial1, нужно ввести следующие команды:

```
Cisco1601#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco1601(config)#interface serial1
Cisco1601(config-if)#priority-group 1
```

Необходимость данных команд объясняется тем, что список приоритетов, по сути, является простой таблицей правил, и маршрутизатор не будет распределять исходящий трафик согласно приоритетам, до тех пор пока созданный список не будет привязан к нему с помощью команды priority-group.

Можно привести другой пример использования очередей с приоритетами. Допустим, в некоторой сети удаленные объекты подключены к центральному офису по сети Frame Relay и получают почту по протоколу POP3, а администратор часто управляет удаленными маршрутизаторами по протоколу Telnet. Для того чтобы обеспечить минимальные задержки при получении почты удаленными пользователями (в ущерб сеансам Telnet), администратор может настроить на центральном маршрутизаторе следующий список приоритетов:

```
priority-list 1 protocol ip high tcp pop3
priority-list 1 protocol ip low tcp telnet
interface serial0 priority-group 1
```

Как видно из последовательности команд, теперь центральный маршрутизатор будет помещать пакеты протокола POP3 в очередь с высоким приоритетом, а пакеты Telnet, соответственно, в очередь с низким приоритетом. Интерфейс маршрутизатора Serial0 подключен к сети Frame Relay, и на нем настроены

подинтерфейсы. Следует отметить, что команда priority-group доступна для в полнения только на интерфейсе, но не на подинтерфейсе. Для того чтобы пр верить факт установки очередей с приоритетами на маршрутизаторе, необход мо выполнить две команды: show queuing priority и show interfaces serial0:

```
Cisco1601#show queuing priority
Current priority queue configuration:

List  Queue     Args
1     high      protocol ip          tcp port pop3
1     low       protocol ip          tcp port telnet

Cisco1601#show interfaces serial0
Serial0 is up, line protocol is up
Hardware is QUICC Serial
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
LMI enq sent  182, LMI stat recvd 182, LMI upd recvd 0, DTE LMI up
LMI enq recvd 0, LMI stat sent  0, LMI upd sent  0
LMI DLCI 0  LMI type is ANSI Annex D  frame relay DTE
Broadcast queue 0/64, broadcasts sent/dropped 126/0, interface broadcasts 0
Last input 00:00:04, output 00:00:04, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops): Total output drops: 0
Queueing strategy: priority-list 1
Output queue: high 0/20/0, medium 0/40/0, normal 0/60/0, low 0/80/0
...
```

Эти команды проверяют настроенные правила в списке приоритетов под ном ром 1 и подтверждают факт активности данного списка на интерфейсе Seria Выделенные жирным шрифтом строки (например, **high 0/20/0**) указывают высокую приоритетность очереди. Три цифры, разделенные косыми черта признак того, что:

- в настоящее время в высокоприоритетной очереди нет пакетов;
- эта очередь способна вместить в себя до 20 пакетов;
- ни один из пакетов не был удален из очереди с момента последнего сбро счетчиков, который может быть выполнен либо перезагрузкой маршрути тора, либо командой clear counters.

Максимально возможное количество пакетов в очереди (иными слова глубина очереди) изменяется командой priority-list queue-limit. Эта команда, в полняемая в глобальном режиме маршрутизатора, устанавливает глубину д высокоприоритетной, средней, нормальной и низкоприоритетной очередей у занного списка:

```
Cisco1601#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco1601(config)#priority-list 1 queue-limit 25 55 70 85
Cisco1601#sh interfaces serial0
```

```
...
Serial0 is up. line protocol is up
 Queueing strategy: priority-list 1
 Output queue: high 0/25/0, medium 0/55/0, normal 0/70/0, low 0/85/0
...
```

Следуя рекомендациям, приведенным в технической литературе по маршрутизаторам фирмы Cisco Systems, без крайней необходимости лучше избегать изменения глубины очереди, оставляя ее равной значению по умолчанию. Эти значения позволяют предоставить расширенную производительность для высокоприоритетных приложений, оставляя достаточно ресурсов другим, менее важным приложениям. Как уже отмечалось, с помощью команды **show interfaces** можно контролировать количество отброшенных пакетов в очередях. Если по результатам работы данной команды видно, что пакеты в низкоприоритетной очереди отбрасываются маршрутизатором в определенном постоянном количестве, то налицо характерный признак того, что в сети присутствует высокоприоритетный трафик с неизменяемым объемом, который монополизирует канал связи. В таком случае простое увеличение размера низкоприоритетной очереди не всегда решает проблему, так как даже очень большая очередь может быть заполнена, хотя на это и потребуется больше времени. Один из подходов к решению означенной проблемы заключается в пересмотре списков приоритетов на маршрутизаторе (точнее, правил) с целью снизить количество пакетов, проходящих через высокоприоритетную очередь. Можно попытаться использовать другую стратегию очередей, например настраиваемые очереди. Если низкоприоритетная очередь требует лишь небольшого увеличения своей глубины и при этом исключается случайный «взрыв» объема высокоприоритетного трафика, то есть смысл лишь немного увеличить глубину этой очереди, скажем на 5–10 пакетов. При тонкой настройке очередей важно избегать создания очень глубоких очередей, которые могут удерживать в себе пакет длительное время, оказывая тем самым негативное влияние на работу приложения.

В заключение рассмотрения очередей с приоритетами необходимо отметить, что они больше всего подходят к тем сетям, которым требуется только два уровня качества обслуживания: высокий уровень для нескольких критичных приложений и нормальный уровень для остального трафика. Жестких правил нет — все зависит от объема высокоприоритетного трафика и приемлемости полученных задержек для низкоприоритетных приложений.

Настраиваемые очереди

Настраиваемые очереди (custom queuing) отличаются от очередей с приоритетами по ряду показателей. Так, например, при работе с настраиваемыми очередями администратор сети имеет возможность определить до 16 очередей на один интерфейс маршрутизатора, что существенно снижает ограничения, которые вводятся очередями с приоритетами. Это способствует решению проблемы «голодания очереди» (queue starvation), так как настраиваемые очереди не являются столь строгими, как приоритетные. Далее, вместо использования заранее предопределенных, именованных очередей (high, medium, normal и low), настраиваемые очереди просто нумеруются от 1 до 16. Здесь следует отметить, что

предоставляемая возможность настройки столь большого количества очере тесно связана с большими усилиями. Поэтому рекомендуется настраивать име но столько очередей, сколько требуется для предоставления желаемого качест обслуживания в сети. Это правило поможет в выборе классификационных крит риев и, в конечном итоге, повысит производительность сети в целом.

Самое существенное различие между настраиваемыми и приоритетными о редями в том, как происходит обслуживание их содержимого. Как уже упоми лось ранее, «голодание очереди» имеет место, когда высокоприоритетная очер удерживает один или несколько пакетов длительный промежуток времени. В личие от приоритетной очереди, настраиваемая обслуживается по алгорит известному под названием round-robin fashion. При этом маршрутизатор обс живает последовательно все очереди, освобождая в каждой определенное ко чество байтов. После того как последняя очередь была обслужена, цикл пов ряется вновь начиная с первой очереди. Такая стратегия позволяет устран монополизацию канала, однако недостатком является некоторое снижение п изводительности высокоприоритетных приложений по сравнению со случа использования приоритетных очередей. На рис. 6.8 показан общий прин работы настраиваемой очереди.

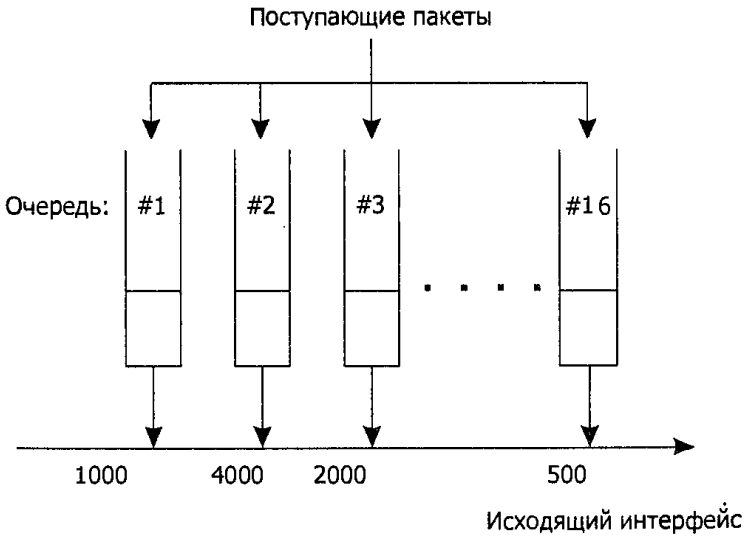


Рис. 6.8. Принцип работы настраиваемой очереди

Как и в случае очередей с приоритетами, классификатор в маршрутизат сортирует пакеты на основе предопределенных правил и помещает их в соот ствующую очередь. Маршрутизатор может освободить из высокоприоритет очередей больше байтов за один цикл, чем из низкоприоритетных. Параметр п оритета задается при настройке очереди, и не следует путать номер очереди приоритетом — номер не имеет специального значения. При классификации кетов в настраиваемой очереди применимы те же критерии, что и при ра с приоритетными очередями. Кроме того, как и очереди с приоритетами, наст ваемые очереди наиболее эффективны на низкоскоростных каналах связи, к задержка и борьба за полосу пропускания скорее правило, чем исключение

Для настройки этого типа очередей применяется команда queue-list, вводимая в глобальном режиме маршрутизатора, имеющая следующий формат (табл. 6.3).

Таблица 6.3. Формат команды queue-list

Команда	Результат
queue-list <номер списка> protocol <имя протокола> <номер очереди> <ключевое слово очереди>	Устанавливает приоритеты на основе указанного типа протокола
queue-list <номер списка> interface <тип интерфейса> <номер интерфейса> <номер очереди>	Устанавливает настраиваемую очередь на основе интерфейса, на который поступают пакеты
queue-list <номер списка> default <номер очереди>	Для тех пакетов, которые не соответствуют другим критериям, указывает номер очереди по умолчанию

Ниже показана последовательность шагов для настройки очереди на маршрутизаторе:

```
Cisco1601#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Cisco1601(config)#queue-list 3 protocol ip 1 list 101
Cisco1601(config)#queue-list 3 protocol ip 2 tcp telnet
Cisco1601(config)#queue-list 3 protocol ip 3
Cisco1601(config)#queue-list 3 interface ethernet0 4
Cisco1601(config)#queue-list 3 default 3
```

Данные команды определяют список под номером 3 для четырех очередей, которые нумеруются от 1 до 4. Смысл команд следующий:

- пакеты, соответствующие списку доступа под номером 101, помещаются в очередь 1;
- пакеты сеанса Telnet помещаются в очередь 2;
- пакеты, поступающие на интерфейс маршрутизатора Ethernet0, направляются в очередь 4;
- самая последняя команда указывает маршрутизатору помещать все остальные пакеты, которые не соответствуют вышеперечисленным правилам, в очередь 3 вместе с трафиком протокола IP, который также помещается в эту очередь.

В том случае, если администратор сети не уточнил очередь по умолчанию, маршрутизатор будет использовать очередь 1. Как и в случае очередей с приоритетами, маршрутизатор проверяет заданные классификационные правила поочередно, и когда обнаруживается совпадение, пакет помещается в соответствующую очередь.

После того как задано соответствие типов пакетов очередям, необходимо определить то количество байтов, которое маршрутизатор может освободить из очереди за один цикл обслуживания. Это осуществляется с помощью команды queue-list, имеющей следующий формат:

queue-list <номер списка> queue <номер очереди> byte-count <количество байтов>

Здесь параметр byte-count означает максимальное количество байтов для ждой очереди. Приведем пример настройки:

```
Cisco1601#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco1601(config)#queue-list 3 queue 1 byte-count 1000
Cisco1601(config)#queue-list 3 queue 2 byte-count 2000
Cisco1601(config)#queue-list 3 queue 3 byte-count 1500
Cisco1601(config)#queue-list 3 queue 4 byte-count 500
```

Из примера видно, что маршрутизатор настроен на освобождение 1000 ба за одни цикл из очереди 1, что в два раза меньше, чем для очереди 2. След вательно, трафик приложений, обслуживаемых очередью с номером 2, обслуж вается с вдва раза большей пропускной способностью, чем трафик приложени обслуживаемого очередью 1. Остальные очереди настраиваются на освобожд ние 1500 байт (очередь 3) и 500 байт (очередь 4). Необходимо отметить, маршрутизатор не будет следовать буквально этому строгому правилу и освоб ждать, например, ровно 1500 байт из очереди 3 за один цикл обслуживани Если 1500-й байт расположен в середине пакета, маршрутизатор опорожни очередь больше чем на 1500 байт, чтобы не «разрывать» этот пакет.

При настройке очередей очень важно учитывать представленную особен ность, так как настройка на слишком малое количество байтов может привест к тому, что в очереди за один цикл будет обслуживаться только один пакет (ил малое их количество), а это прямой путь к ухудшению предоставляемого каче ства обслуживания в сети. Другой крайностью будет указание на освобождени за один цикл слишком большого количества байтов. Здесь маршрутизатор буде тратить больше времени на освобождение байтов из одной очереди, задержива обработку следующей. То есть потребуется значительно больше времени на з вершение цикла обслуживания очередей, что опасно «взрывом» трафика.

Не существует строгих инструкций на ввод ограничений по количеству высво бождаемых байтов, так как это количество зависит от многих факторов. К ни можно отнести, например, следующие: средний размер пакета для каждого пр ложения, сетевую производительность, которую администратор хочет получит для каждого из приложений, количество используемых очередей и т. п. В качест рекомендации можно привести следующую схему настройки: очередь с самы низким приоритетом настраивается на освобождение двух или четырех пакето (не байтов) за один цикл обслуживания; очереди с более высоким приоритето масштабируются последовательно с учетом этих установок. На практике это вы глядит следующим образом: сначала необходимо получить с помощью сетево анализатора некоторые характеристики сети, в частности средний размер паке для низкоприоритетного трафика. Затем умножить полученный размер на чис пакетов (скажем, от 2 до 4) для обработки маршрутизатором на каждом цикл В результате получим число байтов для освобождения из низкоприоритетно очереди. Для достижения требуемого качества обслуживания в более приор тетных очередях необходимо рассчитывать число передаваемых байтов отно тельно очереди с низким приоритетом. Например, если одна из этих очере должна иметь в два раза большую полосу пропускания, чем низкоприоритетн

следует вдвое увеличить количество освобождаемых байтов для последней. В результате получим количество высвобождаемых за один цикл байтов для высокоприоритетной очереди. После создания списка очередей его нужно привязать к требуемому интерфейсу, провести наблюдение за работой приложений и, если требуется, подкорректировать настройки очередей.

Следует исключить из практики настройку освобождаемого количества байтов с таким значением, которое значительно превышает размер пакетов. Например, если таковой для приложения составляет 500 байт, не следует настраивать маршрутизатор на освобождение, скажем, 15 Кбайт за один цикл. Это правило позволяет удовлетворительно работать в большинстве случаев, однако лучше всего соотносить настройку работы настраиваемых очередей с реальным трафиком. В последнем рассмотренном примере для очереди #4 указано наименьшее значение для хранения байтов, и хотя за один цикл из этой очереди маршрутизатор освободит 500 байт, она никогда не столкнется с проблемой своего игнорирования, так как определенное обслуживание гарантируется в каждом цикле.

Настроенную очередь необходимо привязать с помощью команды **custom-queue-list** к нужному интерфейсу. Как это сделать, показано ниже:

```
Cisco1601#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco1601(config)#interface serial0
Cisco1601(config-if)#custom-queue-list 3
```

Однако перед тем как выполнять данную команду, следует убрать привязку приоритетной очереди к этому интерфейсу, в противном случае будет выдано сообщение **Must remove interface priority queueing** (необходимо удалить приоритетную очередь).

Еще раз напомним, что очереди активизируются только при возникновении состояния перегрузки. В нормальном состоянии (при приемлемом объеме трафика) маршрутизатор просто передает пакеты по мере их поступления. После привязки очереди к интерфейсу можно проверить факт ее активности с помощью команд **show queuing custom** и **show interfaces**:

```
Cisco1601#show queueing custom
Current custom queue configuration:

List  Queue    Args
3     3        default
3     1        protocol ip          list 101
3     2        protocol ip          tcp port telnet
3     3        protocol ip
3     4        interface Ethernet0
3     1        byte-count 1000
3     2        byte-count 1000
3     4        byte-count 500

Cisco1601#show interfaces serial0
Serial0 is up, line protocol is up
...
```

```
Queueing strategy: custom-list 3
Output queues: (queue #: size/max/drops)
      0: 0/20/0 1: 0/20/0 2: 0/20/0 3: 0/20/0 4: 0/20/0
      5: 0/20/0 6: 0/20/0 7: 0/20/0 8: 0/20/0 9: 0/20/0
      10: 0/20/0 11: 0/20/0 12: 0/20/0 13: 0/20/0 14: 0/20/0
      15: 0/20/0 16: 0/20/0
...
```

Информация, полученная выполнением этих команд, показывает, что на страиваемая очередь активна на интерфейсе Serial0 и что существует 16 очере дей, пронумерованных с 1 по 16 (несмотря на то, что очередей 16, трафик бу дет классифицироваться только для настроенных очередей). Информация о определенной очереди имеет общий вид 0/20/0. Это означает, что на момент вы полнения команды в очереди не было пакетов; очередь может содержать в себ максимум 20 пакетов (это значение по умолчанию); с момента последнего сбро са счетчика из очереди не удалялся ни один пакет. Необходимо обратить внима ние на то, что существует 17-я очередь с номером 0 — она называется системно очередью и резервируется для управляющих пакетов, таких как контрольные пакеты протокола маршрутизации. Данная очередь обслуживается раньше все прочих.

Если необходимо изменить значение глубины очереди, заданное по умолча нию, это можно сделать с помощью команды queue-list queue limit. Например, за данная в глобальном режиме настройка, показанная ниже, устанавливает макси мальное количество пакетов в очереди 1 равным 40.

```
Cisco1601#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco1601(config)#queue-list 3 queue 1 limit 40
...
Cisco1601#show interfaces serial0
Serial0 is up, line protocol is up
   ...
Queueing strategy: custom-list 3
   Output queues: (queue #: size/max/drops)
         0: 0/20/0 1: 0/40/0 2: 0/20/0 3: 0/20/0 4: 0/20/0
         5: 0/20/0 6: 0/20/0 7: 0/20/0 8: 0/20/0 9: 0/20/0
         10: 0/20/0 11: 0/20/0 12: 0/20/0 13: 0/20/0 14: 0/20/0
...
```

Как уже упоминалось, при настройке очередей с приоритетами не существу ет абсолютно жесткого регламента на установку лимита глубины очереди. В ос новном следует стараться не изменять значения, заданные по умолчанию в марш рутизаторах Cisco Systems, особенно если настраиваемая очередь не является определяющим фактором при достижении требуемого качества обслуживания. Если полученная статистика показывает, что иногда пакеты отбрасываются в низкоприоритетных очередей, можно попробовать поэкспериментировать с уве личением глубины очереди на небольшое количество пакетов (на 5–10).

Очереди на основе классов

Очереди на основе классов (ClassBased Queuing — CBQ) — это подход, при котором трафик делится на несколько классов. Каждый класс имеет собственную очередь, и ему выделяется некоторая часть полосы пропускания канала. Взвешенная справедливая очередь (Weighted Fair Queuing — WFQ) — частный случай очередей CBQ, когда классам соответствуют независимые потоки. Каждому классу сопоставлена одна очередь FIFO, и ей отводится некоторая часть полосы пропускания канала. При этом происходит перераспределение пропускной способности между потоками. Выделение дополнительной пропускной способности для больших потоков позволяет уменьшить задержку при их обработке. Взвешенная справедливая организация очереди является усовершенствованием очереди FQ (Fair Queuing, справедливая организация очереди). Она разбивает потоки на группы и назначает приоритет каждой группе потоков в зависимости от того, какую часть полосы пропускания те используют. Группы потоков со скромными требованиями к ширине полосы пропускания получают более высокий приоритет. При этом время ответа для таких потоков (вероятно, это или интерактивный трафик, или трафик реального времени) значительно улучшается. Кроме того, потокам внутри каждой группы также назначаются приоритеты. Механизм WFQ достаточно эффективен, поэтому во многих случаях использование различных индикаторов приоритета (например, полей «Тип сервиса» в протоколе IP или сетевых адресов или портов) не обязательно. Как следствие, от администратора требуется меньше усилий по настройке приоритетов вручную.

Обработка любой очереди производится каждым сетевым устройством в изоляции от других устройств, и нет никакого метода (отличного от сообщения «Подавление источника» (Source Quench) протокола ICMP) передачи информации между промежуточными сетевыми устройствами, чтобы они могли совместно воздействовать на скорость передачи. Поэтому все, что могут дать очереди, — это буферизация пакетов (с неизбежной задержкой), а промежуточным устройствам, не имеющим буферов, остается надеяться, что повторные передачи, вызванные ошибками, не приведут к замедлению трафика.

Настройка взвешенной справедливой очереди WFQ

Взвешенная справедливая очередь — это изощренный процесс организации очередей, который сам по себе требует очень незначительных усилий по конфигурации, так как динамически распределяет потоки трафика между приложениями и автоматически организует различные очереди для этих потоков. В терминах данной стратегии очередей эти приложения называют *собеседниками* (наиболее близкий перевод английского термина conversations), которые могут быть сеансом Telnet, передачей файла по протоколу FTP, потоками видеоинформации по протоколу IP, потоками TCP/UDP между клиентом и сервером. Эта стратегия, реализуемая на маршрутизаторах, рассматривает пакеты на предмет принадлежности определенным собеседникам. Принадлежность определяют по различной информации, например по адресам отправителя и получателя, по номерам портов TCP или UDP и т. п. После определения собеседников и идентификации принадлежащих им пакетов автоматически создается очередь для данного потока. Причем при наличии более одного потока на одном интерфейсе маршру-

тизатора организуются дополнительные очереди для каждого потока, а паке
отправляются в соответствующие очереди на основе принадлежности к поток
Так как данная стратегия реализации очередей создает очереди и помещает в н
пакеты автоматически, администратору сети не требуется вручную настраива
список классификаций для идентификации различного типа трафика, как э
было необходимо в случаях использования очередей с приоритетами и настра
ваемых очередей. Однако, как и в случае использования этих стратегий, взв
шенная справедливая очередь начинает работать только при возникновении п
регрузки на исходящем канале связи.

Для того чтобы применить на практике эту стратегию очередей, достате
но на нужном интерфейсе выполнить команду fair-queue. Вообще говоря, да
ную команду можно не вводить, так как рассматриваемая стратегия очере
включается по умолчанию на интерфейсах маршрутизаторов компании Cis
Systems, работающих на скоростях 2 Мбит/с и ниже. При этом данная кома
не будет отображаться при просмотре настроек маршрутизатора, выполняем
при помощи команды show running-config. Просмотреть текущие параметры о
реди на интерфейсе можно, выполнив команду show queue:

```
Cisco2511RJ#show queue serial 0
   Input queue: 0/75/48 (size/max/drops); Total output drops: 4
   Queueing strategy: weighted fair
   Output queue: 0/64/0 (size/threshold/drops)
         Conversations  0/2 (active/max active)
         Reserved Conversations 0/0 (allocated/max allocated)
```

Если при настройке определенного интерфейса выполнить команду no fa
queue, запрещая тем самым использование взвешенной справедливой очере
на маршрутизаторе будет реализована стратегия очередей FIFO (с учетом то
что не принимаются во внимание настраиваемые или приоритетные очереди).

Для того чтобы понять, что происходит внутри самого маршрутизатора пр
работе взвешенных справедливых очередей, сначала рассмотрим работу взвеше
ных очередей, которые являются базовой формой WFQ (рис. 6.9).

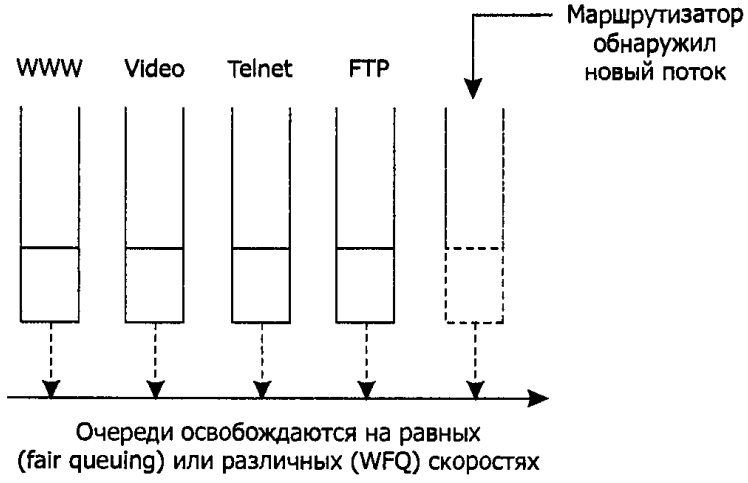


Рис. 6.9. Принцип работы взвешенных очередей

На рисунке показаны четыре активных потока информации между собеседниками, которые проходят через интерфейс маршрутизатора, и новый поток, который маршрутизатор только что обнаружил. Каждый поток имеет свою собственную очередь на маршрутизаторе, поэтому маршрутизатор динамически создает новую очередь для нового потока, а затем помещает пакеты данного потока во вновь созданную очередь. Например, маршрутизатор заталкивает все пакеты, принадлежащие сеансу Telnet, в третью очередь, а пакеты сеанса FTP между клиентом и сервером — в четвертую очередь. По умолчанию максимальное количество динамически создаваемых очередей на интерфейсе маршрутизатора на заданный момент времени составляет 256. Однако администратор сети может изменить это значение с помощью команды **fair-queue**, которая должна быть выполнена применительно к нужному интерфейсу. Например, после выполнения команды **fair-queue 64 128** можно задать максимум 128 очередей, с ограничением в 64 пакета в одной очереди. Так как каждый отдельный поток между собеседниками имеет свою собственную очередь, то он не сумеет негативно повлиять на производительность других потоков при резком изменении объема передаваемой информации — это создаст перегрузку только в его собственной очереди. Кроме того, если все существующие очереди освобождаются маршрутизатором с одинаковой скоростью, то все очереди получают равные полосы пропускания, доступные им на исходящем канале связи. Это означает, что ни один из потоков не сможет монополизировать полосу пропускания на интерфейсе. Можно проиллюстрировать воздействие взвешенных очередей на трафик разной интенсивности и показать различие между очередями, построенными на базе стратегии FIFO, и взвешенными очередями (рис. 6.10).

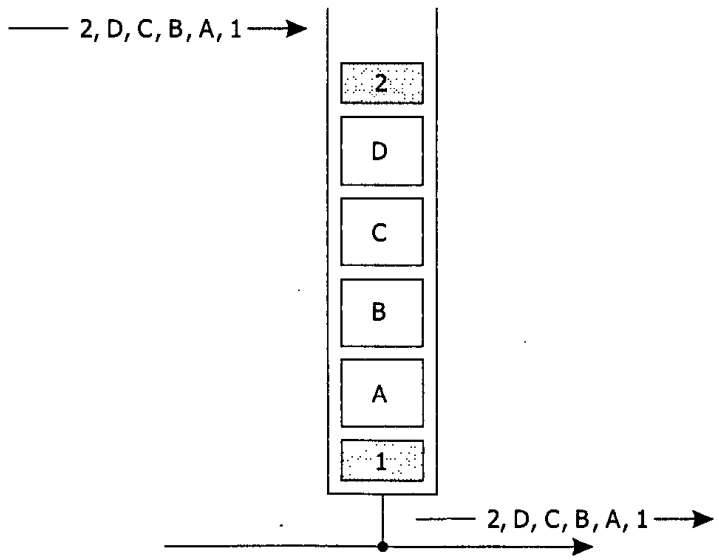


Рис. 6.10. Различия очередей с разными стратегиями

На рисунке рассматриваются два типа трафика: пакеты, предлежащие сеансу Telnet (1, 2), характеризующиеся сравнительно небольшой интенсивностью поступления, и пакеты, имеющие отношение к сеансу FTP (A, B, C, D), которые,

напротив, подразумевают высокую интенсивность, то есть формируют траф значительного объема. В рассматриваемом примере два типа потоков прохо через интерфейс маршрутизатора, настроенного на использование страте очереди FIFO. Предположим, что исходящий канал связи перешел в состоя перегрузки до поступления рассматриваемых пакетов (1, 2, A, B, C и D) и остается в этом состоянии все время, в течение которого эти пакеты поступа Естественно, что они будут помещены в очередь. Пакет 1 пришел первым, и мар рутизатор помещает его в очередь, после этого поступили четыре пакета сеа FTP и маршрутизатор помещает их в очередь вслед за пакетом 1. И, наконе последним поступает пакет 2, который оказывается в самом конце очереди. Р смотрим теперь, что происходит в процессе выхода из состояния перегруз и начале освобождения очереди. Так как рассматривается стратегия FIFO, мар рутизатор просто начинает отправлять пакеты далее по их маршруту по ме поступления. В том случае, если исходящий канал связи имеет невысокую с рость, потребуется некоторое время для передачи больших пакетов, принад жащих сеансу FTP, что задержит пакет 2. И если эта задержка значитель то пользователь, работающий с сеансом Telnet, ощутит появление пауз меж моментом нажатия клавиш и моментом появления информации на экране.

Изменим ситуацию: рассмотрим использование на маршрутизаторе взвеше ной очереди вместо FIFO. В этом случае пакеты поступают на маршрутизат в том же порядке. Маршрутизатор получает и помещает в очередь пакет 1, но в же самое время направляет пакеты сеанса FTP (A, B, C, D) в другую очере так как они принадлежат другому распознанному потоку. Затем поступает кет 2 и помещается в очередь, уже содержащую пакет 1. После того как перегр ка будет устранена, маршрутизатор начнет освобождение очередей. При эт каждой очереди будут предоставлены одинаковые шансы и, следовательно, рав полоса пропускания, доступная на интерфейсе. Это означает, что маршрути тор станет освобождать одинаковое количество байтов из каждой очереди. Ес предложить, что пакеты 1 и 2 значительно меньше пакетов сеанса FTP по раз ру, то освобождение одинакового количества байтов из каждой очереди при дет к тому, что несколько небольших пакетов будут выталкиваться параллель с одним большим. Таким образом, после освобождения пакетов 1 и 2 марщр тизатор освободит пакет A. Но если при этом для сеанса Telnet поступит е несколько небольших пакетов (3, 4 и т. д.), то некоторые из них будут освоб ждены после пакета A, но перед пакетом B. То есть в нашем примере поряд освобождения очереди будет следующим: 1, 2, A, B, C и D.

Рассмотренный пример показывает, что при использовании взвешенных оч редей низкоинтенсивный трафик (например, сеанс Telnet) не будет блокирова ся серией пакетов, принадлежащих высокоинтенсивному трафику. Более то даже если подобный трафик поступает на маршрутизатор с очень высокой с ростью, он не будет оказывать сильного воздействия на очередь трафика с низ интенсивностью.

Вместо того чтобы давать каждой потоковой очереди равную долю поло пропускания, существует возможность предоставить некоторым очередям и ответственно, их потокам большую часть полосы пропускания по сравне с другими. При этом вступает в роль фактор взвешенности, заложенный в тегию очередей WFQ. В данном случае маршрутизатор проверяет поле TOS

головках дейтаграмм протокола IP и рассчитывает так называемый *вес* (weight) для каждого пакета, который затем используется для определения очередности выхода данного пакета из очереди. Весом является простое число, обычно имеющее значение в пределах между 512 и 4096, которое значимо только по отношению к другим весам. Наименьший вес приводит к наиболее быстрому опорожнению очереди. Например, если очередь потока содержит пакеты с весом 2048, то она будет освобождаться с удвоенной быстротой (байтов в секунду) по сравнению с очередью, пакеты которой имеют вес 4096. При этом размер пакетов во внимание не принимается. Данная схема работы означает, что необходимо только помечать пакеты с требуемым значением приоритетов для организации нужного качества обслуживания.

Формула для вычисления веса очень простая:

$$\text{Вес} = 4096 / (\text{значение поля TOS} + 1)$$

Например, при значении приоритета, равном двум, вес пакета составит 1365. На старших моделях маршрутизаторов компании Cisco Systems, например Cisco 7500 с установленным модулем VIP, данная стратегия очередей может работать в распределенном режиме, известном как DWFQ (Distributed WFQ). Эта стратегия очередей призвана функционировать на очень высоких скоростях (от 45 Мбит/с до 155 Мбит/с).

Использование взвешенных справедливых очередей вместе с установкой приоритетов позволяет существенно упростить внедрение технологии качества обслуживания в крупных сетях. Как уже говорилось, если каждый пакет содержит внутри себя биты задания приоритета, он автоматически хранит информацию о требуемом качестве обслуживания. Маршрутизаторы в пути следования пакета от отправителя к получателю проверяют только поле приоритета для получения инструкций о том, как пакет следует обслуживать.

Рекомендации по выбору стратегии очередей

Выбор наиболее подходящей схемы организации очередей включает в себя следующие важные этапы.

1. Определить, находится ли в состоянии перегрузки канал связи с глобальной сетью. В том случае, если перегрузки нет, нет необходимости в дополнительных действиях. Однако если наблюдаются периоды значительной загрузки канала, то существует возможность формировать пакеты по очередям с помощью нескольких стратегий.
2. На следующем этапе следует выяснить, требуется ли строгий контроль над делением трафика согласно приоритетам и является ли автоматическая настройка маршрутизатора приемлемой. Следует отметить, что корректная настройка очередей вовсе не простая задача. Для того чтобы выполнить ее с наибольшей эффективностью, администратор сети должен проанализировать типы трафика, проходящего через рассматриваемый интерфейс, выяснить, как различать разные потоки трафика, и определить относительные приоритеты. Кроме того, следует помнить о том, что характер трафика с течением времени меняется. А это требует регулярного выполнения подобного анализа.

3. Стратегия очередей основывается на проведенном анализе трафика и опред лении относительных его приоритетов (обсуждается на шаге 2).
4. На последнем шаге нужно определить, насколько терпим выделенный т фик к задержкам.

На рис. 6.11 представлен алгоритм действий администратора сети при выб ре типа очереди.

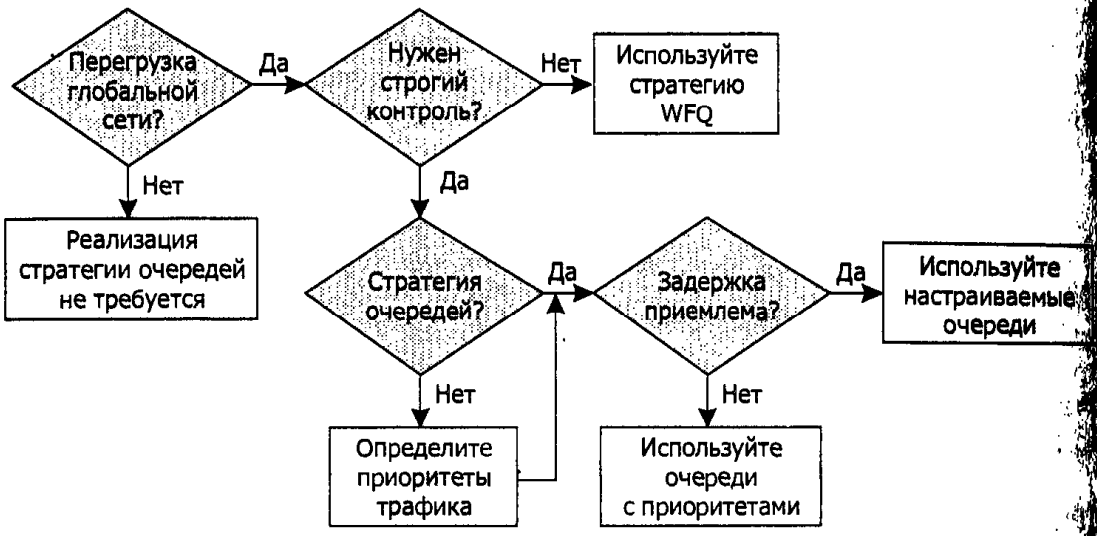


Рис. 6.11. Алгоритм выбора типа очереди

7 Настройка протоколов маршрутизации

Как известно, существует довольно много протоколов маршрутизации. Но в этой главе рассмотрен только один из них — RIP (Routing Information Protocol). Описание протокола дополнено материалом практической настройки его на маршрутизаторах. Выбор этого протокола не случаен. Пожалуй, трудно ошибиться, утверждая, что протокол RIP в настоящее время является наиболее широко используемым протоколом маршрутизации класса IGP в частных распределенных сетях, в том числе и в нашей стране. Ведь именно этот протокол в силу своей простоты чаще всего востребуется нашими специалистами.

Алгоритмы маршрутизации

Определение маршрута передачи данных происходит программно. Соответствующие программные средства носят названия протоколов маршрутизации. Логика их работы основана на алгоритмах маршрутизации. Алгоритмы маршрутизации вычисляют стоимость доставки и выбирают путь с наименьшей стоимостью. Простейшие алгоритмы маршрутизации определяют маршрут на основании наименьшего числа промежуточных (транзитных) узлов на пути к адресату. Более сложные алгоритмы в понятие «стоимость» закладывают несколько показателей, например задержку при передаче пакетов, пропускную способность каналов связи или денежную стоимость связи. Основным результатом работы алгоритма маршрутизации является создание и поддержка таблицы маршрутизации, в которую записывается вся маршрутная информация. Содержание таблицы маршрутизации зависит от используемого протокола маршрутизации. В общем случае таблица маршрутизации содержит следующую информацию:

- действительные адреса устройств в сети;
- служебную информацию протокола маршрутизации;
- адреса ближайших маршрутизаторов.

Основными требованиями, предъявляемыми к алгоритму маршрутизации, являются:

- оптимальность выбора маршрута;
- простота реализации;

- устойчивость;
- быстрая сходимость;
- гибкость реализации.

Алгоритмы маршрутизации должны быть просты в реализации и требов как можно меньше ресурсов. Они также должны быть устойчивыми к зам оборудования на первоначально выбранном маршруте, высоким нагру и ошибкам в построении сети.

Основной характеристикой алгоритма маршрутизации является скорость димости. Сходимость — это процесс согласования между маршрутизаторами формации о топологии сети, приводящий ее в стабильное состояние. Если оп деленное событие в сети приводит к тому, что некоторые маршруты становя недоступны или возникают новые маршруты, маршрутизаторы рассылают со щения об этом друг другу по всей сети. Получив эти сообщения, маршрутиза ры выполняют переназначение оптимальных маршрутов, что, в свою очере может породить новый поток сообщений. Данный процесс должен завершит причем достаточно быстро, иначе в сетевой топологии могут появиться пе или сеть вообще перестанет функционировать. Алгоритмы маршрутизации до ны быстро и правильно учитывать изменения в состоянии сети (например, каз узла или сегмента сети).

Алгоритмы маршрутизации могут быть:

- статическими или динамическими;
- одномаршрутными или многомаршрутными;
- одноуровневыми или иерархическими;
- внутридоменными или междоменными;
- одноадресными или групповыми.

Для статических (неадаптивных) алгоритмов маршруты выбираются зара и заносятся вручную в таблицу маршрутизации, где хранится информация о на какой порт следует отправлять пакет с определенным адресом. Протоко разработанные на базе статических алгоритмов, называют *немаршрутизируемы* протоколами. Примером немаршрутизируемых протоколов может служить пр кол LAT (Local Area Transport — транспортный протокол для локальных об тей) фирмы DEC. Обычно с этими протоколами работают мосты, так как они различают протоколов сетевого уровня.

При использовании динамических алгоритмов таблица маршрутизации авто тически обновляется при изменении топологии сети или трафика в ней. Ди мические алгоритмы различаются по способу получения информации о сост нии сети, времени изменения маршрутов и используемым показателям оце маршрута.

Одномаршрутные алгоритмы определяют только один маршрут. Он не все оказывается оптимальным. Многомаршрутные алгоритмы предлагают несколь маршрутов к одному и тому же получателю. Такие алгоритмы позволяют редавать информацию получателю по нескольким каналам одновременно означает повышение пропускной способности и надежности сети.

Алгоритмы маршрутизации могут работать в сетях с одноуровневой или иерархической архитектурой. В одноуровневой сети все ее фрагменты имеют одинаковый приоритет, что, как правило, обусловлено схожестью их функционального назначения. Иерархическая сеть содержит подсети (фрагменты сети). Маршрутизаторы нижнего уровня служат для связи фрагментов сети. Маршрутизаторы верхнего уровня образуют особую часть сети, называемую магистралью. Маршрутизаторы магистральной сети передают пакеты между сетями нижнего уровня.

Иерархическая структура в больших и сложных сетях позволяет значительно упростить процесс управления сетью, облегчает изоляцию сегментов сети и т. д.

Некоторые алгоритмы маршрутизации действуют только в пределах своих доменов (внутридоменная маршрутизация), другие — как в пределах своих доменов, так и в смежных с ними (междоменная маршрутизация). В данном случае домен означает область маршрутизации, в которой работает один или несколько протоколов маршрутизации. В разных доменах работают разные протоколы. Если необходима связь доменов, используется междоменная маршрутизация.

Одноадресные алгоритмы маршрутизации предназначены для передачи конкретной информации (по одному или нескольким маршрутам) только одному получателю. Многоадресные (или групповые) алгоритмы способны передавать информацию многим получателям одновременно.

Когда маршрутизатор получает пакет, он считывает адрес назначения и определяет, по какому маршруту отправить пакет. Обычно маршрутизаторы хранят данные о нескольких возможных маршрутах. Выбор маршрута зависит от нескольких факторов, в числе которых:

- применяемая система измерения длины маршрута (его метрика);
- маршрутизируемый протокол высокого уровня;
- топология сети.

На уровне маршрутизации существуют три основные группы протоколов маршрутизации (деление на группы определяется типом реализуемого алгоритма выявления оптимального маршрута):

- протоколы длины вектора;
- протоколы состояния канала;
- протоколы политики маршрутизации.

Протоколы длины вектора — самые простые и самые распространенные. Данная группа включает в себя RIP IP, RIP IPX, AppleTalk RTMP и Cisco IGRP. Свое название этот тип протокола получил от соответствующего способа обмена информацией. Маршрутизатор с определенной периодичностью извлекает данные из своей таблицы маршрутизации и помещает их в рассылаемые соседям сообщения об обновлении (можно сказать, что он рекламирует имеющуюся информацию). Соседние маршрутизаторы сверяют полученные данные со своими собственными таблицами маршрутизации и вносят необходимые изменения. После этого они сами рассылают сообщения об обновлении. Таким образом, каждый маршрутизатор получает информацию о маршрутах во всей сети. При очевидной

простоте алгоритма говорить о полной его надежности нельзя. Он может ра
тать эффективно только в небольших сетях. Это связано с тем, что в круп
сетях поток сообщений между маршрутизаторами на порядок интенсивнее
этом большинство сообщений являются избыточными (так как изменения
вой топологии происходят довольно редко). Как следствие — действительно
обходимая информация подчас долго «гуляет» по всей сети, и маршрутизато
обновляют свои таблицы с большой задержкой. Так, уже не существую
маршрут может долго оставаться в таблицах маршрутизации, а трафик, напр
ленный по такому маршруту, не достигнет своего адресата.

Протоколы состояния канала были впервые предложены в 1970 году Э
ром Дейкстрой. Эти протоколы значительно сложнее, чем протоколы дл
вектора. Вместо рассылки соседям содержимого своих таблиц маршрутиза
каждый маршрутизатор осуществляет широковещательную рассылку спи
маршрутизаторов, с которыми он имеет непосредственную связь, и списка на
мую подключенных к нему локальных сетей. Эта информация является час
информации о состоянии канала и рассылается в специальных сообщениях. К
ме того, маршрутизатор распространяет сообщения о состоянии канала тол
в случае его изменения или по истечении заданного интервала времени. Про
колы состояния канала трудны в реализации и нуждаются в значительном об
еме памяти для хранения служебной информации. Примерами этих протокол
служат OSPF, IS-IS, Novell NLSP и Cisco EIGRP.

По Дейкстре, топология сети представляется в виде неориентированного гра
Каждому ребру приписывается некоторое значение. В процессе работы алгор
ма вычисляется сумма показателей для ребер, сходящихся в каждом узле гра
Эта оценка называется *меткой* узла. При определении пути подсчитывается сум
меток на возможном пути и выбирается путь с наименьшей суммарной метко

К третьей группе протоколов относятся протоколы политики (правил) мар
рутизации. Такие протоколы наиболее эффективно решают задачу достав
информации получателю. Данная категория протоколов используется при мар
рутизации в Интернете и позволяет провайдерам получать информацию о мар
рутизации от соседних провайдеров на основании специальных критери
Таким образом, в процессе обмена вырабатывается список разрешенных мар
рутов (путей). Алгоритмы политики маршрутизации опираются на алгорит
длины вектора, но информация о маршрутах базируется на списке операто
Интернета. Примерами протоколов данной категории могут служить BGP и EG

Протокол маршрутизации RIP

Протокол RIP достаточно прост. Протоколы этой группы очень часто называ
протоколами Беллмана-Форда (Bellman-Ford), так как они опираются на ал
ритм выбора кратчайшего пути (shortest path computation algorithm), описан
еще в 1957 году. Протоколы длины вектора использовались в некоторых па
ных сетях. Наиболее известной в России является сеть ARPAnet. В Интер
протокол RIP был изначально внедрен как компонент BSD UNIX.

Итак, RIP является протоколом длины вектора. Для того чтобы объяснить принцип действия протокола, рассмотрим простую сеть, состоящую из 5 маршрутизаторов и 6 каналов связи (рис. 7.1).

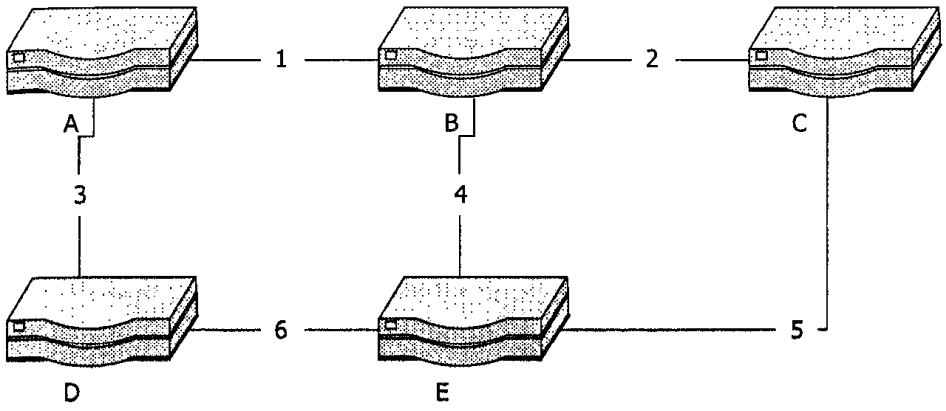


Рис. 7.1. Пример сети из 5 маршрутизаторов

Отметим, что эта конфигурация сети выбрана только с целью пояснения работы протокола, но, опираясь на теорию и полученную практику, можно внедрить протокол RIP практически в любой среде, учитывая только его ограничения. Для рассмотрения принципов действия не будем обращать внимания как на типы каналов, связывающих маршрутизаторы, так и на сами маршрутизаторы. В этой сети может работать оборудование компании Cisco Systems или, например, это могут быть программные маршрутизаторы на базе Windows NT/2000.

В реальной сети каждый маршрутизатор будет идентифицирован с помощью IP-адреса и с помощью имени (первое, естественно, важнее с точки зрения сетевой идентификации). Для того чтобы далее при рассмотрении примера было проще ссылаться на конкретные устройства, присвоим всем маршрутизаторам символьные обозначения (A, B, C, D и E), которые в реальной ситуации и будут их адресами. Кроме того, каждый канал связи между маршрутизаторами также получает свой номер (1–6).

Каждому маршрутизатору при начальном запуске необходимо определить собственный адрес и подключенные к нему каналы связи. Эту информацию можно охарактеризовать как локальную, так как в данный момент времени маршрутизаторы знают только о своем внутреннем состоянии и полностью игнорируют окружающую их сетевую топологию. Маршрутизатор «не знает» о том, сколько помимо него существует в сети маршрутизаторов. У него нет даже информации о том, какое устройство расположено на другом конце канала связи. В этом начальном состоянии работы маршрутизатора его таблица маршрутизации содержит только одну запись — запись о самом себе. Таким образом, таблица маршрутизации маршрутизатора A будет следующей.

От A к	Канал	Стоимость
A	Локальный	0

Обратим внимание на третью колонку таблицы (стоимость), поскольку да
по тексту мы будем постоянно пользоваться понятием стоимости маршру
Длина вектора A = 0, или, другими словами, первоначальная стоимость марш
та равна нулю. После того как маршрутизатор A загрузился и привел в дейст
протокол RIP, он начинает рассылать эту таблицу всем своим соседям, или,
более точно, он инициирует рассылку на все подключенные к нему каналы с
зи (интерфейсы). В результате оба маршрутизатора B и D получат эту инф
мацию и пополнят свои знания о сетевой топологии.

Рассмотрим теперь маршрутизатор B. До получения сообщения от марш
тизатора A его таблица маршрутизации также состояла из одной записи.

От B к	Канал	Стоимость
B	Локальный	0

Маршрутизатор B получает через канал с номером 1 стоимость A = 0. По
получения этого сообщения он обновляет свои записи, добавляя к ним стоимо
локального канала (на данном этапе предположим, что она равна 1), преобра
полученное сообщение в A = 1. После обновления стоимости маршрутизато
проверит свою таблицу маршрутизации и узнав, что она не содержит инфор
ции о маршрутизаторе A, добавит одну запись.

От B к	Канал	Стоимость
B	Локальный	0
A	1	1

Затем маршрутизатор B сформирует свою собственную запись о стоимос
маршрутов (B = 0, A = 1) и пошлет ее через подключенные к нему каналы с
мерами 1, 4 и 2. Так как маршрутизатор A отправил свою таблицу не тол
маршрутизатору B, но и маршрутизатору D, то последний также обновит св
таблицу маршрутизации и разошлет обновленную запись о стоимости марш
тов (D = 0, A = 1) через каналы 3 и 6.

От D к	Канал	Стоимость
D	Локальный	0
A	3	1

Сообщение от маршрутизатора B получат маршрутизаторы A, C и E. А
маршрутизатора D — маршрутизаторы A и E. Для того чтобы упростить изл
жение, предположим, что сообщение от маршрутизатора B поступило первы
Приняв это сообщение, маршрутизатор A обновит свою запись и устано
стоимости маршрутов равными B = 1, A = 2. Обнаружив стоимость маршрута
равную 2, что больше, чем значение в локальной таблице маршрутизации, мар
рутизатор A проигнорирует эту информацию и добавит в таблицу только зап
B = 1. Затем, приняв сообщение от маршрутизатора D, он также присоеди
запись D = 1, формируя следующую таблицу.

От A к	Канал	Стоимость
A	Локальный	0
B	1	1
D	3	1

Маршрутизатор C получит запись о стоимостях B = 0, A = 1 через канал 2 и обновит свою таблицу следующим образом.

От C к	Канал	Стоимость
C	Локальный	0
B	2	1
A	2	2

Маршрутизатор E также получит запись B = 0, A = 1 через канал 4 и обновит свою таблицу.

От E к	Канал	Стоимость
E	Локальный	0
B	4	1
A	4	2

Кроме того, маршрутизатор E затем получит запись D = 0, A = 1 через канал с номером 6, обновит стоимости маршрутов D = 1, A = 2 и затем обнаружит, что информация о маршрутизаторе D отсутствует в его таблице маршрутизации и, следовательно, должна быть вставлена. Расстояние до маршрутизатора A через канал 6 равна расстоянию до этого маршрутизатора через канал 4. Предположим, что в таком случае маршрутизатор воздержится от обновления своей таблицы. Результирующая таблица маршрутизации примет следующий вид.

От E к	Канал	Стоимость
E	Локальный	0
B	4	1
A	4	2
D	6	1

После того как маршрутизаторы A, C и E сформируют свои локальные таблицы маршрутизации, они обобщат информацию и вычислят новые стоимости маршрутов, которые вновь будут посланы через подключенные к ним каналы связи.

- от A через каналы 1 и 3: A = 0, B = 1, D = 1;
- от C через каналы 2 и 5: C = 0, B = 1, A = 2;
- от E через каналы 4, 5 и 6: E = 0, B = 1, A = 2, D = 1.

Получение соседними маршрутизаторами новой информации вызовет ф мирование следующих таблиц.

От B к	Канал	Стоимость
B	Локальный	0
A	1	1
D	1	2
C	2	1
E	4	1

От D к	Канал	Стоимость
D	Локальный	0
A	3	1
B	3	2
E	6	1

От E к	Канал	Стоимость
E	Локальный	0
B	4	1
A	4	2
D	6	1
C	5	1

Маршрутизаторы B, D и E теперь могут подсчитать свои собственные ст мости:

- от B через каналы 1, 2 и 4: B = 0, A = 1, D = 2, E = 1;
- от D через каналы 3 и 6: D = 0, A = 1, B = 2, E = 1;
- от E через каналы 4, 5 и 6: E = 0, B = 1, A = 2, D = 1.

Эти стоимости будут разосланы в сообщениях, заставляя маршрутизаторы C и D обновить свои таблицы.

От A к	Канал	Стоимость
A	Локальный	0
B	1	1
D	3	1
C	1	2
E	1	2

От C к	Канал	Стоимость
C	Локальный	0
B	2	1
A	2	2
E	5	1
D	5	2

От D к	Канал	Стоимость
D	Локальный	0
A	3	1
B	3	2
E	6	1
C	6	2

На этой стадии можно говорить о том, что наступил момент схождения, так как несмотря на то что маршрутизаторы вновь подготовят свои стоимости маршрутов и разошлют их соседям в своих сообщениях, это не повлечет за собой каких-либо изменений в таблицах маршрутизации. Фактически, все маршрутизаторы в рассматриваемой распределенной сети определили сетевую топологию.

После того как маршрутизаторы построили свои таблицы, они знают маршруты во все сети (если отойти от введенных обозначений и возвратиться к адресной схеме протокола IP). Рассматриваемая сеть предполагает некоторую избыточность. Так, например, от маршрутизатора E к маршрутизатору A ведут два разных пути. В реальной же жизни такое бывает нечасто (гораздо чаще встречается противоположная ситуация — избыточность каналов связи отсутствует). Поэтому очень важно рассмотреть такой крайний случай, как поведение протокола маршрутизации RIP при обрыве канала связи. Предположим, что произошел обрыв канала с номером 1. В результате сетевая топология примет вид, показанный на рис. 7.2.

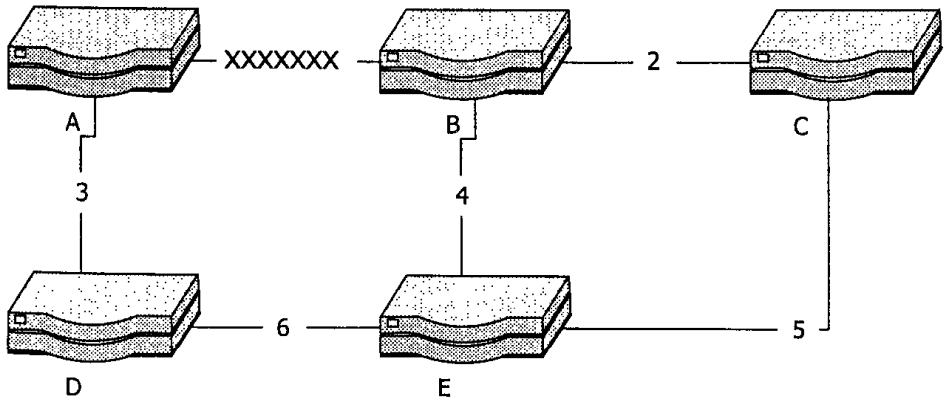


Рис. 7.2. Обрыв канала связи

Допустим, что маршрутизаторы A и B обнаруживают возникшие с кана связи 1 проблемы. Это возможно, если, например, отслеживается работоспо ность подключенного сетевого оборудования. В этом случае они должны не ленно обновить свои таблицы маршрутизации с учетом того, что канал 1 и теперь стоимость, равную «бесконечности», и что все другие маршрутизато до того достижимые через канал 1, теперь недоступны. Новые таблицы марш тизации примут следующий вид.

От A к	Канал	Стоимость
A	Локальный	0
B	1	Inf (∞)
D	3	1
C	1	Inf (∞)
E	1	Inf (∞)

От B к	Канал	Стоимость
B	Локальный	0
A	1	Inf (∞)
D	1	Inf (∞)
C	2	1
E	4	1

Маршрутизаторы A и B подготовят и пошлют свои новые стоимости:

- от A через канал 3: A = 0, B = Inf, D = 1, C = Inf, E = Inf;
- от B через канал 2 и 4: B = 0, A = Inf, D = Inf, C = 1, E = 1.

Сообщение от маршрутизатора A будет получено маршрутизатором D торый обновит стоимости маршрутов, принимая во внимание стоимость ка связи с номером 3: A = 1, B = Inf, D = 2, C = Inf, E = Inf. Эта запись будет нена с записями в таблице маршрутизации. Маршрутизатор D обнаружит, все стоимости больше или равны значений в соответствующих записях лицы. Исключение составит канал 3, только через который и можно дост маршрутизатора B, и поэтому запись для маршрутизатора B должна быть новлена. Новая таблица примет такой вид.

От D к	Канал	Стоимость
D	Локальный	0
A	3	1
B	3	Inf (∞)
E	6	1
C	6	2

Аналогично, маршрутизаторы C и E обновят свои таблицы.

От C к	Канал	Стоимость
C	Локальный	0
B	2	1
A	2	Inf (∞)
E	5	1
D	5	2

От E к	Канал	Стоимость
E	Локальный	0
B	4	1
A	4	Inf (∞)
D	6	1
C	5	1

Эти три маршрутизатора подготовят и вышлют свои новые стоимости маршрутов:

- от D через каналы 3 и 6: D = 0, A = 1, B = Inf, E = 1, C = 2;
- от C через каналы 2 и 5: C = 0, B = 1, A = Inf, E = 1, D = 2;
- от E через каналы 4, 5 и 6: E = 0, B = 1, A = Inf, D = 1, C = 1.

Сообщения, содержащие информацию о стоимости маршрутов, приведут к обновлению таблиц маршрутизации на маршрутизаторах A, B, D и E:

От A к	Канал	Стоимость
A	Локальный	0
B	1	Inf (∞)
D	3	1
C	3	3
E	3	2

От B к	Канал	Стоимость
B	Локальный	0
A	1	Inf (∞)
D	4	2
C	2	1
E	4	1

От D к	Канал	Стоимость
D	Локальный	0
A	3	1
B	6	2
E	6	1
C	6	2

От E к	Канал	Стоимость
E	Локальный	0
B	4	1
A	6	2
D	6	1
C	5	1

Получив очередные сообщения, маршрутизаторы A, B, D и E вышлют с новые стоимости маршрутов:

- от A через канал 3: A = 0, B = Inf, D = 1, C = 3, E = 2;
- от B через канал 2 и 4: B = 0, A = Inf, D = 2, C = 1, E = 1;
- от D через каналы 3 и 6: D = 0, A = 1, B = 2, E = 1, C = 2;
- от E через каналы 4, 5 и 6: E = 0, B = 1, A = 2, D = 1, C = 1.

Эти сообщения приведут к обновлению таблиц маршрутизации.

От A к	Канал	Стоимость
A	Локальный	0
B	3	3
D	3	1
C	3	3
E	3	2

От B к	Канал	Стоимость
B	Локальный	0
A	4	3
D	4	2
C	2	1
E	4	1

От С к	Канал	Стоимость
C	Локальный	0
B	2	1
A	5	3
E	5	1
D	5	2

Маршрутизаторы A, B и C подготовят и вышлют свои собственные стоимости маршрутов, но это не приведет к изменению таблиц. В результате новые маршруты полностью определены и общая связь в распределенной сети восстановлена.

Возвратимся к теме избыточных (резервных) каналов связи. Теоретически ничто не мешает их построению, однако в реальности администратор столкнется со множеством разнообразнейших факторов, мешающих реализации подобной идеи по месту. Прежде всего, сегодня у нас практически невозможно детально спланировать распределенную сеть с учетом всех каналов связи, их скорости и т. п. Этот тезис вступает в противоречие с многочисленными утверждениями англоязычных авторов о том, что сеть с самого начала надо планировать, и планировать как можно детальнее. Не исключено, что, например, в США или в Европе создание и практическая реализация подробного плана распределенной сети вполне осуществима, однако сегодня в России — сомнительно. Ситуация, когда крупная компания заключает договор с провайдером услуг связи и в дальнейшем продолжает работу только с ним, может быть отнесена, скорее, к исключительной. Но спрашивается: все ли готовы платить высокую арендную плату за такого рода услуги? Вот и получаются ситуации, когда сеть в конечном итоге начинает походить на некую «окрошку» из выделенных каналов связи, технологий Frame Relay, ISDN и т. д. Причем, что характерно, под влиянием внешних факторов технологии постоянно вытесняют, заменяют друг друга. В ход идет все, что дешевле и, соответственно, выгодней компании. То же говорит и собственный опыт управления и развития сети, первоначально построенной только с использованием каналов Frame Relay, — по прошествии достаточно короткого промежутка времени остро встал вопрос о повышении скоростей передач. На взгляд западного специалиста естественным решением может быть простая заявка своему провайдеру на повышение скорости. У нас это дорого. Поэтому мы «пошли другим путем» и попробовали своими силами организовать связь объектов, используя прямые провода и модемы серии xDSL. Работает. Однако если в технологии Frame Relay соединения являются виртуальными и задействуют только один физический интерфейс, то для подключения модемов xDSL требуется отдельный интерфейс и, следовательно, при этом частично меняется схема сети. Таким образом, вопрос избыточности обычно упирается в технические возможности и готовность платить деньги.

ИМЕЧАНИЕ

Технология xDSL (Digital Subscriber Line) — это обобщенное название группы технологий. Все тех логии этой группы делятся в основном по способу разделения встречных потоков данных (част ное разделение потоков данных, метод эхоподавления и т. д.). При этом встречные потоки мо быть как симметричными, так и асимметричными. Технология xDSL позволяет передавать циф вой сигнал по телефонной сети общего назначения.

Для этого на концах абонентской линии устанавливаются разделительные фильтры. Фильтры пр назначены для отделения высокочастотной составляющей, на которой работают модемы XDSL, низкочастотной составляющей (до 3,5 кГц) для телефонного оборудования общего назначения. С модемов xDSL на высокой частоте позволяет им значительно превысить скоростной предел рабо обычных модемов, который составляет 56 Кбит/с. Например, технология ADSL (Asymmetric Dig Subscriber Line — асимметричный DSL) позволяет поддерживать скорость к абоненту до 6,1 Мби а от абонента — до 640 Кбит/с. Кроме того, технология xDSL позволяет одновременно переда по одной и той же телефонной линии цифровые данные и вести обычные телефонные перегово Использование обычных модемов не позволяет делать этого. Установка модемов технологии x на абонентской стороне ничем не отличается от установки обычных модемов. На территории провайдер развертывает оборудование, соединяющее абонента с базовой сетью передачи данны Более широкую полосу пропускания можно получить при использовании выделенных линий пример, медной пары между объектами, перекрещенной на промежуточных АТС). Для этого суш ствуют специальные модемы, которые, работая на больших расстояниях, обеспечивают скоро до 2 048 Кбит/с. Эти модемы могут взаимодействовать как в синхронном, так и в асинхронном жимах. Дальность передачи зависит от качества физической линии, в частности от диаметра се ния проводов, от количества проводников в линии и от скорости передачи.

Для простоты изложения стоимость всех каналов связи была равна едини На практике все сложнее: некоторые каналы медленнее других или, наприм их использование дороже в денежном выражении. Предположим, что стоимос канала с номером 5 равна 10, в то время как стоимость всех остальных канал остается равной 1.

Рассмотрим маршруты от разных частей сети к маршрутизатору C. В стаби но работающей сети после успешного запуска всех маршрутизаторов будут вестны следующие маршруты.

Направление	Канал	Стоимость
A — C	1	2
B — C	2	1
D — C	3	3
E — C	4	2

Теперь допустим, что канал связи с номером 2 становится недоступны Этот сбой немедленно обнаруживает маршрутизатор B, и через некоторое вре маршруты к маршрутизатору C будут выглядеть так.

Направление	Канал	Стоимость
A — C	1	2
B — C	2	Inf
D — C	3	3
E — C	4	2

Предположим, что перед тем как маршрутизатор B отправит свою запись о стоимостях своим соседям, маршрутизатор A посылает свою таблицу маршрутизаторам B и D. Большинство реализаций протоколов, использующих алгоритм длины вектора, подразумевают, что сообщения должны посылаться регулярно через определенные интервалы времени. Это нужно для того, чтобы справиться с потерями пакетов в сети в случае возникновения непредвиденных ситуаций. Такая ложная передача вполне может иметь место. Это сообщение маршрутизатора A не существенно для маршрутизатора D, так как указываемые стоимости маршрутов уже находятся в его таблице. Но маршрутизатор B добавит стоимость, равную 1, к стоимости 2, рассылаемой маршрутизатором A, и поймет, что конечная стоимость, равная 3, меньше, чем стоимость, равная бесконечности, которая была записана в его таблице. В результате он обновит таблицу с указанием заведомо ложной стоимости маршрута к маршрутизатору C. Новая таблица указывает на то, что маршрутизатор C стал достижим через канал связи 1 со стоимостью 3. Теперь маршрутизатор B начнет рекламировать новую запись своим соседям — маршрутизаторам A и E. Эти сообщения поступят к ним через каналы с номерами 1 и 4, и маршрутизаторы обновят свои таблицы следующим образом.

Направление	Канал	Стоимость
A — C	1	4
B — C	1	3
D — C	3	3
E — C	4	4

Следует обратить внимание на то, что с текущего момента таблица маршрутизации содержит петлю (loop) — пакеты от маршрутизатора C будут достигать маршрутизатора B, затем «отскакивать» от него и двигаться между A и B до тех пор, пока время их жизни в сети не кончится. Это явление называется эффектом прыжка (bouncing effect). Оно прекратится только тогда, когда сеть перейдет в стабильное состояние (сойдется) и стоимости маршрутов будут правильно отображаться в таблицах маршрутизации.

Если в этот момент маршрутизатор C известит о своей стоимости маршрутов через канал связи 5, то маршрутизатор E добавит стоимость 10 к входящей стоимости, равной 0. При этом он обнаружит, что итоговая стоимость в таблице маршрутизации превышает текущую, и проигнорирует полученные сообщения. Теперь, если маршрутизаторы A и E разошлют свои новые стоимости маршрутов, это приведет к обновлению таблицы маршрутизации на B и D.

Направление	Канал	Стоимость
A — C	1	4
B — C	1	5
D — C	3	5
E — C	4	4

После того как маршрутизатор B передаст свою собственную запись о стои стях, стоимость маршрута к маршрутизатору C на маршрутизаторе A увелич до 6. Маршрутизатор A будет рекламировать новую стоимость маршрутов к D

Направление	Канал	Стоимость
A — C	1	6
B — C	1	7
D — C	3	7
E — C	4	6

На этой стадии сообщения, посылаемые маршрутизатором C, будут игн роваться маршрутизатором E, так как стоимость маршрута к C на нем все меньше 10. На самом деле при каждом обмене маршрутами стоимость марш до C будет увеличиваться на 2 на каждом из маршрутизаторов A, B, D и E. сле следующего «раунда» обмена таблица преобразуется к следующей.

Направление	Канал	Стоимость
A — C	1	8
B — C	1	9
D — C	3	9
E — C	4	8

Очередное сообщение от маршрутизатора B приведет к обновлению запи о стоимости маршрута на маршрутизаторе A. Она станет равной 10. Марш тизатор A перешлет новое значение маршрутизаторам B и D, которые увели свои стоимости до 11. Затем маршрутизатор B передаст значение стоимости, р ное 11, маршрутизаторам A и E. Текущая картина такова.

Направление	Канал	Стоимость
A — C	1	12
B — C	1	11
D — C	3	11
E — C	4	12

Теперь уже маршрутизатор E, получая сообщение от маршрутизатора C, п мет его во внимание и вычислит новые значения.

Направление	Канал	Стоимость
A — C	1	12
B — C	4	11
D — C	6	11
E — C	5	10

Наконец можно сказать, что все вычисления закончены. На постепенное увеличение стоимостей маршрутов потребовалось несколько шагов. Учитывая, что один раунд обмена увеличивает стоимость маршрута на 2 единицы, нам потребуется 5 таких раундов для достижения значения стоимости, равного 10, которое необходимо для перехода к следующему шагу. Следует отметить, что это случайный процесс, так как результат может варьироваться в зависимости от порядка передачи сообщений маршрутизаторами. Также важным обстоятельством является то, что промежуточная стадия может оказаться очень «болезненной» для сети. Это связано с тем, что при образовании петель может возникнуть перегрузка каналов связи. Кроме того, в этот период вполне вероятна потеря пакетов, в том числе пакетов, содержащих информацию о новых стоимостях маршрутов, что также приведет к задержке достижения сходимости.

Счет до бесконечности

Предположим, что в нашей распределенной сети произошел обрыв двух каналов связи — 1 и 6. При этом маршрутизаторы A и D окажутся изолированными от остальных маршрутизаторов в сети. Остановимся на процессах, происходящих на маршрутизаторах A и D. Последний после обнаружения обрыва обновил свою таблицу маршрутизации следующим образом.

От D к	Канал	Стоимость
D	Локальный	0
A	3	1
B	6	Inf
E	6	Inf
C	6	Inf

Если маршрутизатор D получит возможность немедленной передачи своих стоимостей маршрутов, то маршрутизатор A по поступлении сообщения от маршрутизатора D обновит свою таблицу и обнаружит, что все маршрутизаторы, за исключением D, в настоящий момент недостижимы. Однако если A передаст раньше свои последние стоимости маршрутов (A = 0, B = 3, D = 3, C = 3, E = 3), то таблица маршрутизации на маршрутизаторе D примет следующий вид.

От D к	Канал	Стоимость
D	Локальный	0
A	3	1
B	3	4
E	3	4
C	3	4

В результате образуется логическая петля, как и в предыдущем примере с фектом скачка. Однако так как маршрутизаторы B, C и E теперь изолирова то эта сеть не имеет шансов самостоятельно перейти в стабильное состоя На каждом шаге обмена сообщениями стоимости маршрутов к маршрутиз рам B, C и E увеличиваются на 2 единицы. Такой процесс называется счето бесконечности (counting to infinity), и он может быть остановлен только пред рительным соглашением о том, каким значением будет представляться беско ность. Когда стоимости достигнут этого значения, записи в таблицах маршрут ции будут восприниматься как записи, указывающие на бесконечность, и та маршруты будут считаться недостижимыми. Как и в предыдущем примере, добный постоянный обмен может привести к повышению загрузки каналов с зи и возможным потерям пакетов.

Расщепление горизонта

Возможность появления эффекта счета до бесконечности и эффекта скачка как результат, резкое снижение скорости сходимости являются серьезными достатками протоколов длины вектора. Для решения этих проблем были раз ботаны специальные технологии, среди которых выделяются технологии расщ пления горизонта (английский термин split horizon — в книге используе один из возможных переводов) и мгновенного обновления (triggered updat реализованные в протоколе RIP.

Механизм расщепления горизонта опирается на очень простое правило: е маршрутизатор A передает пакеты по адресу некоторого хоста и эти пакеты п ходят через маршрутизатор B, маршрутизатору B нет смысла пытаться иск маршрут к данному хосту через маршрутизатор A. При введении этого прав необходимые изменения в оригинальной версии протокола минимальны: вме того, чтобы рассылать одну и ту же стоимость маршрута через все свои инт фейсы, маршрутизаторы будут посылать разные версии этого сообщения. В можны два варианта работы. В первом (простейшем) маршрутизатор просто ключает из своих сообщений всякую информацию о получателях, достижим через канал связи. Такой подход основан на том, что маршруты никогда анонсируются, или на том, что старые сведения удаляются с помощью механ ма тайм-аута. Второй вариант называется split horizon with poisonous reve (можно перевести как «расщепление горизонта с обратным исправлением и он более агрессивен: маршрутизаторы будут включать маршруты до всех лучателей в рассылаемые сообщения, но станут устанавливать расстояния некоторых получателей в бесконечность. В результате логические петли меж двумя маршрутизаторами будут удаляться немедленно. Можно показать на п стом примере, что подобная стратегия способна устранить проблему счета бесконечности. Однако эта технология не защищает от всех возможных спо бов образования петель.

Чтобы проиллюстрировать сказанное, вернемся к рассмотрению ситуа в сети с обрывом двух каналов связи. Сразу после обрыва канала связи меж маршрутизаторами E и D (канал 6) таблицы маршрутизации на B, C и E бу иметь следующий вид.

Направление	Канал	Стоимость
B — D	4	2
C — D	5	2
E — D	6	Inf

Маршрутизатор E обнаруживает обрыв канала и посылает сообщения через каналы 4 и 5, указывая на то, что расстояние до маршрутизатора D в настоящий момент равно бесконечности. Предположим, что это сообщение получает маршрутизатор B, но далее, например по причине сбоя, оно не достигает маршрутизатора C. Теперь записи в таблицах следующие.

Направление	Канал	Стоимость
B — D	4	Inf
C — D	5	2
E — D	6	Inf

Теперь представим, что маршрутизатору C пришло время рассылать свои сообщения об обновлении. Маршрутизатор C будет действовать совершенно иначе: рассылать сообщение о бесконечности расстояния до маршрутизатора E через канал 5. А через канал 2 он будет отправлять сообщение о стоимости маршрута, равной 2. В результате маршрутизатор B обновит свою таблицу маршрутизации и затем разошлет своим соседям сообщения о бесконечности расстояния через канал 2 и о стоимости маршрута, равной 3, через канал 4. Записи в таблицах маршрутизации примут вид.

Направление	Канал	Стоимость
B — D	2	3
C — D	5	2
E — D	4	4

Как видно, существует петля между маршрутизаторами C, B и E. Легко показать, что следующий обмен сообщениями об обновлении приведет к постепенному увеличению расстояний без изменения инфраструктуры маршрутизации, что возвращает нас к проблеме счета до бесконечности.

Мгновенное обновление

Во всех приведенных ранее примерах был успешно обойден один очень важный вопрос: когда следует посылать сообщения о доступных маршрутах своим соседям. Решение этого вопроса на самом деле зависит от следующих факторов: своевременности информации, необходимости ожидания полного набора сообщений об обновлении до того момента, как будет принято решение, сопротивляемости потерям пакетов, отслеживания доступности соседей и т. п.

Некоторые реализации протоколов длины вектора, в том числе и прото
RIP, полагаются на регулярную отправку сообщений об обновлении. Это не
ходимо как для мониторинга соседей, так и для корректировки случаев пот
пакетов. С каждой записью в таблице маршрутизации ассоциируется тайм
Если информация не обновлена по получении нового сообщения до истече
времени этого таймера, то маршрутизатор полагает, что сосед недоступен, и бу
считать расстояние до получателя равным бесконечности. Таким образом, з
чение таймера должно быть выбрано так, чтобы оно превышало время перед
информации от одного маршрутизатора к другому. Кроме того, было бы нераз
ным помечать некий маршрут как недостижимый только после потери одн
сообщения. Например, протокол RIP полагает, что таймер должен быть устан
лен в шестикратное значение интервала передач сообщений. Поэтому маршр
в таблице маршрутизации остается рабочим, если по крайней мере одно из ше
сообщений было получено. На самом деле маршрутизатору необходимо при
мать во внимание загруженность сети, обусловленную повторением своих со
щений, и этот период не может быть слишком коротким.

Большой интервал между сообщениями, однако, имеет негативное влия
если изменение в сети произошло сразу после посылки сообщений, то маршру
затору все равно придется ждать окончания промежутка до момента получен
извещения о произошедших изменениях у своих соседей. Механизм «мгнов
ного обновления» пытается ускорить реакцию протокола за счет принудите
ной рассылки маршрутизаторами сообщений в самое короткое время с моме
обнаружения изменений в своих таблицах маршрутизации без ожидания око
чания периода рассылки сообщений. Этот механизм значительно повышает с
рость сходимости сети. В документации на протокол RIP (RFC 1058) сказа
что немедленная рассылка информации приведет к тому, что маршрутиза
будет поддерживать актуальность информации, препятствуя тем самым обра
ванию петель маршрутизации. Очевидно, что негативные эффекты работы т
токола длины вектора вызваны именно ложными сообщениями, которые про
не говорят ничего нового о состоянии своих таблиц на текущий период вре
ни. Тем не менее существуют ситуации, особенно обусловленные потерями
общений о маршрутизации в сети, когда даже использование рассматриваем
механизмов неспособно предотвратить возникновение логических петель мар
рутизации. Такие ситуации будут разрешаться методом счета до бесконечнос
Однако даже и в последнем случае механизм мгновенного обновления спосо
несколько смягчить ситуацию, так как потребуется значительно меньше вре
ни для работы механизма счета до бесконечности при условии, конечно, что
вые сообщения посылаются немедленно, без задержки.

Уже отмечалось, что протокол RIP изначально был разработан в качес
одного из компонентов для операционной системы UNIX BSD и реализов
в виде программы routed. В своей первой версии он был достаточно прост в ра
те и, соответственно, в реализации. Протокол использовался в различных п
дуктах и был документирован в RFC 1058 в 1988 году (автор Чарльз Хедр
Этот документ внес некоторые улучшения в начальную реализацию проток
К улучшениям, в частности, можно отнести технологии расщепления гориз
и мгновенного обновления.

Протоколы семейства длины вектора отличаются друг от друга множеством факторов. Например, различия проявляются в используемых метриках, структуре адресации и т. п. Протокол RIP относится к классу IGP и призван работать внутри автономной системы для сетей относительно небольшого размера.

Запись в таблице маршрутизации для протокола RIP может соответствовать хосту, сети или подсети. Для этого протокола нет понятия типа адреса — ему требуется провести анализ адреса, чтобы получить информацию о методе работы с ним. Сначала он отделяет сетевую часть адреса протокола IP от части «подсеть + хост» посредством анализа класса адреса (A, B или C). Если последняя часть равна нулю, то запись представляет сеть, иначе она представляет либо подсеть, либо хост. Для четкого определения содержимого последней части маршрутизатору необходимо знать маску подсети. Если после применения маски часть адреса хоста равна нулю, то это адрес подсети.

По умолчанию протокол RIP использует очень простую метрику маршрута: расстояние равно количеству каналов связи в пути до получателя (другими словами, количеству переходов). Это расстояние выражается числом в интервале между 1 и 15. Значение 16 соответствует бесконечности. В документе RFC 1058 присутствует информация о том, как можно применить более сложную метрику для выделения предпочтения тому или иному каналу связи. Но при этом значительно возрастает сложность реализации с учетом выбранного ограничения по количеству переходов. С другой стороны, выбор большего значения продлит состояние несвязности в маршрутах при выполнении счета до бесконечности, что достаточно опасно в рабочей сети.

Протокол RIP может работать как на каналах типа «точка-точка», так и в широковещательных сетях (например, Ethernet). Для передачи своих сообщений (как для получения, так и для отправки) протокол RIP опирается на протокол UDP (схема инкапсуляции RIP → UDP → IP) с номером порта 520. По умолчанию сообщения рассылаются широковещательно, и в результате они будут доступны всем маршрутизаторам, подключенным к этой широковещательной сети. В нормальном состоянии пакеты рассылаются каждые 30 секунд. В случае включения механизма мгновенного обновления это происходит быстрее. Если запись в таблице маршрутизации (маршрут) не обновляется в течение 180 секунд, расстояние устанавливается равным бесконечности и эта запись удаляется из таблицы.

Каждая запись в таблице маршрутизации содержит по крайней мере следующие значения:

- адрес получателя;
- метрика, ассоциированную с этим получателем;
- адрес следующего получателя в пути до достижения конечного;
- другие таймеры.

При получении сообщения маршрутизатор по очереди проверяет записи в таблице маршрутизации. Он убеждается, например, в том, что это адрес корректного класса, что это не сеть 127 или 0 (за исключением маршрута по умолчанию 0.0.0.0), далее, что часть адреса хоста не является широковещательным адресом и что метрика не «превышает бесконечность». Некорректные записи игнори-

руются. Документ RFC 1058 рекомендует рассматривать их наличие как фа ошибки. Если в сообщении метрика не равна бесконечности, то она увеличи ется на единицу для учета перехода. Затем таблица проверяется на присутств записи, относящейся к этому получателю, и выполняется механизм генера длины вектора.

Сообщения посылаются регулярно каждые 30 секунд или при каждом о новлении таблицы. В реальной ситуации мгновенные обновления могут вызв интенсивный сетевой трафик. В связи с этим документ RFC 1058 определя набор мер предосторожности для снижения трафика. Сообщения не должны сылаться немедленно после получения обновлений о маршрутах. Они долж быть отосланы через некоторый небольшой случайный промежуток време длительностью от 1 до 5 секунд. Это позволяет среагировать на информацию обновлениях, получаемую от других соседей, и обобщить ее в одном следующ сообщении. Такое поведение положительно, хотя и немного, влияет на снижен нагрузки в сети. Кроме того, администратор сети может ограничить скорос с которой мгновенные обновления будут посылаться через выбранный канал

Сообщения, подготавливаемые для всех подключенных интерфейсов, мо отличаться друг от друга. По причине, например, работы механизма расщеп ния горизонта, а также в случае обобщения подсетей. Сообщения обычно соде жат адреса и метрику для всех записей в таблице. Однако когда сообщение сылается в качестве мгновенного обновления, нет необходимости дублиров все записи в таблице — достаточно включить в нее только те, которые изме лись. Максимальный размер сообщения выбран равным 512 байт, что позвол передать до 25 записей. Адрес отправителя сообщения должен быть всегда ра IP-адресу интерфейса, через который посылается сообщение.

Записи о подсетях следует анонсировать только в тех случаях, когда терфейс принадлежит к той же сети, что и подсеть. Для других интерфейсов записи о подсетях должны быть обобщены в одну сетевую запись. Записи с б конечной метрикой необходимо анонсировать только тогда, когда они были давно обновлены. В таком случае нет опасности их постепенного исчезнове из таблиц маршрутизации соседей по истечении времени таймеров, а это все выгодно с точки зрения снижения нагрузки. То же правило применимо к за сям, которые имеют бесконечную метрику, когда работает механизм расщеп ния горизонта — они могут быть проигнорированы, если информация о след щем маршрутизаторе в пути не была обновлена.

Помимо посылки сообщений протокол RIP может рассылать запросы, ко рые обычно используются при начальном запуске маршрутизатора. Эти запро необходимы для получения информации от соседей при построении табл маршрутизации. Существуют всего два типа запроса: один для получения таблицы, второй для получения только определенных записей. В первом сл в запросе указывается адрес 0.0.0.0 с метрикой, равной бесконечности, и от будут такие же, как при нормальной работе, включая обработку результато тода расщепления горизонта. Запросы второго типа обычно не нужны при мальной работе, а используются в процессе отладки.

В архитектуре сети Интернет существует четкое разграничение между хостами и маршрутизаторами. Первые обычно не выполняют функции маршрутизации, хотя могут иметь несколько сетевых интерфейсов. Но в любом случае маршрутизация — «дело рук» самого маршрутизатора. В результате только маршрутизаторы должны «осознавать» присутствие протокола RIP, но это в идеальном случае. На самом деле ввиду того, что сам протокол не новый — его история исчисляется уже десятилетием и захватывает период времени, когда не было четкого различия между хостами и маршрутизаторами, — протокол грешит некоторой «неразборчивостью». В связи с этим разработчики предположили, что достаточно удачным решением будет рассылать информацию протокола широковещательно. В этом случае хосты могут отслеживать сообщения протокола и формировать на их основе свои собственные таблицы маршрутизации. Напомним, что таблица маршрутизации есть даже у простого компьютера с единственным сетевым адаптером. Кроме того, хосты с несколькими сетевыми картами смогут принимать на основе этой информации решения о выборе маршрута для некоторого получателя в случае, когда несколько маршрутизаторов подключены к одной локальной сети. Подобные хосты способны посылать запросы на получение информации, но они никогда не отвечают собственными сообщениями. По этой причине они были описаны в документе RFC 1058 как «молчаливые» (silent hosts).

Такая стратегия работает по крайней мере при использовании протокола RIP первой версии (RIP-1). Однако, учитывая в настоящее время доступность и других протоколов (RIP-2, IGRP и т. п.), появилось опасение, что подобные хосты не сумеют поддерживать корректные таблицы маршрутизации. Поэтому в сообществе Интернета выработано единогласное решение против применения подобных «молчаливых» хостов. Нормальная работа хоста заключается в посылке пакетов удаленному получателю с помощью маршрута по умолчанию.

Деятельность протокола RIP на маршрутизаторе всегда подразумевает наличие некоторых конфигурационных этапов. В самом простом варианте протокол изучает список локальных интерфейсов маршрутизатора, соответствующие им адреса и маски подсетей. При отсутствии другой информации протокол будет инициализировать таблицу маршрутизации только с одной записью для каждого интерфейса и с метрикой, равной 1 (устройства на этих подсетях расположены на расстоянии одного перехода относительно локального маршрутизатора). По итогам обследования протокол будет рассылать запросы своим соседям для формирования таблиц и начала полноценной работы.

Однако существуют причины, по которым администратор сети вправе отказаться от широковещательной рассылки сообщений этого протокола на все локальные интерфейсы. Например, может быть только один маршрутизатор на всю подсеть. В таком случае широковещательная рассылка в эту подсеть приведет к ненужному расходованию ресурсов. Может, например, возникнуть необходимость в поддержке иных протоколов маршрутизации на других интерфейсах. Естественно, нужно учитывать и такой фактор, как безопасность — рассылка сообщений протокола в состоянии раскрыть адресную структуру всей распределенной сети. Кроме того, если используется технология dial-on-demand (например, для каналов ISDN), то регулярные сообщения протокола будут заставлять маршрутизатор устанавливать соединение, что приведет к лишним затратам.

Учитывая это, администратор может назначать выбранные интерфейсы марш тизатора в качестве пассивных (passive interface) с точки зрения протокола R В качестве примера рассмотрим работу протокола в сети (рис. 7.3).

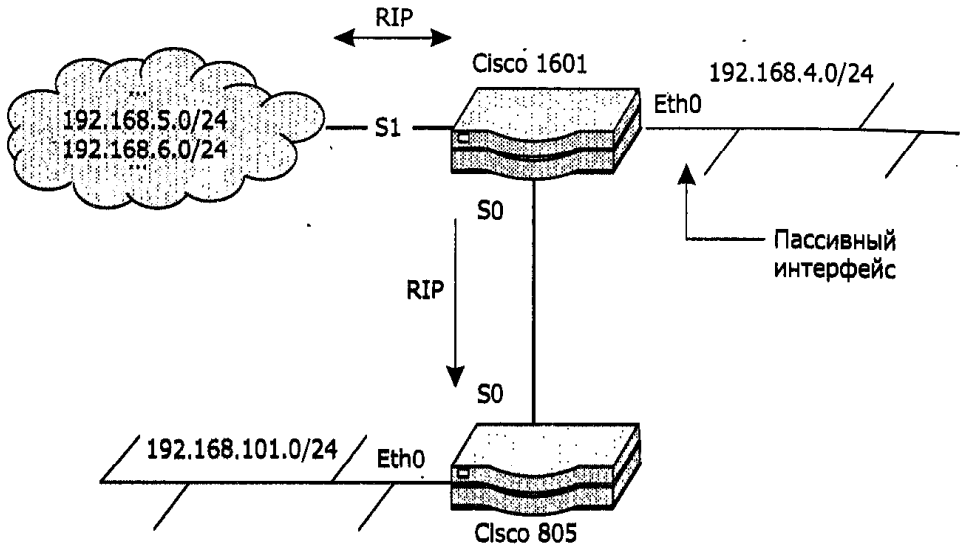


Рис. 7.3. Использование пассивного интерфейса

В приведенном примере сети маршрутизатор Cisco 1601 является как бы т ревалочным пунктом — через его синхронные интерфейсы осуществляется св частей распределенной сети. К нему непосредственно подключается локальн сеть, в которой используются адреса класса С 192.168.4.0/24. В такой конфи рации маршрутизатор должен принимать и отправлять сообщения протокола R через интерфейсы Serial0, Serial1, но поддержка протокола RIP на интерфей Eth0 не является обязательной. Это связано с тем, что в подключенной сети н устройств, опирающихся на работу протокола RIP, — установлены только обы ные клиентские компьютеры. Настроив интерфейс Eth0 как пассивный, мож запретить протоколу посылать свои сообщения через указанный интерфей Для этого предусмотрена команда passive-interface, выполняемая в режиме н стройки протокола RIP:

```
Cisco1601#conf term
Enter configuration commands, one per line. End with CNTL/Z.
Cisco1601(config)#router rip
Cisco1601(config-rout)#passive-interface ethernet 0
```

Несмотря на то что эта команда запрещает посылку сообщений протоко RIP через интерфейс, она не блокирует прослушивания маршрутизатором эт интерфейса на наличие поступающих сообщений протокола. Для выполнен последнего действия нужно использовать фильтрацию маршрутов (route filte Администратор сети может проверить факт работы команды — для этого ну выполнить команду отладки debug ip rip.

Перечисляя все ситуации, когда может потребоваться отключение рассы сообщений протокола RIP через определенный интерфейс, логично выдел

еще одну. Если рассматривать сеть, в которой связь объектов организована через прямые провода по технологии xDSL, то кроме решения, заключающегося в использовании маршрутизаторов с подключенными к ним модемами, у администратора есть альтернатива — модемы со встроенными портами Ethernet. При этом модемы работают как двухпортовые мосты: один порт — Ethernet, второй — канал xDSL. Преимущества такого подхода очевидны: стоимость модемов с портами Ethernet меньше, чем стоимость аналогичных модемов с портом, например V.35, а также не требуется покупать маршрутизаторы и заниматься их настройкой. Но, как всегда и бывает, простота самих устройств и создает определенные недостатки — трудно контролировать передаваемый через них трафик. Невозможно настроить приоритеты для разных типов трафика, недоступна блокировка определенных пакетов и т. п. Однако если вопрос цены был определяющим и на этом основании принято решение использовать именно эти модемы, то администратору стоит обратить внимание еще на один фактор: модемы работают как мосты и, соответственно, не блокируют широковещательный трафик, каковым, например, является трафик протокола RIP. Если к локальной сети подключен маршрутизатор (маршрутизаторы), а также подобные модемы, то будет разумным на маршрутизаторе командой passive-interface запретить трансляцию широковещательного трафика в локальную сеть. Обычно в локальной сети существует множество других источников широковещательного трафика (например, протокол ARP), которые по сравнению с протоколом RIP генерируют его в большем объеме, но если из-за плохого качества выделенной пары проводов не удастся добиться высокой скорости, то и столь малый вклад тоже может быть полезным.

Когда администратор сети настраивает связь на двух маршрутизаторах и указывает на каждом интерфейсе маршрутизатора IP-адрес, в этом случае маршрутизаторы немедленно получают в свое распоряжение информацию о непосредственно подключенных к ним сетях. Так, если на интерфейсе Ethernet0 маршрутизатора настроен, например, адрес 192.168.4.254 с маской 255.255.255.0, то отсюда следует, что маршрутизатор подключен к сети, в которой используется адресная схема класса C 192.168.4.0/24. Эта информация важна, так как если на любой другой порт маршрутизатора придет пакет, адресованный в сеть 192.168.4.0/24, то маршрутизатор просто пошлет запрос протокола ARP на свой порт Ethernet0, а затем напрямую по выявленному MAC-адресу отправит пакет конечному получателю.

Рассмотрим в действии работу протокола маршрутизации RIP. Для этого в качестве примера рассмотрим сеть, в которой 6 объектов связаны между собой через сеть Frame Relay и выделенные каналы связи (рис. 7.4).

Здесь используется внутренняя адресация на базе сетей класса C (192.168.(1–101).0/24) и, таким образом, на каждом порте Ethernet маршрутизатора настроен адрес соответствующей сети, заканчивающийся на .254, с маской 255.255.255.0 — это также указывает на используемый класс адреса в подключенном сегменте сети (например, если адрес маршрутизатора 192.168.1.254, то предполагается, что подключенной сети Ethernet выделены адреса класса C 192.168.4.0/24).

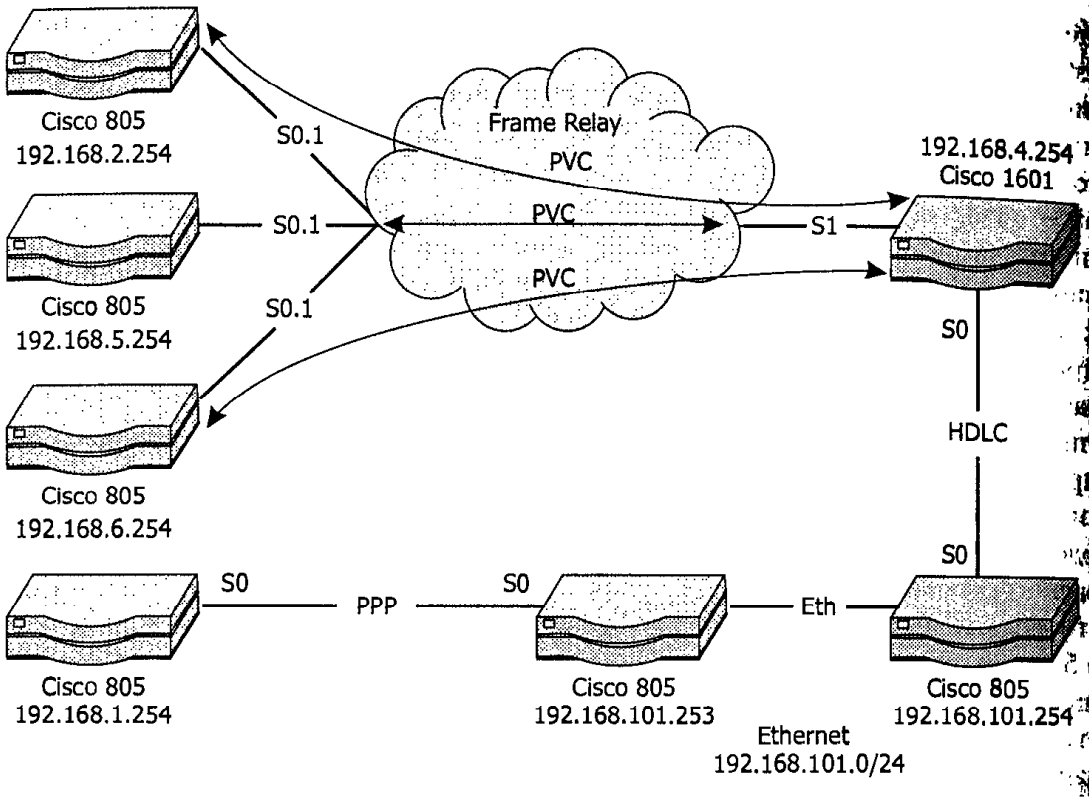


Рис. 7.4. Пример реальной сети

В том случае, если на маршрутизаторах еще не настроена схема маршрутизаци (статическая или динамическая), они обладают только информацией о локальн подключенных сетях. Этот факт можно подтвердить, выполнив команду **show** route на любом маршрутизаторе в сети (например, Cisco 1601).

```
Cisco1601#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate defaul
       U - per-user static route, o - ODR

Gateway of last resort is not set

C       192.168.4.0/24 is directly connected, Ethernet0
```

Символы (коды), которые выведены в начале листинга, помогают определит способ обнаружения маршрута. Код C (connected) показывает, что маршрут изве тен, так как он напрямую подключен к маршрутизатору. Поскольку на вновь н страиваемых маршрутизаторах динамическая маршрутизация по умолчанию задействуется, то в рассматриваемой сети без дополнительных конфигурационны команд маршрутизаторы не будут владеть информацией о возможных маршрут Предположим, что администратор сети, работая с командной строкой маршрут

тора Cisco 1601, выполняет команду ping 192.168.1.254. Маршрутизатор проверит свою таблицу маршрутизации для поиска маршрута в сеть 192.168.1.0/24 и, не найдя таковую, не сможет выполнить команду ping и выдаст ошибку.

Настройка протокола RIP на маршрутизаторах компании Cisco Systems достаточно проста и предполагает использование только двух команд: router rip и network. Начиная с модели Cisco 1601 настройка протокола будет осуществляться следующим образом:

```
Cisco1601#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco1601(config)#router rip
Cisco1601(config-router)#network 192.168.4.0
```

Команда router rip активизирует протокол маршрутизации RIP и изменяет строку приглашения на (config-router)#. Следующая команда — network 192.168.4.0 — дает инструкцию маршрутизатору выполнить обработку динамической маршрутизации для сети 192.168.4.0, или, иными словами, указывает маршрутизатору на необходимость посылки и приема сообщений протокола RIP на всех интерфейсах, являющихся частью этой сети. В рассматриваемом примере это интерфейсы Ethernet0, S1 и S0. Интерфейсы Serial*X* также будут рассматриваться маршрутизатором в качестве части сети 192.168.4.0. Это обусловлено тем, что на них используется механизм сохранения адресов, настроенный с помощью команды ip unnumbered Ethernet0.

ВНИМАНИЕ

При вводе команды network указывается адрес сети, а не подсети.

Аналогичным образом настраиваются и другие маршрутизаторы. Понятно, что в качестве параметра команды network записываются другие адреса сетей (например, network 192.168.1.0), то есть адреса напрямую подключенных сетей.

СОВЕТ

В том случае, если маршрутизатор имеет два порта Ethernet, как, например, модель Cisco 1605R, и к каждому из них подключены сети, например класса B, 172.16.1.0/24 и 172.16.2.0/24, то при настройке протокола RIP на этом маршрутизаторе следует ввести команду network 172.16.0.0.

После того как на всех маршрутизаторах будет включена поддержка протокола RIP, они начнут обмениваться информацией, рассылаемой каждым из них всем своим соседям (точнее говоря, на все свои интерфейсы). Эта информация включает в себя записи о напрямую подключенных сетях и о сетях, обнаруженных с помощью протокола RIP в сообщениях от других маршрутизаторов. Маршрутизаторам в сети потребуется некоторое время для получения сведений обо всех доступных маршрутах. Впоследствии администратор может проверить результат работы протокола RIP, выполняя команду show ip route.

```
Cisco1601#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate defau
        U - per-user static route, o - ODR

Gateway of last resort is 192.168.101.254 to network 0.0.0.0

C       192.168.4.0/24 is directly connected, Ethernet0
R       192.168.5.0/24 [120/1] via 192.168.5.254, 00:00:06, Serial1.3
R       192.168.6.0/24 [120/1] via 192.168.6.254, 00:00:23, Serial1.2
R       192.168.1.0/24 [120/3] via 192.168.101.254, 00:00:25, Serial0
R       192.168.2.0/24 [120/1] via 192.168.2.254, 00:00:15, Serial1.1
R       192.168.101.0/24 [120/1] via 192.168.101.254, 00:00:25, Serial0
R*      0.0.0.0/0 [120/1] via 192.168.101.254, 00:00:25, Serial0
```

Каждая запись в таблице маршрутизации помечается кодом, указывающи на источник информации. Все записи, полученные с помощью протокола RI помечаются кодом R. Рассмотрим для примера одну из записей:

R 192.168.1.0/24 [120/3] via 192.168.101.254, 00:00:25, Serial0.

Эта запись говорит о том, как можно достичь сети 192.168.1.0/24, что и я ляется маршрутом до получателя. Маршрутизатор, принимая пакет с адресо получателя в сети 192.168.1.0/24, проверяет таблицу маршрутизации, находи рассматриваемую запись и анализирует ее. Анализ показывает, что искомая се находится на расстоянии трех переходов от маршрутизатора, что обозначает записью [120/3], где 120 подразумевает административно заданное расстоян (administrative distance), а цифра 3 определяет метрику маршрута (3 перехода Административное расстояние определяет уровень приоритета и значимо, ес маршруты получены с использованием нескольких протоколов маршрутизаци Метрика в протоколе RIP отражает количество переходов, или, иными словам количество маршрутизаторов, которое пакет минует по пути в сеть 192.168.1 /24. Запись 00:00:25 указывает на «возраст» маршрута — 25 секунд назад мар рутизатор получил информацию об этом маршруте. Учитывая, что протоко RIP рассылает сообщения каждые 30 секунд, значения в данной записи нах дятся в интервале от 0 до 30 секунд при нормальной работе сети. В том случа если, например, сеть 192.168.1.0/24 перестанет быть доступной по каким ли причинам, маршрутизатор увеличит этот интервал до 240 секунд, а затем удал маршрут из таблицы маршрутизации. Следующая запись в маршруте — Serial0 уведомляет маршрутизатор, что пакеты, адресованные в сеть 192.168.1.0/2 следует передавать через интерфейс Serial0.

Остановимся более подробно на рассмотрении понятия расстояния, опред ленного в административном порядке, которое используется, если в распред ленной сети работают несколько протоколов маршрутизации, то есть когда мар рутизатор получает сообщения от нескольких протоколов. Естественно, вст вопрос: если маршрутизатор получает сообщения протоколов RIP и OSPF, в конечном итоге каким именно маршрутам он должен отдать предпочтени Ответ на этот вопрос осложняется еще тем, что данные протоколы использу различные метрики и маршрутизатор просто не может выбрать нужный марш

основываясь на простом их сравнении. Именно в таком случае он проверяет административные расстояния.

Предположим, что маршрутизатор на два своих интерфейса (Serial0, Serial1) получает сообщения от двух протоколов маршрутизации RIP и OSPF, рекламирующих маршрут в одну сеть 192.168.1.0/24. При этом для выбора маршрута маршрутизатор сличит соответствующие административные расстояния и выберет тот маршрут, для которого таковое будет меньше. Далее маршрут будет занесен в таблицу для использования при достижении получателя. В нашем примере маршрутизатор отдаст предпочтение протоколу OSPF перед протоколом RIP, так как у первого административное расстояние меньше. Значения административных расстояний для каждого протокола маршрутизации приведены в табл. 7.1. При этом по умолчанию все маршруты от одного протокола будут иметь равные административные расстояния (120 для протокола RIP).

Таблица 7.1. Административные расстояния по умолчанию

Источник маршрута	Административное расстояние
Подключенный интерфейс	0
Статический маршрут	1
Enhanced IGRP (EIGRP)	5
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
Неизвестный	255

Команду show ip route можно выполнить на любом маршрутизаторе в сети, и результаты будут похожи, за исключением, естественно, метрик и значений таймеров. Например, выполняя данную команду на маршрутизаторе с адресом 192.168.5.254, получим результат, из которого видно, что для достижения сети 192.168.1.0/24 пакету потребуется пересечь 4 маршрутизатора в пути своего следования (запись 192.168.1.0/24 [120/4] via 192.168.4.254, 00:00:16, Serial0.1).

```
Cisco805#show ip route
...
Gateway of last resort is 192.168.4.254 to network 0.0.0.0

R     192.168.4.0/24 [120/1] via 192.168.4.254, 00:00:06, Serial0.1
C     192.168.5.0/24 is directly connected, Ethernet0
R     192.168.6.0/24 [120/2] via 192.168.4.254, 00:00:06, Serial0.1
R     192.168.1.0/24 [120/4] via 192.168.4.254, 00:00:16, Serial0.1
R     192.168.2.0/24 [120/2] via 192.168.4.254, 00:00:23, Serial0.1
R     192.168.101.0/24 [120/2] via 192.168.4.254; 00:00:04, Serial0.1
R*    0.0.0.0/0 [120/2] via 192.168.4.254, 00:00:08, Serial0.1
```

СОВЕТ

Обратите внимание на то, через какой интерфейс следует передавать пакеты в целевые сет
Установлен интерфейс Serial0.1. Это объясняется тем, что на маршрутизаторе настроен прото
кол Frame Relay для связи с другими объектами и Serial0.1 является логическим подинтерфейсом.
Однако с точки зрения протокола RIP разницы по сравнению, например, с физическим интерфей
сом Serial0 нет никакой. Кроме того, следует заострить внимание на том, как пакеты от маршрути
затора 192.168.5.254 будут достигать, например, сети 192.168.2.0 /24: для этого им придется
сделать «крюк» — сначала дойти до маршрутизатора Cisco 1601. Очевидно, что это определяется
выбранной схемой постоянных виртуальных соединений в сети Frame Relay. Если условия дик
туют необходимость снижения задержек при передаче информации между двумя точками в рас
пределенной сети, то и схему постоянных виртуальных соединений нужно выбирать соответст
вующим образом (глава 8).

В том случае, если наблюдаются проблемы при работе протокола RIP, адми
нистратор сети может включить на маршрутизаторе режим отладки. Для этого
следует воспользоваться командой debug ip rip, предварительно настроив выдачу
отладочных сообщений по сеансу Telnet с помощью команды terminal monitor (если
подключение к маршрутизатору осуществляется с помощью терминального пор
та, то последнюю команду выполнять не требуется).

Отдельно следует остановиться на рассмотрении маршрута по умолчанию
(default route), который позволяет масштабировать сеть и сокращать ресурсы, за
действованные на промежуточных маршрутизаторах. Основная задача маршрута
по умолчанию заключается в выработке указания о способе достижения неизвест
ного получателя — получателя, информация о котором отсутствует в таблице
маршрутизации, так как она не была введена вручную или не получена из про
токола маршрутизации. Если на маршрутизаторе настроен маршрут по умолча
нию, то он при обнаружении пакетов к неизвестному ему получателю передаст
их тому маршрутизатору, который был указан в маршруте по умолчанию. Если
же маршрут по умолчанию не был настроен, то пакет будет отброшен.

Не следует расценивать маршрут по умолчанию как способ избавления от оши
бок маршрутизации. Его присутствие позволяет не хранить на маршрутизаторе
всю адресную информацию о сети, снижая тем самым объем задач по управ
лению и обслуживанию. При этом размер таблицы маршрутизации снижается,
и на ее анализ требуется меньше циклов процессора маршрутизатора. Наиболее
часто используется маршрут по умолчанию при подключении распределенной
сети к Интернету.

Так, представим, что организация, имеющая сеть, состоящую из нескольких
удаленных объектов, хочет подключиться к Интернету. Одним из вариантов под
ключения может быть покупка и установка граничного маршрутизатора, один
интерфейс которого подключается к внутренней сети, а другой — к интернет-
провайдеру. После этого администратор сети настраивает на маршрутизаторе
маршрут по умолчанию, отмечая необходимость отправки трафика по неизвестным
ему адресам в Интернет. Здесь следует отметить, что трафик будет передаваться
маршрутизатору на стороне провайдера. В ситуации, когда граничный маршру
тизатор соединяет с Интернетом только одну локальную сеть, этого будет доста
точно, но при наличии множества удаленных объектов потребуется приложи

усилия, чтобы маршрут по умолчанию стал также известен и другим маршрутизаторам (рис. 7.5).

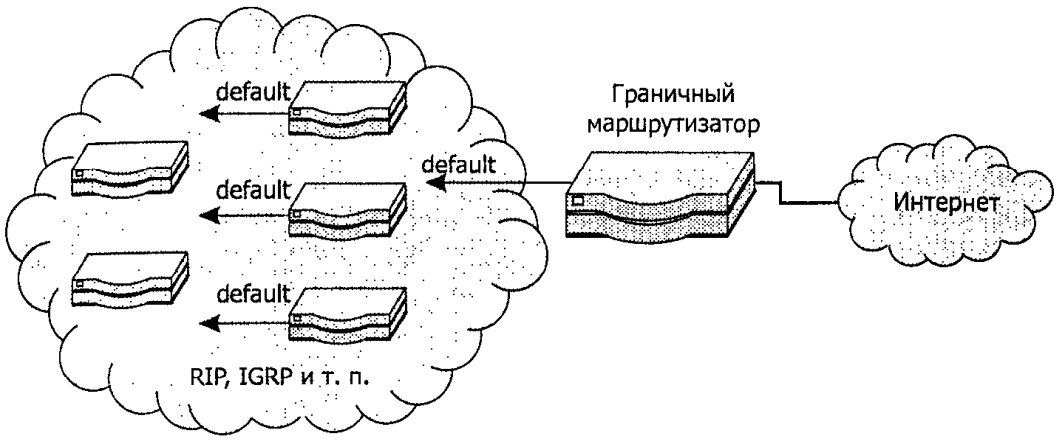


Рис. 7.5. Использование маршрута по умолчанию

В рассматриваемом примере корпоративная сеть подключается к Интернету и для этого используется граничный маршрутизатор с настроенным маршрутом по умолчанию. В результате ему не требуется хранить информацию обо всех возможных получателях в глобальной сети. С этой точки зрения, граничный маршрутизатор выполняет свою задачу, но как быть другим маршрутизаторам? Они также должны владеть информацией о маршруте по умолчанию, иначе пакеты, адресованные в Интернет, будут отбрасываться маршрутизаторами, так как они не знают этого маршрута. Для информирования остальных граничный маршрутизатор включает описание маршрута по умолчанию в сообщения протокола маршрутизации так же, как информацию о других маршрутах. Каждый маршрутизатор, участвующий в работе протокола маршрутизации, получает маршрут по умолчанию, помещает его в свою таблицу и может отправлять трафик в Интернет.

Если в распределенной сети используется протокол RIP, то чтобы граничный маршрутизатор сумел проводить рассылку маршрута по умолчанию, потребуется настройка самого маршрута по умолчанию на этом маршрутизаторе, а остальное протокол RIP сделает сам. Рассмотрим пример сети с защитным экраном (рис. 7.6).

В рассматриваемом примере сети подключение к Интернету выполняется с помощью защитного экрана, который помимо функций фильтрации трафика выполняет также задачи маршрутизации. Но защитный экран предоставляет лишь минимальные возможности по маршрутизации, ограниченные поддержкой статических записей в таблице. Кроме поддержки правил фильтрации трафика на защитном экране настроена технология NAT, позволяющая выполнять динамическую трансляцию внутренних адресов (192.168.X.X) во внешние адреса, зарегистрированные в Интернете. В данном случае защитный экран работает на сервере под управлением операционной системы Windows NT и имеет три сетевые карты Ethernet, одна из которых связана с Интернетом (способ подключения пока не

важен), а две других — к внутренним сетям. В принципе, можно было бы обо тись только двумя сетевыми картами, но рассмотренный способ позволяет п лучить больший контроль над доступом в Интернет с удаленных объектов (се 192.168.3.0/24 рассматривается как сеть центрального офиса). С точки зрен удаленных маршрутизаторов для отправки информации в Интернет им про сто потребуется передать пакеты защитному экрану, который затем перена правит их в Интернет. В данном примере защитный экран стоит рассматривать как маршрутизатор, поддерживающий только статическую маршрутизацию, в т время как остальные маршрутизаторы используют протокол RIP. Из этого сл дует, что на одном из маршрутизаторов, подключенных к сети 192.168.101.0/2 нужно настроить маршрут по умолчанию, указывающий на интерфейс защи ного экрана с адресом 192.168.101.1. Для этого на нем требуется ввести коман ip route 0.0.0.0 0.0.0.0 192.168.101.1.

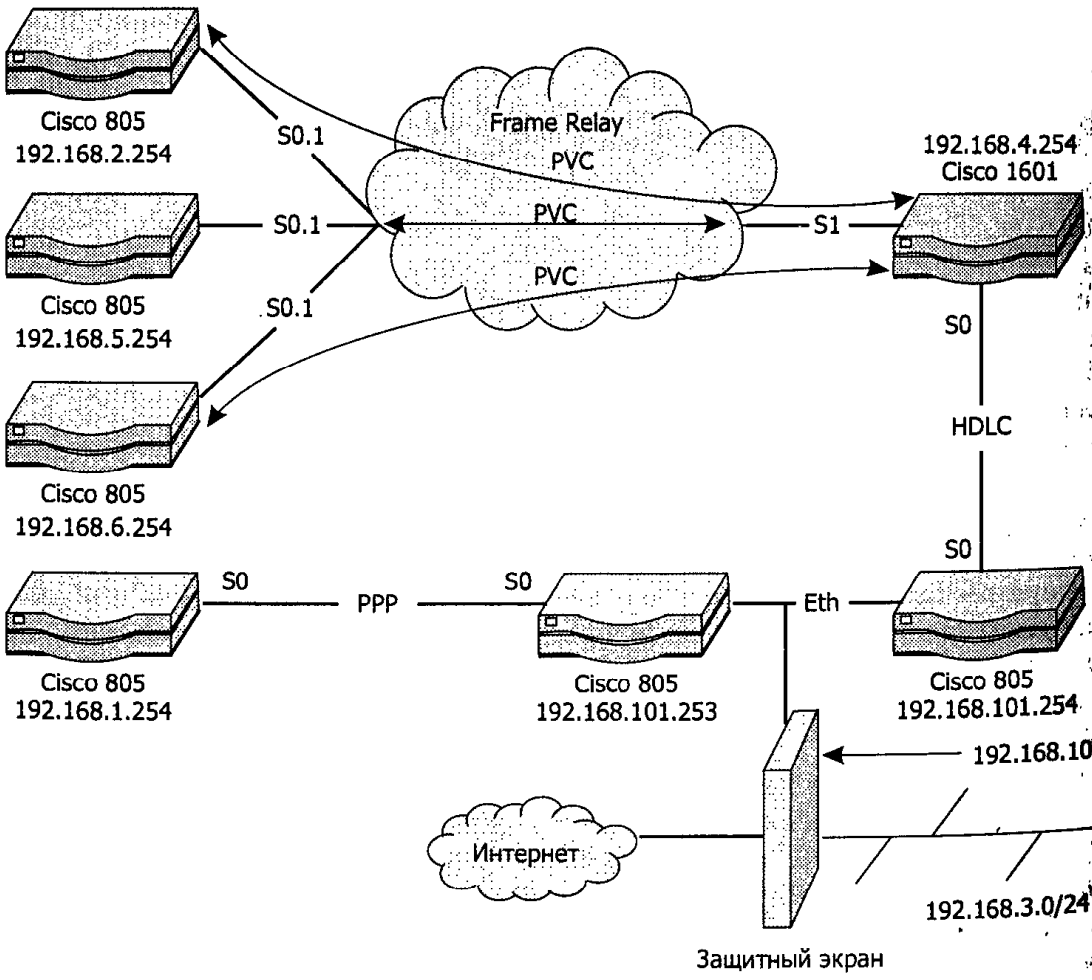


Рис. 7.6. Пример сети с защитным экраном

Данная команда настраивает статический маршрут, но использует специа ные адреса сетей с маской подсети, состоящей из одних нулей. Для маршру

тора он и будет являться маршрутом по умолчанию, а защитный экран будет последним маршрутизатором в пути (не совсем точный перевод английского термина gateway of last resort). Проверить корректность ввода маршрута по умолчанию можно с помощью команды **show ip route**, выполняемой, например, на маршрутизаторе с адресом 192.168.101.254. Далее следует обратить внимание на запись **Gateway of last resort** ...

```
Cisco805#show ip route
...
Gateway of last resort is 192.168.101.1 to network 0.0.0.0

R     192.168.4.0/24 [120/1] via 192.168.4.254, 00:00:21, Serial0
R     192.168.5.0/24 [120/2] via 192.168.4.254, 00:00:19, Serial0
R     192.168.6.0/24 [120/2] via 192.168.4.254, 00:00:03, Serial0
R     192.168.1.0/24 [120/2] via 192.168.101.253, 00:00:16, Ethernet0
R     192.168.2.0/24 [120/2] via 192.168.4.254, 00:00:09, Serial0
S     192.168.3.0/24 [1/0] via 192.168.101.1
C     192.168.101.0/24 is directly connected, Ethernet0
S*    0.0.0.0/0 [1/0] via 192.168.101.1
```

Первая запись результата выполнения команды подтверждает корректность установки маршрута по умолчанию (**Gateway of last resort is 192.168.101.1 to network 0.0.0.0**). Обратите внимание на последнюю запись в таблице маршрутизации. Она имеет код S*, что указывает на то, что запись в таблице статическая и содержит информацию о маршруте по умолчанию. Сразу после настройки маршрута по умолчанию протокол RIP начинает информировать о нем соседние маршрутизаторы. Здесь важно отметить один момент. Маршрут по умолчанию является статическим, а для рассылки в своих сообщениях статических маршрутов протоколу RIP нужно дать прямое указание на это с помощью команды **redistribute static**. Но маршрут в сеть 0.0.0.0 с маской 0.0.0.0 является исключением — протокол будет рассылать о нем информацию, даже если указанная команда пропущена.

Вводя команду **show ip route** на других маршрутизаторах распределенной сети, можно убедиться, что они обладают информацией о маршруте по умолчанию. Ниже приведен результат работы этой команды на маршрутизаторе с адресом 192.168.5.254.

```
Cisco805#show ip route
...
Gateway of last resort is 192.168.4.254 to network 0.0.0.0

R     192.168.4.0/24 [120/1] via 192.168.4.254, 00:00:11, Serial0.1
C     192.168.5.0/24 is directly connected, Ethernet0
R     192.168.6.0/24 [120/2] via 192.168.4.254, 00:00:33, Serial0.1
R     192.168.1.0/24 [120/4] via 192.168.4.254, 00:00:12, Serial0.1
R     192.168.2.0/24 [120/2] via 192.168.4.254, 00:00:23, Serial0.1
R     192.168.3.0/24 [120/2] via 192.168.4.254, 00:00:12, Serial0.1
R     192.168.101.0/24 [120/2] via 192.168.4.254, 00:00:08, Serial0.1
R*    0.0.0.0/0 [120/2] via 192.168.4.254, 00:00:16, Serial0.1
```

АТАКИ RIP ENTRY ADDED И RIP ENTRY TIMEOUT

Атака RIP Entry Added основывается на перехвате сообщений протокола RIP и добавлении в н информации (например, обновление таблицы маршрутизации в результате добавления новых се тей). Естественно, это приведет к тому, что маршрутизаторы (как программные, так и аппаратны в сети будут владеть информацией о новых сетях. Подобный факт появления новых сетей объяс няется по-разному. Во первых, администратор мог подключить к сети новый объект и включив новую сеть в общую схему сети. Другим объяснением может быть то, что некая сеть стала вновь достижимой после некоторых проблем со связью. Есть еще одно объяснение, а именно то, что кто-то пытается получить доступ к сети, внося в таблицы маршрутизации ложную информацию. К сожалению, протокол RIP (первая его версия) не использует аутентификацию, таким образом маршрутизаторы нетрудно обмануть, перехватывая и модифицируя сообщения этого протокола. В качестве меры противодействия можем рекомендовать периодическую проверку таблиц на пред мет новых записей, точнее их источников. В том случае, если источником был какой-либо компью тер в сети, стоит проверить причину, по который он добавил маршрут. Здесь просто не исключает ся и установка дополнительной сетевой карты с настройкой протокола RIP.

В основе атаки RIP Entry Timeout лежит проверка сообщений протокола RIP и поиск в них инфор мации о том, что некая сеть стала недостижимой. Причин этому может быть несколько, первой из них, естественно, будет та, что сеть стала недостижимой по причине сбоя в канале связи или маршрутизатора. Впрочем, вероятно, что просто кто-то пытается добавить ложную информацию в таблицы маршрутизации. Действия для проверки могут быть теми же, что и в случае атаки RIP Entry Added. Продолжением этой атаки иногда является атака RIP Metric Change Decode, при кото рой перехватываются и модифицируются сообщения протокола RIP, а именно расстояния до сетей (метрики).

В заключение для сравнения очень коротко остановимся на настройке прото кола IGRP на маршрутизаторе. Он, так же, как и протокол RIP, относится к се мейству протоколов длины вектора. Практическая настройка протокола IGRP так же проста, как и настройка протокола RIP. Это все те же две базовые коман ды router igrp и network. Разница в том, что при вводе команды router igrp нужно указать номер автономной системы в качестве параметра — это может быть любое число в интервале от 1 до 65 535. Маршрутизаторы, на которых ис пользуется один номер автономной системы, будут обмениваться информацией друг с другом. Иначе они не смогут взаимодействовать, считая, что принадле жат к разным автономным системам. Ниже приведен пример настройки прото кола IGRP на маршрутизаторе Cisco 1601 (номер автономной системы — 100):

```
Cisco1601#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco1601(config)#router igrp 100
Cisco1601(config-router)#network 192.168.4.0
...
```

Пример подключения объектов к сети Интернет

Остановимся на рассмотрении практического примера подключения к сети Ин тернет с помощью динамической маршрутизации. Для этого представим неку организацию, перед администратором которой встала задача обеспечить быстр

доступ в сеть Интернет, желательно минимизируя затраты на саму связь. Собственно, если обратиться к какому-либо крупному провайдеру, который возьмет на себя все заботы по подключению и организации связи (например, протянет оптоволоконный кабель), то такой подход позволит значительно смягчить «головную боль» администратора. Но, предположим, было принято решение самостоятельно сделать подключение и для этого заказать формирование медной пары проводов от места расположения компании до провайдера услуги Интернет (дополнительную информацию о подобном методе подключения можно найти в главе 10). А теперь отвлечемся от гипотетических мыслей и посмотрим, как это происходило некоторое время назад в одной из реально существующих организаций.

Можно сказать, что компании повезло, так как провайдер территориально оказался рядом и до него можно было «дотянуться» прямым проводом. Для самой организации связи была взята на тестирование, а затем закуплена пара модемов ASMi-50 производства компании RAD Data Communication (www.rad.com). На момент покупки это была новая модель, ну а сейчас, естественно, производитель предлагает и другие варианты. Рисунок 7.7 взят с сайта производителя, и на нем показаны три типа исполнения модели, слева и внизу — ведущий, вверху — ведомый, где к модему можно подключить внешний жидкокристаллический дисплей (опционально). В правой части рисунка представлено исполнение модема, предназначенное для установки в специальное шасси (как правило, на стороне провайдера).

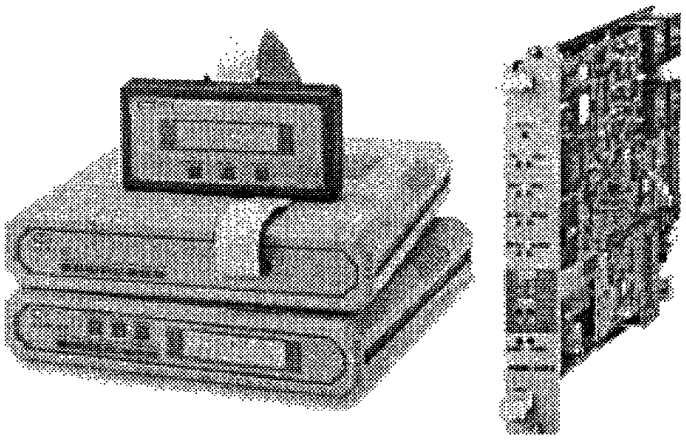


Рис. 7.7. Модем ASMi-50 компании RAD Data Communication

Как видно из рисунка, один из модемов поставляется со встроенным жидкокристаллическим дисплеем, с помощью которого можно управлять параметрами связи (например скоростью передачи). Этот модем является «ведущим» (master), и он был установлен в помещении компании. Второй модем из этой пары работает в качестве «ведомого» (slave), и он был установлен на стороне провайдера. Модемы приобретались с интерфейсом V.35 (тип разъема «мама»), с помощью которого они подключаются к маршрутизатору. Сами модемы поставлялись без кабеля V.35, и при заказе маршрутизатора нужно об этом помнить. Кстати, то, что интерфейс V.35 смонтирован с модемом, очень удобно. В моделях других

производителей такое конструктивное исполнение встречается довольно редк
Обычно модем имеет разъем типа DB9 или DB 25 и для их подключения зак
зывается специальный кабель, на одном конце которого находится разъем DB9
DB25 для подключения к модему, а на другом конце разъем V.35 (тип разъема
«мама») для коммутации с разъемом V.35 («папа») кабеля, приходящего от
маршрутизатора. В результате общая длина составного кабеля способна дос
тигнуть 3 метров, и по этой причине его приходиться скручивать, формируя
толстое кольцо. Это кольцо можно повесить на направляющие коммутационно
го шкафа или положить на полку.

Скорость передачи удалось поднять до 512 Кбит/с при длине провода при
близительно в 2 км и сопротивлении 1000 Ом. При этом время простоя составляе
ориентировочно 2–3 часа в 3 месяца (квартал), а иногда еще меньше. Естественно
все будет зависеть от сформированного прямого провода и его характеристик
Хотя были ситуации, когда в результате крупных неполадок на телефонных
трассах доступ в сеть Интернет отсутствовал пару дней. Хочется отметить, что
в Санкт-Петербурге подключение к сети Интернет по выделенному каналу обыч
но как раз и организуется по прямому проводу. Как альтернатива может быть
подключение по технологии Frame Relay, но если абстрагироваться от самой
технологии, то физически подключение останется прежним — провайдер сфор
мирует медную пару проводов (или, как часто говорят, «пару» или «прямой
провод») от ближайшей АТС, где у него размещено оборудование Frame Relay
а затем установит свое оборудование. В том случае, если выход в сеть Интернет
очень важен для компании или в планах предусмотрена такая потребность, име
ет смысл заказать прокладку оптоволоконного кабеля.

Несколько слов относительно доступа в сеть Интернет по технологии ADSL,
или асимметричной цифровой абонентской линии (Asymmetric Digital Subscriber
Line). В настоящее время эта технология становится все более популярной, так
как позволяет передавать данные по обычным телефонным линиям, занимая не-
используемый обычной телефонией частотный диапазон. Представьте, как это
удобно — на объекте есть телефон и нужно сделать доступ в сеть Интернет. В том
случае, если требуется высокоскоростной доступ, то следует либо заказывать
прямой провод от ближайшего узла провайдера, либо принести в жертву теле-
фон (если возможность проложить дополнительный прямой провод отсутству-
ет, то провайдер может попросить отказаться от телефона и занять освободив-
шийся кабель). В варианте ADSL решение находится гораздо проще. Как видно
из названия технологии, передача данных происходит на различных скоростях,
причем, как правило, скорость в направлении к абоненту выше чем в противо-
положном. Это очень удобно для работы с сетью Интернет, поскольку обычно
в такой ситуации объем входящего трафика гораздо больше исходящего. Одна-
ко наблюдения за работающим web-сайтом показали, что при его определенной
популярности соотношение меняется на противоположное, причем разница бы-
вает двукратной, и это нужно учесть, если есть желание разместить сайт WWW
а подключение сделано по ADSL — внешние пользователи могут ощущать суще
ственные задержки.

Вернемся к действиям нашей организации. Первоначально в качестве марш
рутизатора для доступа в Интернет использовалась модель Cisco 1005, кото
рая имеет один порт Ethernet и один порт Serial (поддерживается скорость

редачи до 2 Мбит/с). Постепенно компания стала расширяться и открыла еще офис в городе, с которым нужно было также организовать связь. Причем в обоих офисах сотрудники должны были иметь возможность работать в Интернете. Для связи были выбраны маршрутизаторы 1720 компании Cisco Systems, которые изначально поставляются с одним портом 10/100 Ethernet и двумя слотами расширения. Для каждого маршрутизатора был заказан модуль WIC-2T, который имеет два синхронных порта (Serial) и каждый такой порт поддерживает передачу со скоростью до 2 Мбит/с (рис. 7.8).

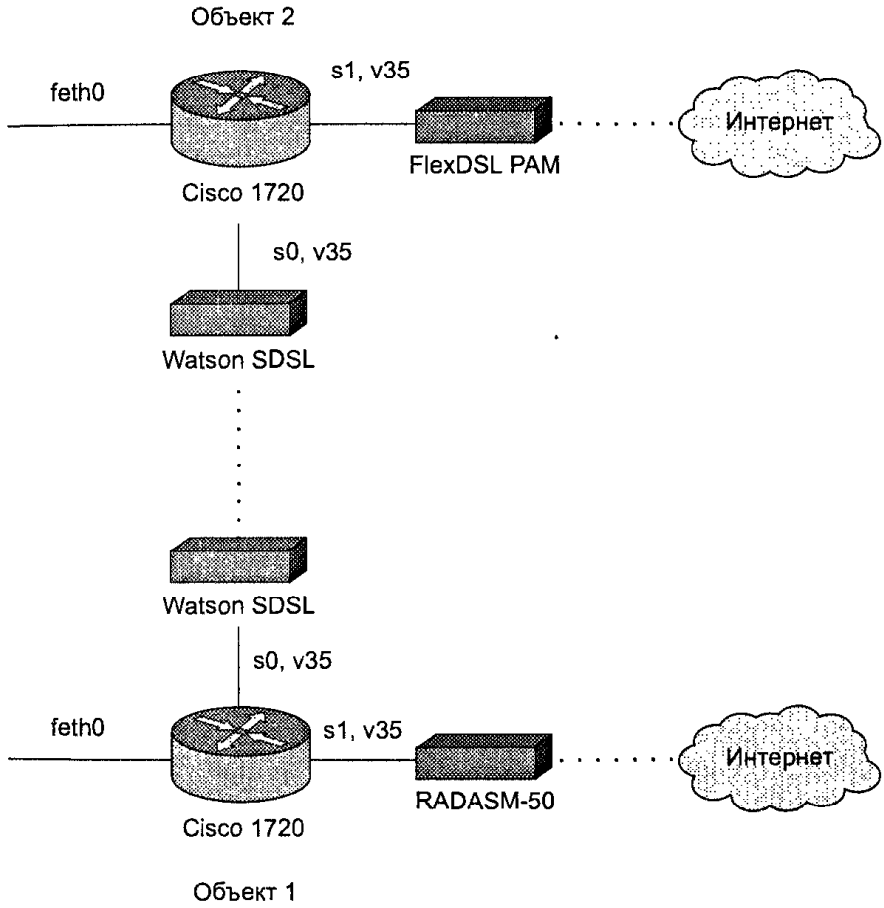


Рис. 7.8. Связь двух сетевых объектов

Учитывая тот факт, что для осуществления доступа в сеть Интернет используются прямые провода, которые не обеспечивают достаточной надежности, было принято решение организовать два канала в сеть Интернет из двух объектов. Каналы не зависят друг от друга, хотя оба они заканчиваются на площадках одного и того же провайдера. Оба канала работают одновременно и в то же время обеспечивают обоюдное резервирование на случай возникновения проблем со связью. Например, если в какой-либо момент времени с объекта 1 пропадает связь с Интернетом, то сетевой трафик будет автоматически переадресован в Интернет через объект 2 — при этом на пути трафика просто возникнет дополнительный переход (между объектами удалось сформировать прямой провод и были установлены

модемы Watson компании SCHMID TELECOM AG). Детали организации подо ной схемы резервирования будут рассмотрены далее, а пока отметим, что так подключение требует активного участия провайдера и может быть выполне с помощью протокола динамической маршрутизации, например OSPF.

Организация канала в сеть Интернет для объекта 2 выполнялось точно так же как и для объекта 1 — была написана заявка в отдел ПТС (Петербургская теле фонная сеть), занимающийся выделением прямых проводов, и после оплаты бы сформирована одна пара медного провода с сопротивлением порядка 1000 Ом Для организации канала были закуплены модемы «Натекс» FlexDSL с модуля цией PAM и портами V.35. На сформированном прямом проводе максимальна поддерживаемая скорость составляла порядка 1,8 Мбит/с, но при такой скоро сти наблюдалась периодическая рассинхронизация между модемами. На скоро сти 1,6 Мбит/с модемы работали довольно устойчиво. К слову, время восста новления синхронизации между модемами составляет порядка 15–20 секунд.

ПРИМЕЧАНИЕ

Модемы FlexDSL PAM основаны на самой современной линейной технологии из ряда DSL — TC-PAM TC-PAM была разработана для одновременного решения двух задач — достижения максимально дальности работы и обеспечения электромагнитной совместимости с другими DSL-устройствам работающими по соседним парам. Технология TC-PAM лежит в основе первого всемирного стандарта ITU на высокоскоростную симметричную передачу по одной паре — G.shdsl. При заданной длин абонентской линии использование TC-PAM обеспечивает увеличение скорости на 15–45% по срав нению с CAP или 2B1Q. При фиксированной скорости использование TC-PAM обеспечивает увели чение дальности на 10–20% по сравнению с другими технологиями (данные компании «Натекс»).

Сама настройка модемов «Натекс» FlexDSL выполняется с помощью кабеля подключаемого к COM-порту компьютера. При настройке можно указать ско рость и другие параметры, в частности уточнить, какой модем будет работат как «ведущий» (master), а какой — как «ведомый» (slave). В рассматриваемо модели предоставлены очень удобные возможности управления «ведомым» мо демом непосредственно с ведущего (после того как модемы установили дру с другом связь). Кроме того, скорость передачи данных меняется только на ведущем модеме, и если параметры прямого провода позволяют, ведомый полу чает эту информацию автоматически. В описываемой схеме построения сети можно рекомендовать устанавливать ведущий модем в корпоративной сети, а ве домый отдать поставщику услуг Интернета. Такое решение позволит оператив но менять скорость, если внезапно произошло ухудшение параметров прямого провода. Также к достоинствам этой модели модемов можно отнести отсутствие внешнего блока питания (хотя обычно модемы поставляются с внешним блоком питания).

Как уже было сказано, модем подключается к маршрутизатору с помощь интерфейса V.35. Для этого при приобретении маршрутизатора нужно не забы включить в заказ кабель. В обсуждаемой сети на объекте 2 установлен маршру тизатор Cisco 1720, для которого дополнительно был закуплен модуль WIC-2 предоставляющий два синхронных порта. Эти два порта при настройке маршр тизатора обозначены как Serial0 и Serial1. Для подключения к сети Интерне или, иными словами, к модему FlexDSL, использовался порт Serial1 (S1 — краткости). Непосредственно настройка порта на маршрутизаторе очень прост

В том случае, если в вашей сети будет только один канал в сеть Интернет, то следует обсудить с провайдером схему адресации и тип протокола канального уровня (например PPP или HDLC), который будет работать между вашим маршрутизатором и маршрутизатором провайдера. В реальности ситуация усложнялась тем, что существовали два канала в сеть Интернет, хотя и к одному провайдеру (на его разные площадки, связанные друг с другом). Провайдер выделил IP-адрес для интерфейса S1 (212.113.96.142, маска 255.255.255.252 или /30), а в качестве протокола канального уровня согласовали использование протокола HDLC. Этот протокол работает по умолчанию на маршрутизаторах Cisco Systems, и поэтому не отображается при просмотре конфигурации. Кроме того, на интерфейсе указан вес (метрика) для протокола маршрутизации OSPF — это требуется для поддержания работоспособности двух каналов в сеть Интернет:

```
!
interface Serial1
ip address 212.113.96.142 255.255.255.252
ip ospf cost 20
!
```

Если модемы были настроены правильно и маршрутизаторы «увидели» друг друга, то при выполнении команды show interface serial1 (в нашем случае) будет выведена следующая информация.:

```
cisco-1720-object2#sh int s1
Serial1 is up, line protocol is up
Hardware is QUICC Serial
Internet address is 212.113.96.142/30
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 186/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
...
798740 packets input, 245195353 bytes, 0 no buffer
Received 9185 broadcasts, 0 runts, 0 giants, 0 throttles
128807 input errors, 37397 CRC, 49704 frame, 0 overrun, 0 ignored, 41706 abort
292075 packets output, 38991848 bytes, 0 underruns
0 output errors, 0 collisions, 20 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

Самое важное отображается в первой строке результата — Serial1 is up означает, что порт маршрутизатора «видит» подключенный к нему модем и получает от него необходимую сигнализацию по интерфейсу V.35 (DCD=up DSR=up DTR=up RTS=up CTS=up), а строка line protocol is up указывает на то, что протокол канального уровня HDLC (Encapsulation HDLC) успешно обменивается информацией между маршрутизатором организации и маршрутизатором на стороне провайдера.

Кстати, довольно интересную информацию можно почерпнуть из счетчиков ошибок на интерфейсе. Они в определенной мере отражают качество физиче-

ского канала между модемами. Чем меньше скорость, установленная на мод мах, тем меньше должно быть количество ошибок. Тут следует найти неки компромисс между скоростью и качеством.

Абсолютно аналогично было осуществлено подключение к сети Интернет объ екта 1, за исключением того, что использовались другие модемы (RAD ASMi-50 и скорость канала была меньше (512 Кбит/с). Ну и естественно, на интерфейс был назначен другой адрес IP — а именно 193.125.189.158/30.

Наличие двух независимых каналов в сеть Интернет продиктовано тем, чт несмотря на относительную дешевизну применения прямых медных проводо для организации связи, вероятность возникновения различных проблем довол по высока. Нужно учитывать, что качество соединения может резко ухудшить при изменении климатических условий (в Санкт-Петербурге с его непредска зуемой погодой подобное происходит часто), могут проводиться ремонтные ра боты или замены кабеля и т. п. Поэтому когда оценивались разные решения ор ганизации доступа, выбор был сделан в пользу организации двух независимы каналов. В общем, по опыту эксплуатации, можно утверждать, что это решени работоспособно и позволяет при относительно небольших затратах получит высокие скорости и приемлемую надежность.

Разумеется, идея поддержания двух независимых каналов потребовала очен тесной интеграции с провайдером услуг сети Интернет. Его специалисты пред ложили использовать протокол динамической маршрутизации OSPF, который помимо всего прочего позволяет очень быстро реагировать на изменения в сете вой топологии. Перед тем как рассматривать технические детали организации связи, остановимся на адресной схеме.

Очевидно, что для организации нормального доступа в сеть Интернет нужно использовать зарегистрированные адреса протокола IP, которые, как правило, выдаются провайдером. Количество выделяемых адресов зависит от множества факторов. Определяющим, как правило, является конкретное техническое реше ние для организации доступа в сеть Интернет. Например, если для доступа ис пользуется одновременно технологии NAT и PAT, то достаточно будет иметь один зарегистрированный (внешний) адрес IP. Если же принято решение о вне дрении только технологии NAT (Network Address Translation) без осуществле ния трансляции портов протоколов TCP/UDP (технология PAT — Port Address Translation), то количество адресов будет зависеть от количества одновременно работающих с Интернетом внутренних пользователей сети. В нашем примере при заключении договора провайдер выделил целиком сеть класса C — 193.125.203.0 /24. Вначале такое количество адресов казалось чересчур большим, но потом при развитии сети и увеличении количества объектов практически все адресное пространство было задействовано.

Выделенный провайдером пул адресов можно назвать внешним или заре гистрированным. Иными словами, если в сеть Интернет отправить дейтаграмму с адресом отправителя из диапазона выделенных адресов и с корректным адре сом получателя, то последний может послать ответную дейтаграмму, и она благо получно дойдет до адресата. Однако администратор для своих внутренних сете вправе использовать специальные адреса, которые являются зарезервирован ми и не маршрутизируются в сети Интернет. Например, к таким адресам отн

сятся адреса из пула 192.168.*.*. Символ * указывает на то, что в данной позиции может быть любое приемлемое число (1–254). Использование подобных внутренних адресов предоставляет администратору очень удобную возможность распределять адреса так, как ему нравится. Например, если на одном из объектов корпоративной сети имеется не более десяти компьютеров, то для них можно выделить целиком сеть 192.168.2.0/24. При этом останется много неиспользуемых адресов, но так как данная сеть будет прочно ассоциироваться с данным объектом, то останется запас на будущее развитие.

Внутренние адреса не будут работать, если они будут установлены в качестве адресов отправителей дейтаграмм, уходящих в сеть Интернет. Точнее, дейтаграмма встретится с получателем (промежуточные маршрутизаторы будут проверять только целевой адрес) в сети Интернет, но вот ответа ждать не приходится. Интересно выполнить команду **tracert** с указанием любого адреса из внутреннего пула и посмотреть, как «далеко» дойдет дейтаграмма.

```
C:\>tracert 192.168.10.10
Трассировка маршрута к 192.168.10.10 с максимальным числом прыжков 30
1 <10 мс <10 мс <10 мс beetle.alternativa.spb.ru [192.168.3.254]
2 <10 мс <10 мс <10 мс 193.125.203.30
3 <10 мс <10 мс 10 ms StPetersburg-BM-1.Relcom.EU.net [212.113.96.141]
4 <10 мс 10 ms <10 мс StPetersburg-R-1.Relcom.EU.net [212.113.96.81]
5 <10 мс 10 ms <10 мс fa0-0-1.SPB-4.Relcom.net [193.125.15.134]
6 10 ms 20 ms 20 ms s1-0.M9-16.Relcom.net [193.124.254.41]
7 * * * Превышен интервал ожидания для запроса.
```

Маршрутизация дейтаграммы с адресом получателя 192.168.10.10 выполняется следующим образом. Сначала она отправляется к шлюзу по умолчанию в локальной сети (192.168.3.254 — в рассматриваемой сети это адрес внутреннего интерфейса защитного экрана с именем **beetle**), который, проверив свою таблицу маршрутизации и не найдя маршрута в целевую сеть, отправляет ее по адресу шлюза по умолчанию, настроенного на нем (193.125.203.30), и так продолжается до 6-го перехода, на котором, по всей видимости, дейтаграмма просто отбрасывается маршрутизатором. А теперь выполним ту же самую операцию, но для зарегистрированного адреса.

```
C:\>tracert www.rbc.ru
Трассировка маршрута к www.rbc.ru [194.186.36.154] с максимальным числом
  прыжков 30:
1 <10 мс <10 мс <10 мс beetle.alternativa.spb.ru [192.168.3.254]
2 <10 мс <10 мс <10 мс 193.125.203.30
3 <10 мс 10 ms <10 мс StPetersburg-BM-1.Relcom.EU.net [212.113.96.141]
4 <10 мс 10 ms <10 мс StPetersburg-R-1.Relcom.EU.net [212.113.96.81]
5 <10 мс 10 ms <10 мс fa0-0-1.SPB-4.Relcom.net [193.125.15.134]
6 10 ms 20 ms 20 ms s1-0.M9-16.Relcom.net [193.124.254.41]
7 311 ms 30 ms 741 ms M9-10.Relcom.Net [193.125.15.10]
8 10 ms 20 ms 21 ms cisco12.Moscow.ST.NET [193.232.244.43]
...
```

Видно, что дейтаграмма минует 6-й переход и маршрутизируется дальше до ее окончательного получателя.

В том случае, если бы вся распределенная сеть находилась за защитным эк раном, то, учитывая возможность реализации технологии NAT/PAT, можно бы бы обойтись всего одним зарегистрированным адресом. Принимая во вниман то, что в технологии PAT теоретически доступно более 64 000 портов для трансля ции, мы получаем в свое распоряжение поддержку такого же количества вну ренних пользователей. На практике, скорее всего, возникнут проблемы с мощ ностью защитного экрана.

На рис. 7.8 видно, что оба маршрутизатора Cisco 1720 находятся в зоне Интер нета, или иными словами, они полностью открыты для доступа извне. К порта Fast Ethernet (на рисунках порты обозначаются как feth0 — нулевой интерфей этих маршрутизаторов подключаются соответствующие локальные сети, которы защищаются с помощью защитных экранов. В принципе, при соответствующ настройке маршрутизаторы могут выполнять также роль и защитных экран но в рассматриваемой схеме на них возложена функция просто граничных марш рутизаторов, которые отвечают за поддержание доступа в Интернет. То ест учитывая тот факт, что провайдер выделил одну сеть класса С, каждому из дву объектов для доступа в сеть Интернет нужно присвоить свои блоки зарегистри рованных адресов. Одним из способов реализации этого является разделени сети 193.125.203.0/24 на подсети, которые затем назначаются объектам.

Покажем в деталях процесс разделения. Изначально, сеть класса С обладае некоторыми обобщенными характеристиками, представленными в табл. 7.2.

Таблица 7.2. Первичные показатели выделенной провайдером адресной сети

Характеристика	Значение
Сетевой адрес	193.125.203.0
Класс	C
Адрес подсети	193.125.203.0 (совпадает с адресом сети)
Маска подсети	255.255.255.0
Битовая маска	110nnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh
Число битов в поле адреса подсети	24
Число битов в поле адресов хостов	8
Количество хостов в подсети	254

При выделении подсетей следует оценить два важных показателя — сколько нужно подсетей и сколько доступных адресов должно быть в каждой (подробност в главе 2). Например, можно разбить сеть на две подсети, по 126 адресов в каждо При этом битовая маска подсети примет вид — 110nnnnn.nnnnnnnn.nnnnnnnn nhhhhhhh. То есть для выделения подсетей будет использован один бит (ста ший) из поля адресов узлов. В результате такого разделения администрато получит две подсети 193.125.203.0/25 и 193.125.203.128/25. В рассматриваем компании было принято решение о разбиении на 8 подсетей, по 30 адресов в к ждой. Такое количество адресов удовлетворяет насущным задачам и оставл

резерв на будущее, что важно. После разделения в распоряжении администратора появились следующие блоки адресов:

Подсеть	Маска	Диапазон адресов хостов
193.125.203.0	255.255.255.224	193.125.203.1 - 193.125.203.30
193.125.203.32	255.255.255.224	193.125.203.33 - 193.125.203.62
193.125.203.64	255.255.255.224	193.125.203.65 - 193.125.203.94
193.125.203.96	255.255.255.224	193.125.203.97 - 193.125.203.126
193.125.203.128	255.255.255.224	193.125.203.129 - 193.125.203.158
193.125.203.160	255.255.255.224	193.125.203.161 - 193.125.203.190
193.125.203.192	255.255.255.224	193.125.203.193 - 193.125.203.222
193.125.203.224	255.255.255.224	193.125.203.225 - 193.125.203.254

Процедура выделения такая же, как в примере из двух подсетей, за исключением того, что из поля адресов хостов задействуются три бита, а не один — 110nnnnn.nnnnnnnn.nnnnnnnn.nnnhhhhh. То есть допустимо ссылаться на одну из подсетей следующим образом — 193.125.203.128/27 (27 бит занимает поле адреса подсети).

После того как подсети были выделены, их можно назначать соответствующим объектам. К этому моменту было выполнено подключение еще двух объектов в дополнение к существующим (например, для подключения объекта 4 использовался маршрутизатор Cisco 1601, а для подключения объекта 3 — дополнительный модуль Ethernet, к которому подключаются модемы серии xDSL с портом Ethernet, а не V.35 — глава 10). На рис. 7.9 приведен возможный вариант схемы подсетей, хотя, по большому счету, все это зависит от предпочтений администратора.

В рассматриваемой сети маршрутизаторы не выполняют дополнительных функций (например, трансляции адресов) и, соответственно, с точки зрения расположения, например, объект 2 прозрачно доступен из сети Интернет. Аналогично и для других показанных на рисунке объектов. В результате, их локальные сети необходимо защищать с помощью определенных методов.

Например, можно установить защитный экран с двумя сетевыми адаптерами и настроить на нем технологию трансляции адресов. При этом нужно только учитывать, что в каждой подсети доступно до 30 адресов для задействования, из которых пара адресов выделяется для адреса интерфейса Ethernet маршрутизатора и внешнего интерфейса защитного экрана. В том случае, если используется технология NAT совместно с технологией PAT, защитный экран транслирует внутренние адреса отправителей в один адрес своего внешнего интерфейса и одновременно меняет номера портов на стороне отправителя. Здесь будет достаточно двух адресов и при разделении сети на подсети, таким образом, можно было бы в поле адресов хостов оставить только два бита — 110nnnnn.nnnnnnnn.nnnnnnnn.nnnnnnhh, отдавая 30 битов для поля адреса подсети. Тогда общее количество подсетей составило бы 64. Кстати, такое решение вполне применимо для назначения адресов синхронным интерфейсам маршрутизатора — если зарезервированные адреса нельзя использовать по каким-либо причинам.

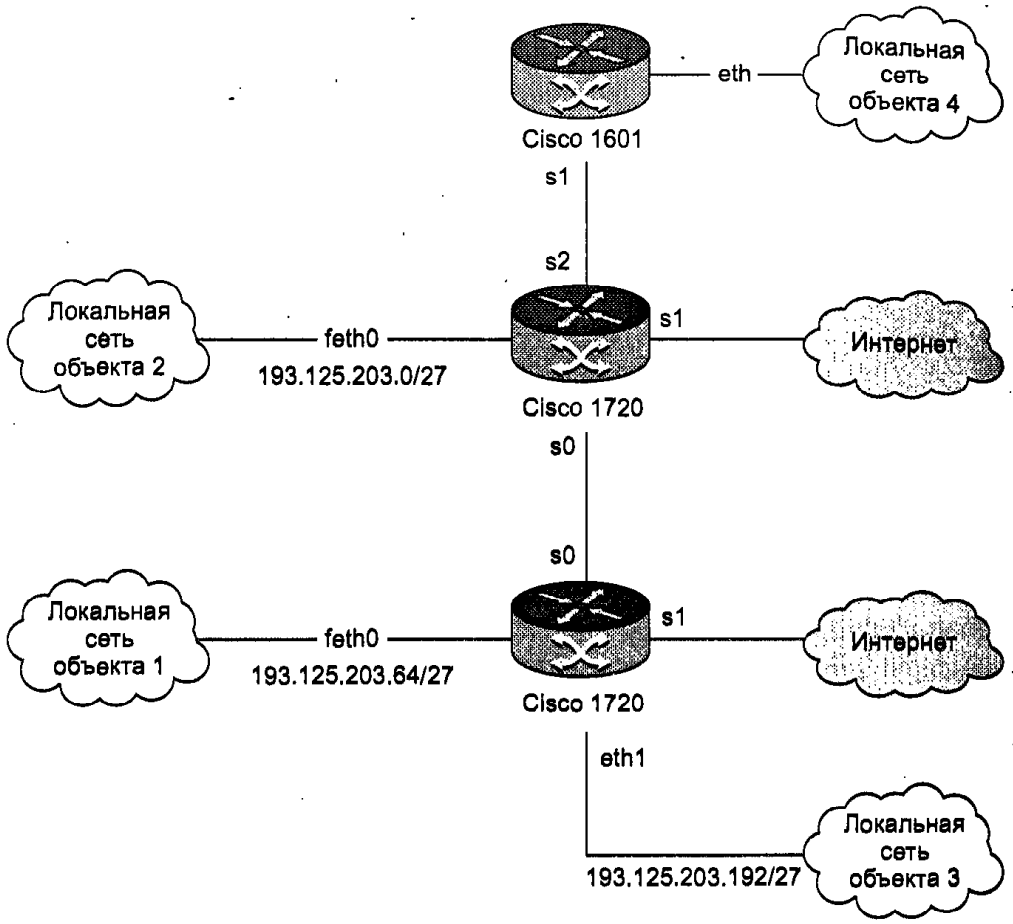


Рис. 7.9. Назначение подсетей разным объектам

Важно отметить, что для хостов, отправляющих извне данные в сеть 193.12 203.*, разделение на подсети абсолютно прозрачно. То есть, например, дейтагра ма с адресом получателя 193.125.203.100 поступает на интерфейс S1 маршрути затора Cisco 1720 (сейчас не важно, на какой именно из двух), а он, провер свою таблицу маршрутизации, определяет, на какой интерфейс нужно отправи дейтаграмму дальше, чтобы она достигла получателя. Ведь, несмотря на то, ч сеть была поделена на 8 подсетей, адреса остаются уникальными и не пересек ются друг с другом и факт разделения на подсети остается неизвестным отпр вителям в сети Интернет.

После того как каждой части корпоративной сети были назначены подсет приходит время назначить конкретные адреса нужным сетевым интерфейса Можно порекомендовать один раз подробно нарисовать схему распределения а ресного пространства, а потом постоянно обновлять ее — это пригодится при оп ративном решении различных проблем. Естественно, вопрос назначения адрес интерфейсам сетевых устройств полностью зависит от воли администратора, лучше придерживаться какой-то заранее отработанной схемы. Например, авт старается присваивать интерфейсам Ethernet граничных маршрутизаторов н большие адреса из перечня возможных в подсети. В нашем примере после значения адресов схема будет выглядеть, как это показано на рис. 7.10.

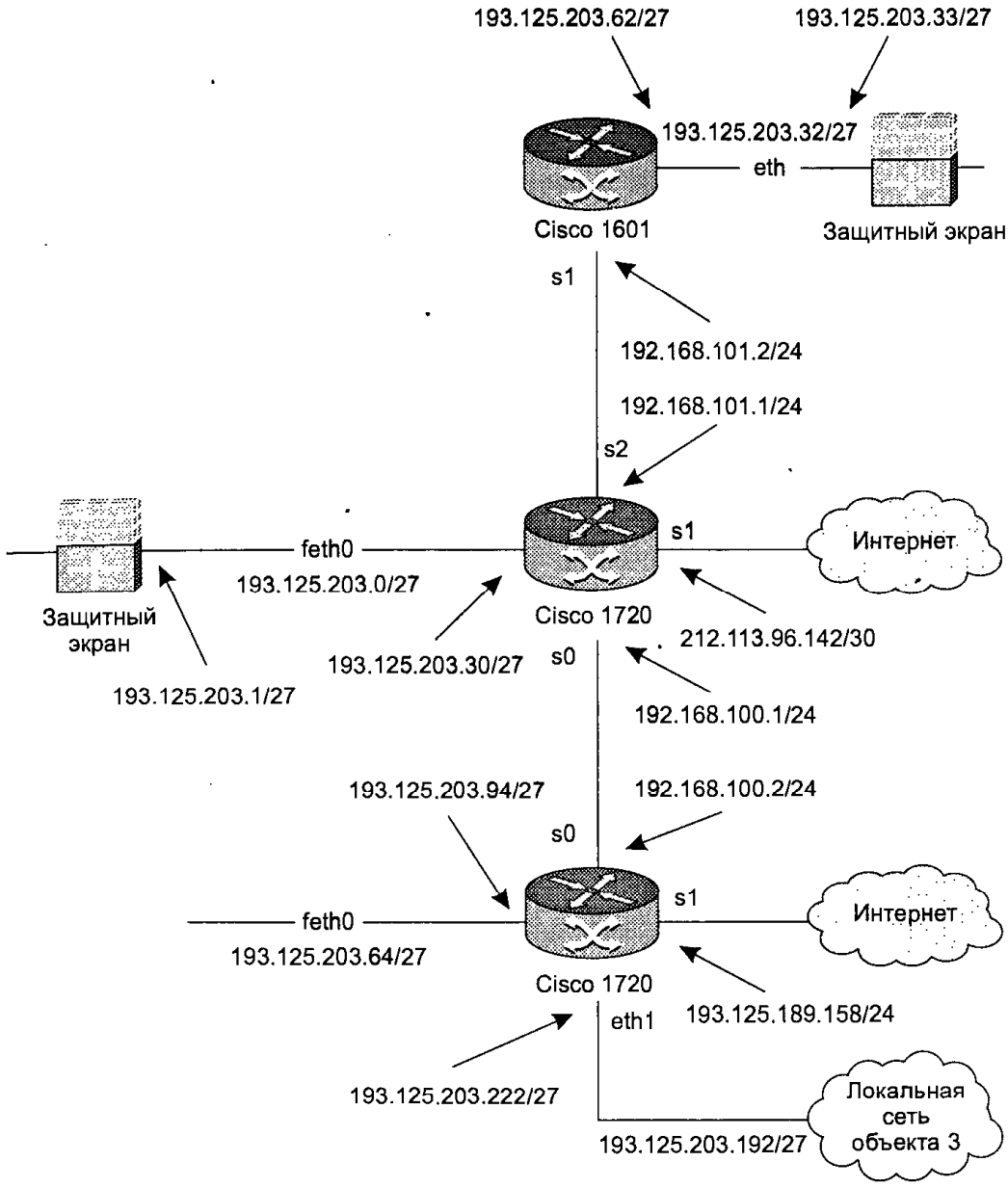


Рис. 7.10. Пример назначения адресов интерфейсам сетевых устройств

Адреса интерфейсов, непосредственно подключаемых к сети Интернет, как правило, назначаются провайдером услуг. Обратите внимание на то, что на каналах между маршрутизаторами используются адреса из сети класса С (192.168.100.0 и 192.168.101.0/24), принадлежащие к зарегистрированным адресам. Другим вариантом может быть реализация так называемого механизма безадресных интерфейсов с помощью команды ip unnumbered. В рассматриваемой сети задействован первый способ для корректной работы протокола маршрутизации OSPF.

Как уже было отмечено, для организации двух одновременно работающих каналов в сеть Интернет используется протокол OSPF. Обратимся еще раз к схеме организации связи. Для того чтобы оба канала были одновременно рабочими

и обеспечивали взаимное резервирование друг друга при доступе в Интерне
нужно сделать так, чтобы шлюз по умолчанию (default gateway) на каждом мар
рутизаторе менялся в ответ на изменение в сетевой топологии. Если то особо н
оговорено, при нормальной работе всех каналов (двух каналов в сеть Интерне
и каналов между маршрутизаторами Cisco 1720), шлюз по умолчанию на каждо
маршрутизаторе указывает на интерфейс S1, или, иными словами, на ближай
ший к нему маршрутизатор провайдера. В такой ситуации трафик, наприме
поступающий из локальной сети объекта 2 на интерфейс feth0 (Fast Ethernet 0
будет (если иных маршрутов не прописано) передаваться через интерфейс S
в сеть Интернет. Схема его дальнейшего движения не принципиальна, так ка
этот вопрос остается на «совести» провайдера (рис. 7.11).

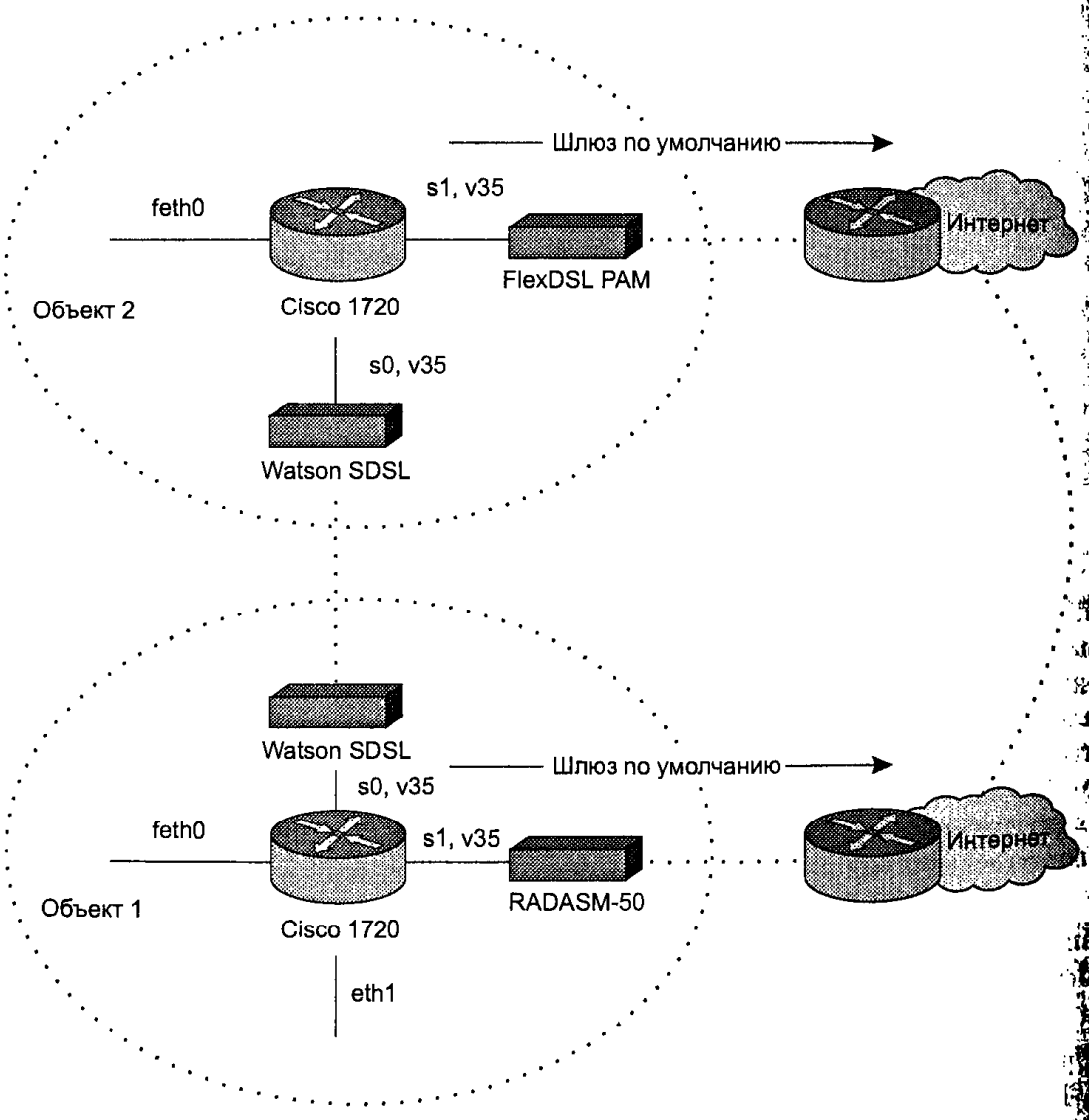


Рис. 7.11. Рассылка маршрутов по умолчанию

Теперь предположим, что канал в сеть Интернет с объекта 2 пропал. В резу
тате, трафик должен автоматически перенаправляться через интерфейс S0 на

логичный маршрутизатор, но уже на объекте 1, который и отправит его дальше в сеть Интернет. Причем подобное переключение должно производиться автоматически и как можно быстрее. Для этого и используется протокол маршрутизации OSPF, поддержка которого включена также и на стороне провайдера. Технические детали организации взаимодействия маршрутизаторов на двух разных площадках провайдера не суть важны, но с точки зрения логики получается некий контур из четырех маршрутизаторов и связующих их каналов. Если один из этих каналов перестает работать, то протокол маршрутизации фиксирует это и вносит соответствующие изменения в таблицу маршрутизации. К слову, протокол OSPF характеризуется довольно быстрым реагированием на изменение сетевой топологии.

Эксперименты показывают, что при пропадании одного из каналов в сеть Интернет маршрутизаторам требуется порядка 10 секунд (период прохождения сообщений keepalive) на то, чтобы перестроить свои таблицы маршрутизации. В течение этого времени доступ у клиентов, само собой, работать не будет, но это вовсе не большой срок простоя. Правда, к сожалению, для туннеля PPTP (глава 10) такого перерыва вполне достаточно, чтобы считать, что туннель виртуальной частной сети нарушен, и приступить к его восстановлению.

При искусственном отключении канала с объекта 2 в сеть Интернет на консоль маршрутизатора выдается сообщение:

```
1w2d: %OSPF-5-ADJCHG: Process 2121, Nbr 212.113.96.141 on Serial1 from FULL
   to DOWN, Neighbor Down: Dead timer expired (протокол маршрутизации ospf
   не получил сообщения от удаленного (адрес 212.113.96.141) маршрутизатора
   через интерфейс Serial1)
1w2d: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1, changed state
   to down (протокол канального уровня на интерфейсе Serial1 перешел
   в состояние down)
```

После обнаружения изменений в сетевой топологии таблица маршрутизации обновляется и теперь маршрут по умолчанию будет указывать на интерфейс Serial0.

```
cisco-1720-object-2#sh ip route
...
O*E2 0.0.0.0/0 [110/80] via 192.168.100.2, 00:00:18, Serial0
...
```

Однако когда канал восстанавливается вновь, для клиентов переключение на основной канал происходит незаметно. При тестировании этот момент удалось обнаружить только при наблюдении за изменением времени отклика для команды ping (добавился еще один переход и время ответа чуть-чуть возросло).

ИЗ ОПЫТА

При настройке работающего маршрутизатора, подключая его к работающему компьютеру через консольный порт, рекомендуется быть предельно осторожным, так как можно вывести из строя порт на маршрутизаторе. Автор столкнулся с подобной ситуацией, когда настраивал связь двух маршрутизаторов (Cisco 1601 и Cisco 1720) с одного компьютера через модемы FlexDSL с портами v35. Поскольку компьютер был один, а маршрутизаторов вдвое больше, приходилось поочередно подключаться к каждому из них. И в какой-то момент консольный порт на Cisco 1720 перестал работать. Этот маршрутизатор пришлось отдавать в ремонт. Правильнее было бы использовать на компьютере два последовательных порта с двумя кабелями, каждый из которых подключается к маршрутизатору.

8 Технология Frame Relay в построении распределенной сети

В этой главе рассматриваются теоретические и практические вопросы использования технологии Frame Relay, которая является одной из наиболее популярных в нашей стране. Отдельно будет обсуждена настройка управления трафиком, что может быть востребовано при наличии низкоскоростных соединений и интенсивного обмена информацией. Основное внимание будет уделено именно практической стороне работы Frame Relay.

Общие сведения о технологии Frame Relay

Так как в большинстве случаев организации пользуются услугами провайдера сети Frame Relay, администратору нет особой необходимости досконально знать всю подноготную технологии Frame Relay. Но понимать все параметры, определяющие предлагаемый сервис, методы оптимизации этих параметров он должен. А с учетом большого количества таких параметров задача определения сервиса становится достаточно сложной. И чтобы отношения с провайдером не доставляли массу хлопот, администратор должен иметь представление, какие моменты не следует упускать из виду при согласовании этих параметров и настройке своего граничащего с сетью Frame Relay устройства.

Как и многие другие средства и технологии связи, Frame Relay появилась в исследовательском подразделении Bell Labs компании AT&T. В 1988 году протокол Frame Relay был включен в стандарт ISDN в качестве рекомендации I.122 и утвержден подкомитетом по стандартам Международного консультативного комитета по телеграфии и телефонии (CCITT). К моменту появления окончательного варианта стандарта на технологию ISDN рекомендация I.122 превратилась в независимый протокол со своей практической областью применения.

Эта технология разрабатывалась с учетом высокоскоростной передачи данных и низкого уровня ошибок современных сетевых средств. Первые сети с коммутацией пакетов были рассчитаны на скорость передачи 64 Кбит/с, в то время как сети Frame Relay предназначались для гораздо больших скоростей. Достичь повышения скорости передачи помогло исключение накладных расходов, которые неизбежны при контроле ошибок.

Издержки при пакетной коммутации вызываются контролем вызовов, поиском ошибок и контролем потоков. В технологии X.25, которая в настоящее время практически не используется, но ранее была довольно популярна, пакеты управления вызовами, используемые для установления и разрыва виртуальных соединений, переносятся по тому же соединению, что и пакеты данных. Фактически вся передача сигналов осуществляется по основному каналу (так называемая передача in-band).

В сети Frame Relay передача сигналов контроля вызова осуществляется по виртуальному соединению (Virtual Circuit), отличному от используемого для передачи пользовательских данных. В пользовательском интерфейсе один канал управления соединением служит для контроля всех коммутируемых соединений передачи данных. Так как в настоящее время провайдеры услуг Frame Relay предлагают в большинстве своем только постоянные виртуальные соединения (Permanent Virtual Circuit, PVC), то промежуточным коммутирующим узлам нет необходимости поддерживать таблицы состояний или обрабатывать управляющие вызовы для каждого соединения в отдельности.

Наиболее очевидно преимущество Frame Relay над X.25 в управлении потоками и контроле ошибок. Технология X.25 задействует физический, канальный и сетевой уровни, то есть три нижних уровня эталонной модели OSI. На канальном уровне осуществляется контроль ошибок в транзитных узлах сети с коммутацией пакетов. При этом каждому узлу присваивается порядковый номер. После проведения контроля одновременно с передачей данных на следующий узел предыдущему передается подтверждение приема. На сетевом уровне происходит мультиплексирование нескольких потоков данных разных виртуальных соединений в единый поток к пользователю. Для этого каждый передаваемый кадр имеет свой номер виртуального соединения, который используется для маршрутизации и коммутации трафика. Управление потоком и контроль ошибок на всем пути следования пакетов от отправителя до получателя осуществляются при помощи схемы нумерации сетевого уровня.

Основное различие между технологиями Frame Relay и X.25 состоит именновмеханизме коррекции ошибок. Так как технология X.25 разрабатывалась более 20 лет назад для передачи данных через аналоговые каналы связи, которые характеризовались плохим качеством, то требовались различные механизмы коррекции ошибок и алгоритмы повторной передачи потерянных данных.

В соединениях Frame Relay мультиплексирование осуществляется на канальном уровне модели OSI, а контроль ошибок и управление потоком вовсе отсутствуют. Каждый кадр канального уровня содержит номер логического соединения, который используется для маршрутизации и коммутации трафика. Порядковые номера для управления потоком и контроля ошибок не используются. При этом контроль правильности передачи данных от отправителя получателю должен осуществляться на более высоком уровне модели OSI (например, протоколом TCP).

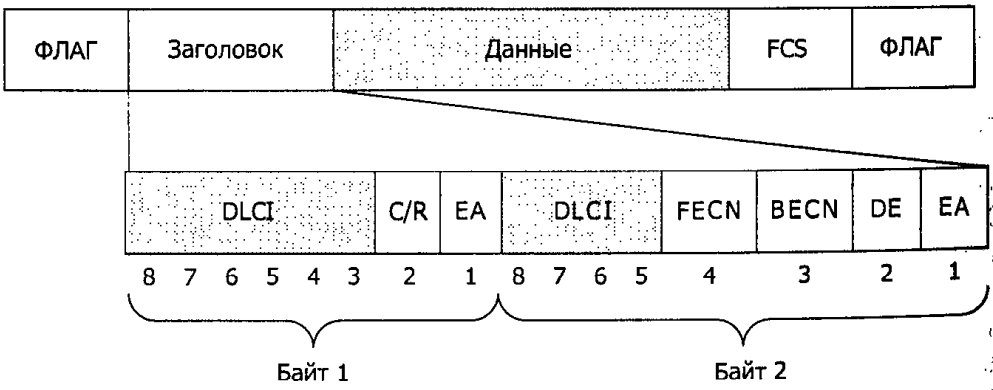
Как и технология X.25, Frame Relay выполняет статическое мультиплексирование передаваемых кадров с данными от различных отправителей и направляет их через один канал связи. При этом могут поддерживаться скорости передачи между 56 Кбит/с и 45 Мбит/с (в то время как скорость сети X.25 составляет 56 или 64 Кбит/с). Другой возможностью, унаследованной от технологии X.25,

является механизм передачи, ориентированный на предварительное установ ние виртуального соединения между взаимодействующими абонентами.

Причиной того, что технология Frame Relay столь стремительно заним нишу локальных сетей, является ее экономическая эффективность. Frame Re не требует построения новой коммуникационной инфраструктуры. Обычно в что требуется, — это программное обновление существующих маршрутизиру щих систем или незначительная модернизация программно-аппаратного обес чения систем коммутации кадров X.25.

Технология Frame Relay имеет много общего с ATM (Asynchronous Trans Mode — асинхронный режим передачи). Основное различие между ними на уро не организации блоков информации состоит в том, что в первой длина кад переменна, а во второй — постоянна и равна 53 байт. Большинство современ сетей Frame Relay рассчитаны на максимальную длину кадра в 1024 байт, из торых от 6 до 8 байт занимают служебные данные.

Структура кадра передачи данных в сетях Frame Relay достаточно про (рис. 8.1). Данные помещаются между полем заголовка (часто все поле загол ка называется DLCI — Data Link Connection Identifier, идентификатор свя с источником данных), которое может занимать от 2 до 4 байт, и полем контро ной суммы в конце кадра (FCS), призванного обнаруживать любые битовые оши ки. Числовое значение, записанное в полях DLCI, служит для идентифика виртуального соединения между абонентами. В одном кадре может быть пере но до 8000 байт пользовательских данных.



DLCI — Data Link Connection Identifier
C/R — Command/Response Field Bit (не изменяется внутри сети Frame Relay)
FECN — Forward Explicit Congestion Notification
BECN — Backward Explicit Congestion Notification
DE — Discard Eligibility Indicator
EA — Extension Bit (бит, указывающий на больший заголовок)

Рис. 8.1. Формат кадра Frame Relay

Поскольку можно передавать как короткие, так и очень большие кадры, с ствует вероятность того, что большие кадры вызовут увеличенную задер между передачей двух коротких кадров. Однако несмотря на это обстоятел во, следующие несколько лет технология Frame Relay будет играть важную

в локальных сетях, хотя ее использование для типичных широкополосных служб с изменяющимся профилем трафика ограничено.

С уверенностью можно сказать, что сегодня технология Frame Relay — это зрелая и чрезвычайно популярная, в том числе и в нашей стране, технология создания сетевой магистрали. Ее основное назначение состоит в выполнении роли связующего звена между узлами корпоративной сети. При построении корпоративной сети на базе технологии Frame Relay, как правило, рассматриваются три основных варианта ее организации.

1. Частная сеть на базе выделенных линий. Организация арендует линии связи и приобретает необходимое оборудование (например, коммутаторы, маршрутизаторы или мультиплексоры). Построенная на их базе сеть является ее собственностью и находится под полным контролем и управлением.
2. Виртуальная частная сеть. Организация покупает услуги сетей Frame Relay у компаний, предоставляющих телекоммуникационные услуги. Таким образом, организация создает частную корпоративную сеть, обращаясь к услугам сетей Frame Relay общего пользования, и осуществляет полный контроль над сетью и административное управление ею.
3. Соглашение с внешней организацией о создании и управлении сетью. Существующая корпоративная сеть передается телекоммуникационной компании, которая проводит административное управление этой сетью в интересах фирмы-клиента, а также предоставляет услуги связи, оборудование и реализует поддержку сети. За рубежом существует ярко выраженная рыночная тенденция к таким соглашениям; на данной основе в мире функционирует 30 % корпоративных сетей. Это обусловлено в большинстве случаев неспособностью организации самостоятельно управлять сложной по эксплуатации корпоративной сетью.

Однако независимо от способа создания корпоративной сети администратору необходимо иметь представление о средствах и приемах управления такой сетью, способах оптимизации режимов работы. Причем технология для магистрали корпоративной сети должна отвечать трем основным требованиям.

1. **Надежность**. В случае если организация не владеет всей сетью, а пользуется услугами телекоммуникационных компаний, то соединение, скажем, между офисами характеризуется некоторой неопределенностью из-за отсутствия контроля над линиями связи. Технология Frame Relay предоставляет достаточную отказоустойчивость благодаря обеспечению динамической перемаршрутизации в случае отказа постоянного виртуального соединения. Физически сети Frame Relay образованы соединением коммутаторов в ячеистую структуру. Одно из преимуществ такой структуры состоит в том, что она обеспечивает хорошую степень отказоустойчивости. Если выходит из строя используемый постоянным виртуальным соединением коммутатор, то соседний коммутатор создаст новый путь для этого соединения. В результате характеристики передачи могут лишь несколько ухудшиться. Другим достоинством такой структуры является возможность направлять кадры в обход перегруженных коммутаторов.

2. **Доступность**. Frame Relay является вполне доступной со стороны локальных сети, так как поддерживает пакетный трафик. Frame Relay позволяет передавать кадры относительно большого размера, которого достаточно для пакетов Ethernet и Token Ring (максимальная длина для них составляет 1500 и 4096 байт соответственно). Благодаря этому пакеты не надо фрагментировать при передаче, соответственно Frame Relay лишена накладных расходов на сегментацию и сборку.
3. **Удобство**. Соединение по глобальной сети должно поддерживать различные скорости передачи исходящего и входящего трафика. Ввиду того что входящий трафик к узлу локальной сети отличается от исходящего трафика, глобальная сеть должна поддерживать асимметричный режим работы. Кроме того, для обеспечения необходимого управления потоком технология глобальной сети должна поддерживать стек протоколов TCP/IP на верхних уровнях модели OSI. Технология Frame Relay отвечает этим требованиям за счет возможности предоставления асимметричных виртуальных соединений.

Для защиты от сбоев на уровне узла провайдеры Frame Relay часто предлагают запасное постоянное виртуальное соединение. При этом оно имеет существенно меньшую оговоренную скорость передачи (Committed Information Rate CIR), чем основное. В экстренных случаях запасное виртуальное соединение будет активизировано практически немедленно.

CIR — минимальная полоса пропускания, гарантированная каждому постоянному или коммутируемому виртуальному соединению. Эта скорость (измеряется в битах в секунду) заявляется абонентом в соответствии с объемом данных, которые он собирается передавать по сети, и гарантируется провайдером услуги Frame Relay (обычно этот параметр отмечается в договоре на предоставление услуги).

Скорость может варьироваться от 16 Кбит/с до 44,8 Мбит/с. Если скорость отправки кадров не превосходит скорость порта подключения абонента (так называемую AR — Access Rate) и пропускная способность сети Frame Relay в данный момент имеет резерв, то абонент может превысить согласованное значение CIR. Скорость, с которой абонент посылает данные при наличии достаточной пропускной способности, называется oversubscription rate (то есть скорость, превышающая ранее оговоренную).

Предположим, что организация получила выделенный канал, имеющий максимальную скорость передачи 128 Кбит/с, но заключила договор на CIR в 64 Кбит/с, то есть канал позволяет передавать пакеты со скоростью 128 Кбит/с, но сеть не дает гарантии на успешность такой передачи. Кадры, которые передаются на скорости большей, чем CIR (в данном случае 64 Кбит/с), называются избыточными кадрами и помечаются специальным битом (бит DE — Discard Eligibility) в заголовке. Эти кадры могут быть доставлены адресату при наличии на данный момент неиспользуемой части полосы пропускания. Поскольку провайдер не гарантирует передачу информации с большей скоростью, чем оговоренная, используется метод справедливого сброса кадров. При возникновении перегрузки коммутаторы Frame Relay будут в первую очередь отбрасывать имеющие кадры с установленным битом DE. Это, забегая вперед, можно показать на

альном примере, запрашивая статистику по постоянному виртуальному соединению на маршрутизаторе Cisco Systems:

```
Router#sh fr pvc
PVC Statistics for interface Serial1 (Frame Relay DTE)

DLCI = 18, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial1.1

    input pkts 785864      output pkts 673709       in bytes 106316635
    out bytes 420859843    dropped pkts 134         in FECN pkts 0
    in BECN pkts 0         out FECN pkts 0          out BECN pkts 0
    in DE pkts 12361       out DE pkts 0
    out bcast pkts 96186   out bcast bytes 15421788
    pvc create time 5w5d, last time pvc status changed 02:42:20
```

В результате работы команды присутствует статистика полученных по виртуальному соединению кадров Frame Relay с установленным битом DE (in DE pkts 12361). Эта информация может служить неким измерителем загруженности магистральной сети провайдера Frame Relay, так как позволяет оценить, с каким увеличением скорости относительно CIR сеть может принять данные на выбранном промежутке времени. Кроме того, следует обратить внимание на параметр dropped pkts, указывающий количество пакетов, которые были отброшены маршрутизатором на уровне Frame Relay, например по причине перегрузки постоянного виртуального соединения. Для того чтобы очистить статистику, можно воспользоваться командой clear counters.

Важно отметить, что из-за того, что пользователям позволено передавать кадры с превышением скорости CIR, может оказаться так, что сеть Frame Relay будет не в состоянии обработать кадры всех пользователей, если многие из них станут передавать информацию одновременно.

Когда коммутатор в сети Frame Relay испытывает перегрузку, он начинает вставлять биты явного оповещения о перегрузке для отправителя BECN (Backward Explicit Congestion Notification) и получателя FECN (Forward Explicit Congestion Notification) в кадры Frame Relay. Установленный бит BECN информирует устройство отправителя (маршрутизатор) о возникновении перегрузки в виртуальном соединении на пути к получателю. Получив предупреждение, устройство отправителя уменьшает скорость передачи или даже временно приостанавливает передачу, в зависимости от выбранной политики управления трафиком. Если говорить об информировании получателя, то последний может предпринимать лишь некоторые косвенные действия в ответ на состояние перегрузки.

Рассматриваемый механизм извещений имеет свои недостатки. Во-первых, он должен быть явно включен (для оборудования Cisco Systems). Во-вторых, важно знать время и тип реакции оборудования пользователей на эти сообщения, а также знать, будет ли оно вообще реагировать. Но на перегрузку надо реагировать в любом случае, иначе ситуация может только ухудшаться. По этой причине введен механизм пометки кадров, которые будут отбрасываться с момента наступления перегрузки до момента срабатывания аппаратуры пользователей. Но и здесь есть свои проблемы. Вероятность того, что будут отбрасываться только кадры, помеченные битом DE, в разных сетях различна. Это связано с тем,

что для поддержки ожидаемого уровня трафика каждый провайдер строит сво сеть по-своему. Ввиду отсутствия стандартов на архитектуру сети или управлен перегрузками, таковые могут возникнуть при разных уровнях интенсивност трафика в зависимости от возможностей и особенностей коммутаторов, пр пускной способности каналов связи между ними и топологии сети. Таким образо при выборе провайдера и заключении с ним договора необходимо иметь предста ление о том, какой гарантированный процент пакетов кадров не будет отброше и достигнет получателя. Сеть Frame Relay может отбрасывать любые кадры, п этому необходимо знать процентное соотношение отбрасываемых помеченны и не помеченных битом DE кадров. Эту информацию следует получить у прова дера или определить самим, используя методы управления и мониторинга сет

Сервис сети Frame Relay характеризуется тремя основными параметрами: пр пускной способностью CIR, гарантированным объемом кадров Bc (Committ Burst size) и Be (Excess Burst). Каждый из этих параметров привязан к опред ленному интервалу времени (T). Как указывалось выше, CIR — это скорос с которой провайдер обязуется передавать данные за период времени T. *Bc* пр ставляет собой максимальное число битов, которое сеть может передавать время T, а *Be* — это максимальное число битов сверх Bc, которое сеть способ в принципе принять за время T. Соотношение между T, Bc и CIR следующее

$$T = Bc / CIR$$

Если принять интервал времени равным одной секунде, как это делают м гие провайдеры, то в предыдущем равенстве значение CIR равно Bc. Несмот на то что большинство провайдеров используют длительность в одну секун интервалы времени не стандартизованы.

Выбор конкретного значения того или иного параметра целиком зависит тех целей, которые преследует организация, используя сеть Frame Relay. Нап мер, для передачи чувствительной к задержкам информации необходима сеть встроенной организацией очередей и гарантированной задержкой, а также с вольно высокими значениями CIR в расчете на виртуальное соединение. Е же сеть используется главным образом для передачи обычных данных, задер не является критичным параметром. В этом случае высокоскоростной ка с малым или даже нулевым значением CIR способен обеспечить необходим пропускную способность.

Настройка Frame Relay на маршрутизатора Cisco Systems

Базовая настройка маршрутизатора при его подключении к сети Frame Relay п полагает выполнение следующих простых шагов. На первом этапе требуется брать интерфейс, на котором будут производиться настройки, и перейти в ре конфигурирования этого интерфейса. Далее указывается тип инкапсуля данных. Выбор осуществляется посредством выполнения команды encapsu frame-relay, а в качестве ее параметра указывается собственно тип инкапсул

Им может быть тип cisco (значение по умолчанию) или ietf. Первый тип используется, если на другом конце соединения осуществляется подключение к оборудованию компании Cisco Systems.

СОВЕТ

Перед изменением типа инкапсуляции компания Cisco Systems рекомендует принудительно отключить интерфейс с помощью команды shutdown.

Далее следует указать тип протокола LMI (Local Management Interface — интерфейс локального управления), который настроен на установленных у провайдера коммутаторах Frame Relay. Тип инкапсуляции и тип протокола LMI должны сообщаться представителями провайдера на этапе начальной настройки оборудования.

Настройка LMI осуществляется с помощью команды frame-relay lmi-type с указанием необходимого протокола, которым может быть ansi, cisco или q933i. При использовании операционной системы Cisco IOS версии 11.2 и старше не обязательно уточнять тип LMI, так как маршрутизатор умеет определять его автоматически. В следующем примере интерфейс Serial1 маршрутизатора подключен к оборудованию провайдера услуг Frame Relay, и на нем указывается тип инкапсуляции (frame-relay) и тип используемого протокола LMI (ansi).

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial1
Router(config-if)#no ip address
Router(config-if)#encapsulation frame-relay
Router(config-if)#frame-relay lmi-type ansi
```

ПРИМЕЧАНИЕ

Если при настройке синхронного интерфейса администратор дал команду не использовать сообщения keepalive, то протокол LMI остановится. Напомним, что сообщения keepalive посылаются через соединение для подтверждения его активности.

Следует отметить, что корректная настройка протокола LMI является одним из условий правильной работы с сетью Frame Relay. В случае возникновения проблем со связью рекомендуется, в числе прочего, проверить функционирование этого протокола с помощью команды show frame-relay lmi:

```
Router#sh fr lmi

LMI Statistics for interface Serial1 (Frame Relay DTE) LMI TYPE = ANSI
   Invalid Unnumbered info 0          Invalid Prot Disc 0
   Invalid dummy Call Ref 0           Invalid Msg Type 0
   Invalid Status Message 0           Invalid Lock Shift 0
   Invalid Information ID 0           Invalid Report IE Len 0
   Invalid Report Request 0           Invalid Keep IE Len 0
   Num Status Enq. Sent 182568        Num Status msgs Rcvd 182535
   Num Update Status Rcvd 0           Num Status Timeouts 33
```

В перечне отображаемых параметров следует обратить внимание на два: N
Status Enq. Sent и Num Status msgs Rcvd, которые показывают количество отпра
ленных и полученных маршрутизатором статусных сообщений протокола LM
Значения этих параметров при нормальной работе должны увеличиваться с ра
ной скоростью.

После того как выполнена настройка маршрутизатора на работу с технолог
ей Frame Relay, администратор сети может проверить корректность выполне
ных настроек с помощью команды show. Так, например, команда show interfac
покажет информацию о типе инкапсуляции, статусе первого и второго уровн
модели OSI и т. п.

```
Router#show interfaces s1
Serial1 is up, line protocol is up
   Hardware is QUICC Serial
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
   LMI enq sent  9570, LMI stat recvd 9563, LMI upd recvd 0, DTE LMI up
   LMI enq recvd 0, LMI stat sent  0, LMI upd sent  0
   LMI DLCI 0  LMI type is ANSI Annex D  frame relay DTE
...
```

Выполнение рассмотренных команд позволяет настроить на определенно
интерфейсе технологию Frame Relay и проверить ее функционирование. Но дл
начала практической работы этого будет недостаточно. Необходимо настроит
на выбранном интерфейсе подинтерфейсы, которые будут представлять собо
отдельные виртуальные соединения. Для этого очень важно четко представлят
логическую топологию объектов.

Технология Frame Relay позволяет связывать удаленные объекты разнообра
ными способами. Например, может быть создана логическая звезда, когда в це
тре выделяется главный офис, а все удаленные офисы подключаются к нем
с помощью постоянных виртуальных соединений. С точки зрения затрат (к
административных, так и финансовых) это наиболее выгодная топология, так к
она требует наименьшего количества соединений для своей реализации. В це
лях повышения надежности могут существовать дополнительные связи межд
некоторыми удаленными филиалами, но следует учесть, что дополнительны
виртуальные соединения придется оплачивать. В случае топологии звезды
одном физическом интерфейсе центрального маршрутизатора будет настроен
несколько виртуальных соединений.

Другим вариантом может быть полносвязная топология, при которой в
объекты связаны друг с другом. Такой подход, несмотря на большие затрат
(как прямые, так и косвенные — административные), обеспечивает наикратча
шую связь между объектами без промежуточных переходов между ними. О
видно, что при увеличении количества связываемых объектов сложность с
возрастает. Можно определить общее количество соединений в полносвяз
топологии при помощи формулы: (n(n – 1))/2, где n — это количество свя
ваемых объектов.

ИЗ ОПЫТА

Полносвязная топология характеризуется повышенной надежностью. Но это — теория. А что же происходит на практике? Если говорить о принятом у нас подходе (в частности, в Санкт-Петербурге), то подключение к сети Frame Relay осуществляется следующим образом. Провайдер запрашивает адрес объекта, который планируется подключить к сети. При этом часто просят дать информацию о наличии на объекте телефона и о его номере. Это необходимо для того, чтобы уточнить, через какую АТС проще организовать подключение. Затем провайдер формирует прямой провод от этой АТС до объекта. Причем ввиду нехватки магистральных кабелей вполне возможен вариант, когда занимается какой-либо телефонный номер. Далее провайдер ставит свое оборудование и предоставляет разъем V.35 (типа female) для подключения маршрутизатора клиента. Собственно говоря, на этом процедура физического подключения и заканчивается. Клиент не знает, какая топология получилась в результате. Затем провайдер у себя на магистральных коммутаторах Frame Relay настраивает запрошенные виртуальные соединения межу объектами и предоставляет номера DLCI для дальнейшей настройки на сторонах маршрутизаторов клиента. При этом несколько виртуальных соединений будут организованы на одном прямом проводе. Если теперь говорить о надежности, то следует помнить, что обрыв этого прямого провода приведет к тому, что все виртуальные соединения, замкнутые на объект, становятся неактивными. Если средства позволяют и надежность является критичным фактором, то можно привлечь еще одного провайдера услуг Frame Relay, который организует дополнительное физическое подключение.

По умолчанию интерфейсы, которые поддерживают технологию Frame Relay, являются групповыми (multipoint). Подобный тип соединения не создает проблем, если работает только одно постоянное виртуальное соединение на одном интерфейсе. Однако могут возникнуть проблемы, когда множество постоянных виртуальных соединений настраивается на одном физическом интерфейсе. В этой ситуации сообщения протоколов маршрутизации, поступающие на маршрутизатор, не могут широковещательно рассылаться на удаленные объекты (рис. 8.2). Это связано с тем, что технология Frame Relay не поддерживает рассылку широковещательных сообщений, так как принадлежит к сетям NBMA (Non-Broadcast Multiple Access — нешироковещательные сети множественного доступа).

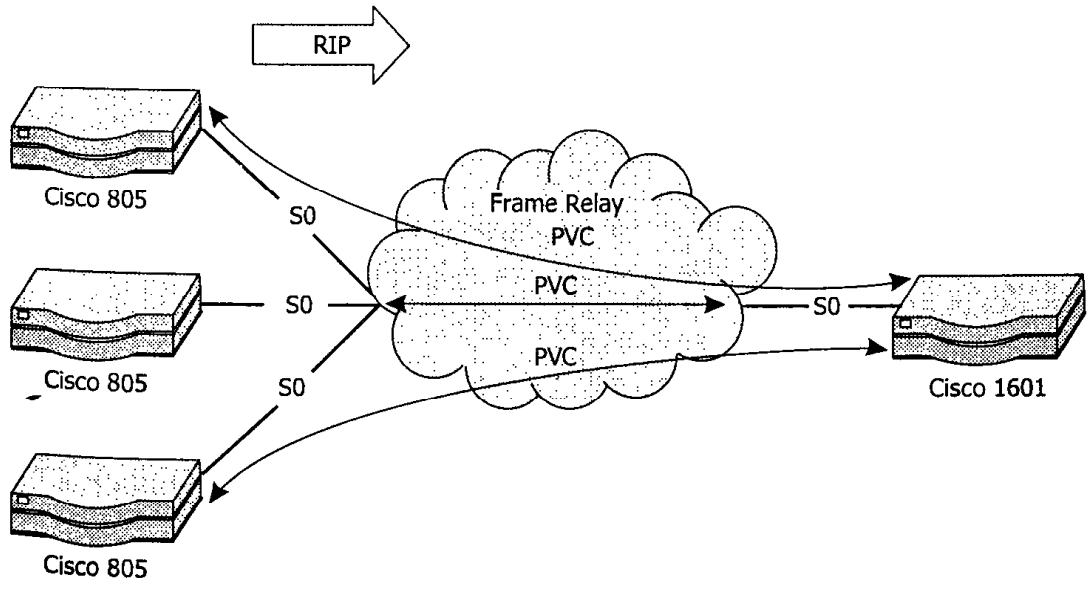


Рис. 8.2. Обновление маршрутов в сети Frame Relay

Рассмотрим пример фрагмента сети, в которой три удаленных маршрутиза-
тора подключаются к маршрутизатору в центральном офисе по технологии Fra
Relay с помощью постоянных виртуальных соединений. Алгоритм расщеплен
горизонта (split horizon — глава 7) будет препятствовать рассылке централ
ным маршрутизатором сообщений протокола маршрутизации через тот же ин
терфейс, через который они поступили. Удаленные объекты никогда не получ
сообщений протокола маршрутизации друг от друга и от центрального маршру
тизатора.

Таким образом, возникают два ограничения на использование множества по
стоянных виртуальных соединений на одном интерфейсе — технология расщепл
ния горизонта и поддержка широковещательных сообщений. Однако широковещ
ние не создает проблем, если существует только одно постоянное виртуальн
соединение на интерфейсе, так как это простое соединение типа «точка-точка».

Далее, если маршрутизатор поддерживает множество виртуальных соедин
ний на одном физическом интерфейсе, то это приводит к тому, что он должен
рассылать обновления маршрутов для протоколов маршрутизации на каждо
виртуальное соединение. Ожидаемый результат — их чрезмерная загрузка. О
ним из решений проблемы расщепления горизонта являются подинтерфейсы,
которые рассматриваются ниже.

Казалось бы, наиболее простым решением проблем, связанных с технологи
расщепления горизонта, могло быть простое отключение этого механизма, одн
ко и здесь существуют «подводные камни». Во-первых, означенный механи
можно отключить только для протокола IP, а для протоколов IPX и AppleTa
нельзя. Во-вторых, с повышенной вероятностью можно ожидать появления п
тель (loops) маршрутизации в сети. Для того чтобы все же пользоваться воз
можностью передачи широковещательных сообщений об обновлении маршр
тов в сети, администратор сети может настроить подинтерфейсы (subinterfaces
которые являются логическими компонентами физического интерфейса мар
рутизатора. В распределенной сети, использующей механизм расщепления г
ризонта, сообщения обновления маршрутов, полученные на один подинтерфей
могут быть переданы на другие подинтерфейсы. При этом каждое постоянн
виртуальное соединение можно настроить как соединение типа «точка-точка»
что позволяет подинтерфейсу работать по аналогии с выделенным каналом с
зи с гарантированной скоростью CIR.

Главная цель применения подинтерфейсов в том, чтобы протоколы маршр
тизации, в основе которых лежит алгоритм длины вектора, могли корректно р
ботать в среде совместно с механизмом расщепления горизонта. Маршрути
тор, получая, например, широковещательный пакет на подинтерфейс Serial
способен передать ту же информацию об обновлении маршрутов далее, чер
интерфейсы Serial0.2 и 0.3.

Администратор сети имеет право настроить подинтерфейсы для поддерж
следующих типов соединений: «точка-точка» (point-to-point) и «точка-груп
(multipoint). В первом варианте один подинтерфейс поддерживает постоян
виртуальное соединение с удаленным маршрутизатором. В соединении «т
ка-группа» подинтерфейс используется для установления множества посто
ных виртуальных соединений с удаленными маршрутизаторами.

Для того чтобы настроить подинтерфейс на физическом интерфейсе, следует выполнить следующие шаги.

1. Выбрать интерфейс, на котором планируется настраивать подинтерфейсы, а затем войти в режим его настройки:

```
Router#conf term
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial1
```

2. Удалить все адреса сетевого уровня (например, адреса протокола IP), назначенные этому интерфейсу. Если физический интерфейс имеет адрес, то подинтерфейсы корректно функционировать не будут.
3. Настроить инкапсуляцию Frame Relay, как было показано ранее (команда Router(config-if)#encapsulation frame-relay).
4. Выбрать подинтерфейс для последующей настройки. В рассматриваемом примере конфигурируется подинтерфейс Serial1.2, являющийся интерфейсом «точка-точка»:

```
Router(config)#interface serial1.2 point-to-point
Router(config-subi)#
```

Обратите внимание, что строка приглашения маршрутизатора изменилась и указывает на то, что работа ведется в режиме подинтерфейса (приглашение Router(config-subi)#).

1. Настроить адрес сетевого уровня на подинтерфейсе. В нашем случае интерфейс типа «точка-точка» и мы планируем использовать протокол IP. В такой ситуации можно сохранить адресное пространство, задействуя команду ip unnumbered <имя интерфейса>. В качестве параметра можно указать интерфейс Ethernet0:

```
Router(config-subi)#ip unnumbered Ethernet0
```

2. Далее нужно настроить на подинтерфейсе номер DLCI, который обычно предоставляется провайдером услуги Frame Relay для обоих концов постоянного виртуального соединения, с помощью команды frame-relay interface-dlci с параметром в виде номера DLCI:

```
Router(config-subi)#frame-relay interface-dlci 22
```

Если на маршрутизаторе уже настроено постоянное виртуальное соединение с аналогичным номером, то маршрутизатор выдаст сообщение об ошибке: %PVC already assigned to sub-interface Serial1.2. Указание номера DLCI является обязательным условием для технологии Frame Relay с учетом того факта, что протокол LMI не располагает никакой информацией о подинтерфейсах.

После выполнения приведенного набора команд настройку технологии Frame Relay можно считать завершенной. В большинстве случаев их вполне достаточно для того, чтобы начать работу. Выполненные настройки можно проверить с помощью команды show running-config, результат выполнения которой должен походить на следующий пример:

```
...
interface Serial1
   no ip address
   encapsulation frame-relay
   frame-relay lmi-type ansi
!
interface Serial1.1 point-to-point
   ip unnumbered Ethernet0
   frame-relay interface-dlci 18
!
interface Serial1.2 point-to-point
ip unnumbered Ethernet0
   frame-relay interface-dlci 22
...
```

В этом примере к сети Frame Relay физически подключен интерфейс Serial, на котором настроены два подинтерфейса — Serial1.1 и Serial1.2 со своими номерами DLCI, то есть каждый подинтерфейс соответствует отдельному постоянному виртуальному соединению. Следует отметить, что с точки зрения протоколов маршрутизации подинтерфейсы рассматриваются как обычные интерфейсы, при помощи которых можно достичь определенных получателей (в этом можно убедиться, выполнив команду **show ip route**). Администратору сети также доступна команда **show interface** с номером подинтерфейса в качестве параметра, выдающая следующий результат:

```
Router#show interfaces S1.1
Serial1.1 is up, line protocol is up
   Hardware is QUICC Serial
   Interface is unnumbered. Using address of Ethernet0 (192.168.4.254)
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation FRAME-RELAY
```

В распоряжении администратора сети имеется также команда **show frame-relay pvc**, которая выводит статус каждого настроенного постоянного виртуального соединения и статистику по нему. В рассматриваемом далее примере эта команда выполняется вместе с указанием номера DLCI виртуального соединения Frame Relay. Информация выводится только для этого соединения, а без указания номера она будет расширена для всех настроенных виртуальных соединений.

```
Router#show frame-relay pvc 18
PVC Statistics for interface Serial1 (Frame Relay DTE)
DLCI = 18, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial1.1

input pkts 10759          output pkts 11160         in bytes 2465757
out bytes 2628143         dropped pkts 14           in FECN pkts 0
in BECN pkts 0            out FECN pkts 0           out BECN pkts 0
in DE pkts 523            out DE pkts 0
out bcast pkts 2626       out bcast bytes 530028
pvc create time 1d02h, last time pvc status changed 03:16:03
```

Администратор волен настраивать нужное ему количество подинтерфейсов, несмотря на то что их поддержка в сети Frame Relay обеспечиваться не будет. В этом случае статус соединения при выполнении команды show frame-relay pvc явится указателем на это. В следующем примере активным является интерфейс Serial1.1 (PVC STATUS = ACTIVE) и по нему маршрутизатор может отправлять и получать информацию. Другой подинтерфейс — Serial1.3 — является неактивным (PVC STATUS = DELETED), он не поддерживается провайдером услуг, однако маршрутизатор продолжает его поддерживать и отслеживать его состояние. Например, если провайдер изменит конфигурацию и включит соответствующее постоянное виртуальное соединение с номером DLCI = 21, то состояние подинтерфейса Serial1.3 автоматически изменится. Такая модель работы может оказаться полезной в ситуации, когда распределенная сеть состоит из множества объектов и администратор сети решил модернизировать логическую топологию связи. Следует обратить внимание, что если используются протоколы маршрутизации, они автоматически начнут информировать соседние маршрутизаторы о доступных маршрутах, как только виртуальное соединение поменяет свой статус.

```
Router#show frame-relay pvc
PVC Statistics for interface Serial1 (Frame Relay DTE)

DLCI = 18, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial1.1

input pkts 780679        output pkts 669218    in bytes 105683698
out bytes 418481876      dropped pkts 134      in FECN pkts 0
in BECN pkts 0           out FECN pkts 0       out BECN pkts 0
in DE pkts 12344         out DE pkts 0
out bcast pkts 95801     out bcast bytes 15359940
pvc create time 5w5d, last time pvc status changed 00:39:48

DLCI = 21, DLCI USAGE = LOCAL, PVC STATUS = DELETED, INTERFACE = Serial1.3

input pkts 0          output pkts 8         in bytes 0
out bytes 1624        dropped pkts 0        in FECN pkts 0
in BECN pkts 0        out FECN pkts 0       out BECN pkts 0
in DE pkts 0          out DE pkts 0
out bcast pkts 8      out bcast bytes 1624
pvc create time 5w5d, last time pvc status changed 02:49:52
```

Управление трафиком в сети Frame Relay

Начиная с версии 11.2 операционной системы Cisco IOS администратору сети доступен очень мощный механизм управления трафиком (traffic shaping — обратите внимание, что иногда в технической литературе может встретиться перевод «формирование трафика», что в сущности то же самое). Как можно понять из названия, основная цель этого механизма заключается в предоставлении способов контролировать скорость, с которой маршрутизатор обрабатывает поток

трафика на своих интерфейсах. Данный механизм может работать как на физи
ческих (например, интерфейс Serial0, к которому подключен модем серии xDSL
так и на логических интерфейсах (например подинтерфейсы Frame Relay) марш
рутизатора, обеспечивая большую гибкость в применении. Последний факт очен
важен, так как именно он в определенной мере способствует внедрению техно
логии передачи голосового и другого чувствительного к задержкам трафика по
сети Frame Relay. Разработчики добавили удобный механизм работы с техноло
гией управления трафиком — она либо используется на выбранном по опреде
ленным критериям трафике, либо обслуживает весь проходящий через маршру
тизатор трафик, определяемый с помощью списков доступа (глава 9).

Вообще говоря, операционная система Cisco IOS предоставляет администра
тору два механизма регулирования потока трафика. Первый ориентирован н
ограничение скорости передачи (rate-limiting) и реализован в технологии CA
(Committed Access Rate), а второй — уже упомянутый в предыдущем абзаце ме
ханизм управления трафиком. Важно отметить, что в самой системе управлени
трафиком выделяют два компонента: так называемый Generic Traffic Shapin
(GTS — базовая система управления трафиком) и Frame Relay Traffic Shapin
(FRTS — управление трафиком в сетях Frame Relay).

Принципы работы указанных двух механизмов отличаются друг от друга
Например, в случае реализации технологии CAR на маршрутизаторе послед
ний будет либо отбрасывать пакеты, либо изменять поле «Тип сервиса» в заго
ловке дейтаграмм. В противоположность этому, технологии управления тра
фиком обычно задерживают трафик, привлекая буферную память или систем
очередей для хранения пакетов и «сглаживания» потока трафика в том случае
если скорость поступления данных больше, чем ожидается. Например, техноло
гия GTS использует взвешенную справедливую очередь (WFQ, см. главу 6 дл
получения детальной информации) для хранения пакетов, а FRTS может опе
рировать очередью с назначением приоритетов, настраиваемой очередью ил
очередью FIFO в зависимости от выполненных настроек. Далее в книге буде
рассматриваться именно механизм управления трафиком. Следует отметить, чт
обе технологии могут работать сообща и в основе их взаимодействия лежит теори
«ведра с меткой» (не совсем точный перевод английского термина token bucke
но как будет видно далее, довольно точно отражающий суть).

Как уже отмечалось, администратор сети может воспользоваться двумя ме
ханизмами управления трафиком — GTS и FRTS. Их базовая реализация имее
много общего, хотя команды для настройки отличаются (как и используемые схе
мы очередей). Кроме того, FRTS поддерживает формирование трафика на уров
не DLCI, в то время как GTS настраивается на интерфейсе или подинтерфейсе
Следует отметить, что администратор может настроить GTS для выполнения т
же функций, что и FRTS, при выделении одного DLCI на подинтерфейс (чащ
всего так и делается) и задействуя технологию извещения о перегрузках BEC
При настройке технологии GTS администратору также доступны списки доступ
для выбора управляемого трафика. GTS работает с большим количеством техноло
гий канального уровня, включая Frame Relay, ATM, Ethernet и SMDS (Switch
Multimegabit Data Service — служба коммутируемой мультимегабитовой пе
дачи данных). При использовании Frame Relay технология GTS может дина

чески адаптироваться к доступной пропускной способности путем отслеживания уведомлений BECN или же просто ограничивать трафик заранее определенной скоростью. Технология FRTS, в отличие от GTS, ориентирована на работу с технологией Frame Relay и призвана повысить качество обслуживания именно в таких сетях.

При рассмотрении общей теории работы «ведра с меткой» выделяют два параметра. Параметр R определяет скорость поступления информации (или скорость потока) в течение относительно длительного периода времени; параметр B определяет размер «ведра», или, иными словами, величину, на которую поток данных способен превысить значение R за короткий промежуток времени. Более точное условие следующее: в течение любого периода T количество посылаемых данных не может превысить $(RT + B)$. Применительно к трафику протокола IP работу «ведра с меткой» можно рассматривать следующим образом. Само «ведро» представляет собой счетчик, указывающий на разрешенное количество байтов данных, которые могут посылаться в любое время. При этом «ведро» заполняется со скоростью R (то есть счетчик увеличивается R раз в секунду) до достижения края «ведра» (иными словами, до максимального значения счетчика). Дейтаграммы протокола IP поступают и помещаются в очередь для обработки, и они обрабатываются маршрутизатором, если есть свободное место в «ведре». В этом случае дейтаграмма маршрутизируется и «ведро» опорожняется. В том случае, если места в «ведре» нет, дальнейшая судьба дейтаграммы зависит от конкретной реализации рассматриваемой технологии на маршрутизаторе. Следует обратить внимание, что в течение длительного времени при наличии задержек в потоке данных наполнение «ведра» может освобождаться, так что оно в состоянии принять дополнительные B байтов свыше оговоренного количества. Следовательно, B является, скажем так, показателем максимального «взрывного» объема трафика, который система сумеет принять.

На маршрутизаторах Cisco Systems принцип «ведра с меткой» определяется такими величинами, как средняя битовая скорость (average bit rate, также называется CIR), взрывной объем трафика (traffic burst size — Bc) и временной интервал (time interval — T), связанными между собой выражением:

$$CIR = Bc / T$$

Это означает, что для любого временного интервала исходящий поток трафика не должен превышать среднюю скорость. По умолчанию объем взрывного трафика устанавливается равным битовой скорости, деленной на 8, что приводит к временному интервалу 1/8 с или 125 мс. Учитывая, что средняя битовая скорость определяется за определенный интервал времени и она не постоянна в целом, реальная скорость передачи будет меняться значительно, пока общий объем передаваемых за этот интервал данных не больше, чем взрывной объем трафика.

Покажем это на примере. Предположим, что администратор сети настраивает управление трафиком для ограничения исходящего трафика по виртуальному соединению Frame Relay до 64 Кбит/с. Скорость подключения к сети Frame Relay составляет 256 Кбит/с. Так как исходящий трафик может передаваться на скорости подключения (то есть 256 Кбит/с), технология управления трафиком

должна понижать эту скорость каждый раз по истечении временного интервала. Предположим, что в рассматриваемом примере взрывной объем трафика устанавливается в 8000 бит (64 Кбит/с / 8 с), а временной интервал равен 125 мс. Это означает, что каждые 125 мс интерфейс будет передавать 8000 бит данных через виртуальное соединение. Однако так как интерфейс работает на скорости 256 Кбит/с, то потребуется 31,25 мс для передачи выделенных 8000 бит. Обобщая сказанное, получаем, что маршрутизатор будет каждые 125 мс передавать по виртуальному соединению 8000 бит за 31,25 мс, а затем он будет ожидать оставшиеся 93,75 мс до передачи следующих 8000 битов. Важно отметить, что маршрутизатор будет завершать передачу кадров, не обращая внимания на то, что количество оставшихся битов в кадре превысит настроенное значение взрывного объема трафика. Иными словами, если последний равен 8000 бит и маршрутизатор уже успел передать 7500 бит в течение заданного временного интервала, а следующий кадр имеет размер 1500 байт, то маршрутизатор будет передавать этой кадр длиной 1500 байт не дожидаясь следующего временного интервала.

Для настройки технологии GTS администратор может воспользоваться командой **traffic-shape rate bit-rate [burst-size [excess-burst-size]]**, выполняемой в режиме конфигурирования интерфейсов. Параметр **bit-rate** определяет желаемую скорость передачи исходящего трафика и указывается в битах в секунду (то есть если нужно ограничить скорость значением 64 Кбит/с, то следует вводить число 64 000). Следующий, необязательный параметр **burst-size** определяет количество битов, разрешенных к передаче за заданный интервал времени. В том случае, если данный параметр не определен, маршрутизатор получает это значение путем деления параметра **bit-rate** на 8. И последний необязательный параметр **excess-burst-size** определяет максимальное количество битов, которые могут превышать Bc в первом интервале в случае появления перегрузки в сети. Если технология управления трафиком настраивается на подинтерфесах Frame Relay, то битовая скорость (**bit-rate**) должна быть равна CIR, количество битов, разрешенных к передаче (**burst-size**), приравнивается Bc, а максимальное количество битов (**excess-burst-size**) равно Be для настраиваемого виртуального соединения.

Технология управления трафиком в сети Frame Relay (то есть FRTS) находит применение в нескольких случаях. Например, если существующая сетевая топология имеет высокоскоростной доступ в сеть в центральном офисе и несколько низкоскоростных соединений на удаленных объектах. Также технологию управления трафиком можно использовать тогда, когда существуют несколько виртуальных соединений на одном физическом интерфейсе. Еще одной причиной регулирования трафика может стать периодическое нахождение сети Frame Relay в состоянии перегрузки. При этом существует постоянная необходимость передачи трафика различных приложений с гарантией того, что разные типы трафика получат определенную долю полосы пропускания.

В первом случае из-за разницы в скоростях не исключено возникновение «узких мест» на виртуальных соединениях между удаленными объектами и центральным офисом. Технология управления трафиком дает возможность придительно ограничить скорость отправки данных по виртуальным соединениям из центрального офиса.

Если имеют место периодические перегрузки сети, можно дать команду маршрутизатору приостановить трафик до его отправки в сеть, что позволит сгладить проблему потери кадров. В этом случае маршрутизатор отслеживает получение сообщений BECN из сети и на основе принятой информации формирует отправляемый трафик. Собственно говоря, сам процесс формирования трафика заключается в том, что маршрутизатор помещает пакеты, принадлежащие этому трафику, в очередь, снижая тем самым скорость их отправки в сеть. Но заодно возрастает задержка, что бывает критичным для определенного типа трафика.

Довольно часто встречаются ситуации, когда требуется по одному виртуальному соединению передавать трафик различных приложений или протоколов. Тогда администратор может извлечь из своего арсенала настраиваемые очереди (custom queuing). Причем до версии 11.2 операционной системы Cisco IOS настраиваемые очереди поддерживались только на уровне интерфейса, а с выходом новой версии они могут быть определены на уровне виртуального соединения (то есть подинтерфейса).

Для настройки технологии FRTS необходимо выполнить несколько шагов. На первом шаге администратору следует сформировать карту класса (map class) с помощью команды map-class frame-relay, которой в качестве параметра необходимо указать имя класса:

```
Router#conf term
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#map-class frame-relay class1
Router(config-map-)#
```

О самом классе можно говорить как о некоем наборе параметров, определяющих поведение технологии FRTS. Как видно из примера, маршрутизатор переходит в режим настройки класса, что отображается в приглашении. Далее определяем сам класс. А именно: пиковую и среднюю скорость (в битах в секунду), которые разрешены для виртуального соединения, связанного с этим классом; указываем на то, что маршрутизатор должен динамически менять скорость отправки в зависимости от полученных сообщений BECN; уточняем тип очереди (либо настраиваемая, либо с заданным приоритетом) применительно к виртуальному соединению. Параметры, вводимые при настройке класса, будут рассмотрены отдельно.

После того как определен класс с соответствующими параметрами, следует включить поддержку технологии Frame Relay на интерфейсе с помощью команды encapsulation frame-relay. Далее на очереди активация механизма управления трафиком на интерфейсе с помощью команды frame-relay traffic-shaping. Ее выполнение включает этот механизм на всех постоянных и коммутируемых виртуальных соединениях, настроенных на данном интерфейсе:

```
Router#conf term
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial 1
Router(config-if)#frame-relay traffic-shaping
```

И на последнем этапе связываем настроенный класс с виртуальным соединением на данном интерфейсе (точнее, подинтерфейсе). Выполнение этой про-

цедуры производится командой frame-relay class с указанием в качестве параметра имени класса, которое было назначено на первом шаге. Следует отметить, что класс может быть связан как с интерфейсом, так и с выбранными подинтерфейсами отдельно:

```
Router(config)#int s1.1
Router(config-subi)#frame-relay class class1
```

Существуют несколько подходов к определению класса для управления трафиком. Первый (так называемый Rate Enforcement) заключается в задании средней и пиковой скорости передачи информации, если данные посылаются быстрее, чем получатель может их принять. Для настройки этих скоростей предназначена команда frame-relay traffic-rate, ожидающая в параметрах значения средней и пиковой скоростей. Скорости указываются в битах в секунду. Средняя скорость эквивалентна согласованному с провайдером услуг значению CIR, а пиковая скорость равна сумме:

$$CIR + Be/T = CIR + (1 + Be/Bc) = CIR + EIR$$

Рассмотрим пример настройки этого метода управления трафиком. Предположим, что существует распределенная сеть, построенная на базе Frame Relay, к которой подключен центральный офис на скорости 1,5 Мбит/с, а два удаленных офиса используют низкоскоростное подключение (например 9,6 Кбит/с). Администратор сети заказал создание постоянных виртуальных соединений между удаленными и центральным офисами со значением CIR = 9,6 Кбит/с. В организованной сети данные из центрального офиса могут посылаться на максимальной скорости, равной 1,5 Мбит/с, несмотря на то что гарантируемая скорость на виртуальных соединениях значительно ниже. Если сеть Frame Relay справится с передачей данных на столь высоких скоростях, то при их поступлении на удаленные маршрутизаторы может возникнуть состояние перегрузки, так как скорость поступления данных выше, чем скорость, на которой эти данные могут быть приняты. Естественно, одним из решений предполагается снижение скорости подключения центрального офиса, однако при использовании технологии управления трафиком администратор вправе определить скорость отправки маршрутизатором данных индивидуально для каждого виртуального соединения.

Шаги настройки маршрутизатора в центральном офисе следующие.

1. Сначала следует определить класс и перейти в режим его настройки:

   ```
   Router(config)#map-class frame-relay <имя-класса>
   ```

2. Далее нужно задать скорости передачи информации:

   ```
   Router(config-map-class)#frame-relay traffic-rate <средняя пиковая скорость>
   ```

3. Средняя скорость передачи эквивалентна значению CIR, а пиковая, как уже отмечалось, вычисляется по формуле: CIR + Bc/T = CIR * (1 + Be/Bc). По умолчанию пиковая скорость равна скорости подключения к сети Frame Relay и это значение задается как аргумент команды bandwidth. Для рассматриваемого примера средняя скорость может быть установлена равной 9600 бит/с, а пиковая — 18 000 бит/с.

4. Следующий шаг — включение поддержки управления трафиком для всех виртуальных соединений на выбранном интерфейсе:

```
Router(config-if)#frame-relay traffic-shaping
```

5. И последним действием нужно выполнить связывание настроенного класса с интерфейсом или подинтерфейсом:

```
Router(config-if)#frame-relay class <имя-класса>
```

Следует обратить внимание на то, что все виртуальные соединения, созданные на интерфейсе, будут наследовать все параметры, определенные в классе.

Помимо явного задания скорости передачи администратор может настроить более гибкий механизм, позволяющий маршрутизаторам автоматически реагировать на возникновение перегрузки в сети (метод Dynamic Enforcement).

Маршрутизаторы Cisco допускают два режима: либо они игнорируют сообщения BECN/FECN, либо изменяют поток трафика при их получении. Для включения поддержки этих сообщений необходимо воспользоваться командой frame-relay adaptive-shaping, при настройке классов. Если маршрутизатор получает сообщение BECN в течение определенного промежутка времени, он уменьшает скорость передачи информации на 25%. Причем уменьшение скорости будет продолжаться с каждым полученным сообщением BECN, до тех пор пока скорость трафика не достигнет наименьшего приемлемого значения (так называемое значение MINCIR).

После того как скорость передачи снижена, маршрутизатор будет выжидать 16 временных интервалов, в течение которых, если при этом не получит ни одного сообщения BECN, он станет постепенно увеличивать скорость передачи информации. Приращение скорости будет производиться на величину (Be + Bc) /16 или, что более точно, на величину Byte Limit/16. Значение Byte Limit можно узнать, выполняя команду show traffic-shape в глобальном режиме:

```
Router#sh traffic-shape
      Access  Target   Byte   Sustain   Excess          Interval  Increment Adapt
I/F   List    Rate     Limit  bits/int  bits/int (ms)   (bytes)   Active
Se1.1 56000   875      7000   0         125             875       BECN
Se1.2 56000   7875     56000  56000     125             875       BECN
Se1.3 56000   7875     56000  56000     125             875       BECN
```

Отсюда ясно, что маршрутизатору потребуется гораздо больше времени для достижения значения скорости, равной CIR, чем для ее снижения до величины MINCIR (алгоритм очень напоминает алгоритм технологии медленного старта для протокола TCP). Одним из способов сокращения этого временного интервала будет установка значения Be в семикратное значение Bc. Такая установка гарантирует немедленное достижение скорости передачи, равной CIR, при отсутствии сообщений BECN в течение 16-кратного временного интервала.

Рассмотрим поэтапно пример использования этого механизма. Предположим, что выполняется подключение к сети Frame Relay с физической скоростью 64 Кбит/с. При этом заключено соглашение с провайдером, согласно которому последний обеспечивает гарантированную скорость передачи CIR, равную 16 Кбит/с.

Но при таком подключении маршрутизатор в случае отсутствия перегрузки в сети Frame Relay способен передавать данные с максимальной скоростью 64 Кбит/с.

На первом этапе настраиваем интерфейс маршрутизатора (в примере — Serial0), который физически подключен к сети Frame Relay. Для этого интерфейса указываем тип инкапсуляции и сообщаем маршрутизатору при помощи команды frame-relay traffic-shaping о необходимости использования технологии управления трафиком:

```
!
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay traffic-shaping
 frame-relay lmi-type ansi
!
```

На втором этапе выполняем настройку класса с именем class2 с указанием скорости, с которой маршрутизатор должен посылать данные при отсутствии перегрузки в сети. Это мы осуществим командой frame-relay cir 64000, где число 64 000 обозначает скорость в битах в секунду.

Далее указываем с помощью команды frame-relay bc 8000 количество байтов (8000), которые должны быть отосланы в течение интервала T. Рекомендуется устанавливать это значение равным 1/8 значения CIR. Напомним, что значение Bc определяет количество битов данных, посылаемых за каждый интервал T.

Затем с помощью команды frame-relay be 0 задаем максимальное количество данных, посылаемое в первом интервале. В том случае, если маршрутизатор уже посылает данные на скорости подключения, это значение должно быть равно 0.

Последним действием будет выполнение команды frame-relay mincir 16000, которая устанавливает значение MINCIR, то есть значение, до которого маршрутизатор будет снижать скорость передачи в случае обнаружения перегрузки в сети. В нашем примере это значение соответствует значению CIR, которое согласовано с провайдером услуг Frame Relay и равно 16 Кбит/с.

В технологии управления трафиком Frame Relay временной интервал определяется значением T, равным отношению (Bc/CIR). Длительность этого временного интервала для эффективной работы технологии управления трафиком не должна быть больше 1/8 с. В нашем примере T = 8000 / 64 000 = 1/8 с. Таким образом, маршрутизатор будет передавать следующие порции данных каждую 1/8 с: 8000 (Bc+Be), 8000 (Bc), 8000, 8000, 8000, 8000, 8000, 8000, что и соответствует значению скорости в 64 000 бит/с. Если бы скорость доступа была равна 128 Кбит/с, а Be = 64 000, то маршрутизатор передавал бы следующие порции: 72 000, 8000, 8000, 8000 ...

Обобщая все вместе, можно конкретизировать этапы настройки механизма управления трафиком.

1. Создаем именованный класс (имя класса class2):

```
Router(config)#map-class frame-relay class2
Router(config-ma)#frame-relay cir 64000
Router(config-ma)#frame-relay bc 8000
Router(config-ma)#frame-relay be 0
```

```
Router(config-ma)#frame-relay mincir 16000
Router(config-ma)#frame-relay adaptive-shaping becn
...
```

2. Применяем созданный класс на выбранном подинтерфейсе (в нашем примере Serial0.1):

```
Router#conf term
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int s0.1
Router(config-su)#frame-relay class class2
```

3. Убеждаемся в правильности произведенных настроек с помощью команды show frame-relay pvc:

```
Router#sh fr pvc
PVC Statistics for interface Serial0 (Frame Relay DTE)
DLCI = 19, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

    input pkts 24767          output pkts 24681          in bytes 5100026
    out bytes 4720420         dropped pkts 4             in FECN pkts 0
    in BECN pkts 0            out FECN pkts 0            out BECN pkts 0
    in DE pkts 374            out DE pkts 0
    out bcast pkts 7385       out bcast bytes 989410
    Shaping adapts to BECN
    pvc create time 1d15h, last time pvc status changed 08:48:32
```

Как видно, в процессе выполнения данной команды администратор сети получает значения множества параметров. Таблица 8.1 содержит их описание.

Таблица 8.1. Описание результатов команды show frame-relay pvc

Поле	Описание
DLCI	Идентификатор постоянного виртуального соединения
DLCI USAGE	Значение SWITCHED, если маршрутизатор работает как коммутатор Frame Relay, или LOCAL, если маршрутизатор выступает в роли DTE
PVC STATUS	Статус PVC: ACTIVE, INACTIVE или DELETED
INTERFACE = Serial0.1	Подинтерфейс, ассоциированный с этим номером DLCI
Input pkts	Количество полученных пакетов по PVC
Output pkts	Количество переданных пакетов по PVC
In bytes	Количество полученных байтов
Out bytes	Количество переданных байтов
dropped pkts	Количество пакетов, которые маршрутизатор отбросил на уровне Frame Relay
in FECN pkts	Количество полученных пакетов с установленным битом FECN
in BECN pkts	Количество полученных пакетов с установленным битом BECN
out FECN pkts	Количество отправленных пакетов с установленным битом FECN

продолжение

Таблица 8.1 (продолжение)

Поле	Описание
out BECN pkts	Количество отправленных пакетов с установленным битом BECN
in DE pkts	Количество полученных DE-пакетов (помечены для удаления)
out DE pkts	Количество отправленных DE-пакетов
outbcast pkts	Количество отправленных широковещательных пакетов
outbcast bytes	Количество байтов в оправленных широковещательных пакетах
pvc create time	Время создания данного PVC
last time pvc status changed	Время изменения статуса PVC

Кроме того, администратор сети может просмотреть настроенные параметры управления трафиком, выполнив команду show traffic-shape:

```
Router#show traffic-shape
      Access Target Byte  Sustain  Excess          Interval  Increment Adapt
I/F   List  Rate  Limit bits/int bits/int (ms)   (bytes)   Active
Se0.1 64000 1000  8000  0        125             1000      BECN
```

Таблица 8.2 содержит описание значений параметров, выдаваемых в результате выполнения команды show traffic-shape.

Таблица 8.2. Описание результатов выполнения команды show traffic-shape

Поле	Описание
Target Rate	Скорость, на которой происходит управление трафиком (бит/с)
Byte Limit	Максимальное количество байтов, передаваемых за интервал
Sustain bits/int	Настроенное значение количества битов, передаваемых за интервал
Excess bits/int	Настроенное количество дополнительных битов в первом интервале
Interval (ms)	Интервал для внутреннего использования. Он может быть меньше, чем Bc/CIR, если маршрутизатор определяет, что поток трафика будет более стабилен с меньшим интервалом
Increment (bytes)	Количество байтов, которые поддерживаются на каждом интервале
Adapt Active	Содержит BECN, если была включена поддержка BECN adaptation

Рассмотрим более детально результат команды show traffic-shape для следующих данных:

Target Rate = CIR = 64 000 бит/с
Mincir = CIR/2 = 64 000 / 2 = 32 000 бит/с
Sustain = Bc = 8000 бит/интервал
Excess = Be = 8000 бит/интервал
Interval (T) = Bc/CIR = 8000 / 64 000 = 125 мс
Increment = Bc/8 = 8000 / 8 = 1000 байт
Byte Limit = Increment + Bc/8 = 1000 + 8000 / 8 = 2000 байт

При настройке этих параметров для управления трафиком в технологии Frame Relay результаты, выдаваемые командой show traffic-shape, можно представить в виде последовательности (рис. 8.3).

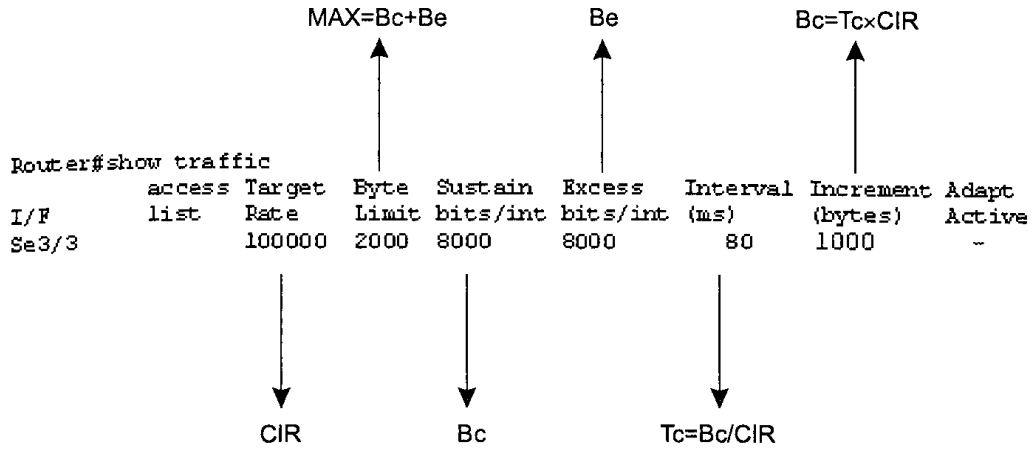


Рис. 8.3. Пример результата выполнения команды show traffic-shape

Наконец, в арсенале администратора имеется третий метод определения класса для управления трафиком — метод с использованием очередей (настраиваемых или с заданным приоритетом). В этом случае маршрутизатор будет руководствоваться настроенной стратегией очередей для формирования стабильного потока трафика. Для настройки этого метода администратору сети следует воспользоваться командой frame-relay custom-queue-list (в случае настраиваемой очереди) с номером очереди в качестве параметра. Соответственно, если было принято решение в пользу очереди с заданным приоритетом, то синтаксис команды меняется: frame-relay priority-group <номер> (детальное описание команд для каждого типа очереди в главе 6).

9 Настройка базовой безопасности

Все рассмотренные ранее настройки различных технологий, все усилия сетевого администратора могут быть сведены на нет при попытке несанкционированного проникновения в вашу сеть. В связи с этим функции обеспечения безопасности, встроенные в Cisco IOS, являются, несомненно, важнейшим средством, направленным на создание сети с эффективной защитой.

Списки управления доступом Cisco IOS

Списки доступа (access control list — ACL) являются основным инструментом операционной системы Cisco IOS по обеспечению безопасности, и их синтаксис используется для реализации множества функциональных возможностей маршрутизатора. Из названия нетрудно понять, что основное предназначение списков управления доступом заключается в создании фильтра безопасности, направленного на блокирование трафика, входящего в сеть или покидающего часть сети. Однако по мере развития операционной системы IOS компания Cisco Systems расширила синтаксис команд формирования списков доступа для решения и других задач. К ним относятся:

- формирование фильтров маршрутов;
- классификация пакетов приложений для предоставления качества обслуживания и для реализации стратегии очередей;
- задачи шифрования и маршрутизации по запросу (dial-on-demand routing).

Такой широкий спектр задач и определяет всю важность понимания работы списков контроля доступа и синтаксиса их команд.

Наиболее широко списки доступа используются при фильтрации трафика на интерфейсах маршрутизатора. Работая как фильтр, список доступа определяет трафик, который запрещен или, наоборот, разрешен к прохождению через интерфейс. После составления списка доступа, отвечающего потребностям фильтрации, администратор сети применяет его на выбранном интерфейсе (или на множестве интерфейсов). В результате маршрутизатор проверяет следующий через интерфейс трафик и отбрасывает запрещенные к передаче пакеты. Важно отметить, что фильтрация пакетов осуществляется либо во входящем (inbound) направлении, либо на исходящем (outbound). Для фильтрации в обоих направлениях необходимо применение двух списков доступа.

Рисунок 9.1 иллюстрирует простое применение списка доступа, который проверяет поступающий трафик на интерфейсе S0 маршрутизатора и отбрасывает трафик, приходящий из сети компании Б.

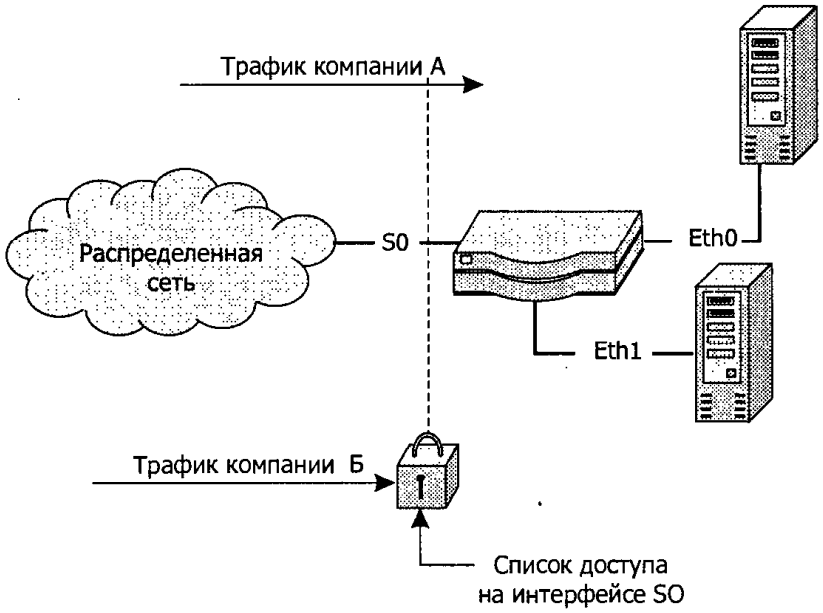


Рис. 9.1. Проверка трафика по спискам доступа

Трафик из компании А проходит через интерфейс и может достичь двух сегментов сети, подключенных к маршрутизатору через порты Ethernet0 и Ethernet 1. Если же на интерфейс маршрутизатора поступает трафик от других компаний, то дальнейшая его судьба целиком будет зависеть только от деталей составления списка доступа. Например, установленные правила могут разрешать прохождение только трафика от компании А, а остальной трафик (включая компанию Б) запрещать. Другим примером могут быть соглашения, разрешающие любой трафик, за исключением трафика от компании Б. Синтаксис команд, формирующий правила управления доступом, достаточно гибок и позволяет создавать правила поведения маршрутизатора любой сложности. Отметим, что в примере сознательно для наглядности используются названия компаний (хотя и весьма условные), в реальности же администратор работает с такими параметрами, как номера портов протоколов, адреса сетей, подсетей и т. п. При хорошо разработанной схеме документирования создаваемых списков доступа работа с этими параметрами не вызывает больших проблем.

Рассмотренный пример показывает действие входящего списка доступа (inbound access list), анализирующего трафик, поступающий на интерфейс Serial0. Администратор может настроить также и исходящий список доступа (outbound access list) для трафика, покидающего определенный интерфейс (рис. 9.2).

В рассматриваемом примере исходящий список доступа применяется на интерфейсе Ethernet1 (Eth1) маршрутизатора. Правила таковы, что только трафику компании Б запрещено проходить в подсеть 2, в то время как подсеть 1 имеет полный доступ через маршрутизатор, так как нет списков доступа, блокирующих доступ к ней на интерфейсах Ethernet0 или Serial0 (S0).

Рис. 9.2. Настройка списков доступа для подсети 2

Таким образом, каждый список доступа состоит из набора правил (rules), инструкций, определяющих политику доступа. Каждое правило описывает шаблон для совпадения (это может быть адрес протокола IP, тип протокола и т. п.) и поведение маршрутизатора (пропустить пакет или отбросить). Важно отметить, что правила в списке доступа обрабатываются маршрутизатором последовательно. Когда маршрутизатор принимает пакет, соответствующий инструкции, он выполняет предопределенное действие и останавливает обработку списка доступа для данного пакета.

Списки доступа являются очень гибкой технологией — администратор может настраивать их с высокой степенью детализации или, наоборот, оставлять очень широкими, всеохватывающими, в зависимости от потребностей. Например, можно создать список доступа, запрещающий прохождение через маршрутизатор сеанса Telnet с определенного устройства в сети.

При создании списка доступа ему присваивается номер или имя. Имя назначается в случае использования именованных списков. Присвоенный номер позволяет ссылаться на список доступа как на единый объект в операционной системе IOS, что повышает удобство его применения на интерфейсах. Например, можно однажды создать список доступа, а затем применять его на необходимом количестве интерфейсов. Следует отметить, что существуют ограничения в нумерации списков доступа. Диапазон допустимых номеров зависит от типа фильтруемого протокола. Например, для протокола IP существуют два типа списков доступа, каждый со своим числовым диапазоном:

- стандартный список доступа IP (standard IP access list) — можно назначать номера от 1 до 99;
- расширенный список доступа IP (extended IP access list) — разрешенные номера от 100 до 199.

Другие диапазоны номеров списков доступа зарезервированы для фильтрации трафика таких протоколов, как IPX, AppleTalk, DECNet и т. д. Таблица 9.1 представляет диапазоны нумерации, предназначенные для фильтрации трафиков различных протоколов.

Таблица 9.1. Диапазоны номеров списков доступа для протоколов

Протокол	Диапазон номеров
IP	1–99 и 1300–1999
Extended IP	100–199 и 2000–2699
Ethernet type code	200–299
Ethernet address	700–799
Transparent bridging (тип протокола)	200–299
Transparent bridging (код производителя)	700–799
Extended transparent bridging	1100–1199
DECNet and extended DECNet	300–399
XNS	400–499
Extended XNS	500–599
AppleTalk	600–699
Source-route bridging (тип протокола)	200–299
Source-route bridging (код производителя)	700–799
IPX	800–899
Extended IPX	900–999
IPX SAP	1000–1099
Standard VINES	1–100
Extended VINES	101–200
Simple VINES	201–300

Ввиду того что в книге рассматривается только один сетевой протокол из перечисленных — IP, акцент будет сделан именно на списках доступа для протокола IP. Стандартный список доступа фильтрует трафик, основываясь только на адресах отправителя. При этом каждое правило в списке доступа содержит некий битовый шаблон (bit pattern), аналогичный записи IP-адреса в точечно-десятичной нотации. Этот шаблон маршрутизатор сравнивает с адресами отправителей поступающих пакетов. Если адрес отправителя совпадает с настроенным битовым шаблоном, маршрутизатор выполняет предопределенное действие — отбрасывает либо пропускает пакет. Администратор сети может формировать правила, которые работают не только с одним устройством в сети, а охватывают подсети или сети целиком.

Для создания стандартного списка доступа сначала следует выбрать номер в интервале от 1 до 99, который будет идентифицировать список доступа на маршрутизаторе. При этом абсолютно неважно, какое именно число будет выбрано,

хотя для простоты последующего управления большим количеством списк доступа рекомендуется нумерацию осуществлять с единицы.

Создание списка доступа начинается с объявления правил (одного или н скольких), которые добавляются последовательно. При этом администратор до жен будет использовать команду access-list, выполняемую в глобальном режи работы (табл. 9.2).

Таблица 9.2. Синтаксис команды access-list

Цель	Команда
Определить стандартный список доступа, используя адрес отправителя и маску	access-list <номер списка доступа> {deny | permi <отправитель> [маска] [log]
Определить стандартный список доступа, используя сокращение для отправителя и маску, равную 0.0.0.0 255.255.255.255	access-list <номер списка доступа> {deny | perm any [log]

Для того списка доступа, который разрешает прохождение трафика (ключе вое слово permit) с устройства в сети с адресом 192.168.3.2, можно выполни следующие команды:

```
Cisco805#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco805(config)#access-list 1 permit 192.168.3.2 0.0.0.0
```

В рассматриваемом примере число 0.0.0.0 является маской (mask — в ан лоязычной литературе может также встретиться название wildcard) для шабло на 192.168.3.2. Маска указывает маршрутизатору на те биты в шаблоне, которы следует сравнивать с адресом отправителя, и те, которые можно проигнорироват Как и маска в схеме адресации протокола IP, маска в списке доступа состоит и 32 бит, записанных в точечно-десятичной нотации. То есть маска 0.0.0.0 соответ ствует следующему двоичному представлению: $(00000000)_2(00000000)_2(00000000)$ $(00000000)_2$. Нулевые биты в маске списка доступа указывают маршрутизатор на необходимость сравнения этих битов — иными словами, нужно сопоставлять биты в адресе отправителя пакета с аналогичными битами в шаблоне. Соответ ственно, единичные биты указывают на то, что сравнение проводить не нужно Таким образом, маска 0.0.0.0 в сочетании с шаблоном 192.168.3.2 указывае маршрутизатору на то, что он должен проверять все 32 бит на соответствие, и в р зультате только адрес 192.168.3.2 будет считаться удовлетворяющим правил списка доступа.

Используя единичные биты в маске, администратор получает в свое распоряже ние очень гибкий инструмент, оперирующий не отдельными адресами, а их диапа зонами. В следующем примере список разрешает доступ с адреса 192.168.1.12 по адрес 192.168.1.159 включительно.

```
Router(config)#access-list 2 permit 192.168.1.128 0.0.0.31
```

Точечно-десятичная нотация не позволяет понять, как именно будет работат данное правило в списке доступа. Для более наглядного представления следуе преобразовать последний байт шаблона и соответствующей маски в двоично представление и проверить биты, установленные в единицу. В нашем пример

не требуется приведения первых трех байтов в двоичную запись, так как все биты в маске нулевые. Если представить последний байт маски в двоичном виде, то результат будет указывать маршрутизатору на то, что необходимо сравнивать 3 младших бита и игнорировать 5 старших.

Шаблон: 192.168.1.128 соответствует192.168.1.$(10000000)_2$.

Маска: 0.0.0.31 соответствует 0.0.0.$(00011111)_2$.

Накладывая друг на друга шаблон и маску, получаем следующее правило для сравнения: 192.168.1.$(100XXXXX)_2$, где символ X указывает на биты, которые не учитываются при сравнении. То есть 5 младших битов в адресе отправителя могут быть любой комбинацией нулей и единиц в пределах от 192.168.1.$(10000000)_2$ до 192.168.1.$(10011111)_2$, что как раз и соответствует диапазону адресов в десятичном представлении с 192.168.1.128 по 192.168.1.159.

Рассмотрим другой пример. Предположим, администратор сети настроил список доступа на маршрутизаторе, разрешающий прохождение пакетов с шаблоном 172.30.16.0 и маской 0.0.15.255.

```
Router(config)#access-list 3 permit 172.30.16.0 0.0.15.255
```

Предположим также, что устройство с адресом 172.30.20.32 пытается передавать пакеты через интерфейс маршрутизатора, на котором был применен этот список доступа. В этом случае маршрутизатор выполняет следующее сравнение (табл. 9.3).

Таблица 9.3. Схема № 1 сравнения маски и шаблона

Адрес устройства	172.30.20.32	**(10101100).(00011110).(0001**	0100).(00100000)
Шаблон	172.30.16.0	**(10101100).(00011110).(0001**	0000).(00000000)
Маска	0.0.15.255	**(00000000).(00000000).(0000**	1111).(11111111)
		Биты сравниваются	Биты игнорируются

В результате сравнения маршрутизатор разрешит передачу трафика, отправителем которого является устройство с адресом 172.30.20.32, так как нулевые биты маски у адреса устройства и у шаблона совпадают.

Теперь представим, что через интерфейс маршрутизатора с установленным списком доступа начинает передавать информацию устройство с адресом 172.30.32.8. Результаты сравнения, выполняемого маршрутизатором, показаны в табл. 9.4.

Таблица 9.4. Схема № 2 сравнения маски и шаблона

Адрес устройства	172.30.32.8	**(10101100).(00011110).(0010**	0100).(00001000)
Шаблон	172.30.16.0	**(10101100).(00011110).(0001**	0000).(00000000)
Маска	0.0.15.255	**(00000000).(00000000).(0000**	1111).(11111111)
		Биты сравниваются	Биты игнорируются

По результатам сравнения маршрутизатор будет отбрасывать пакеты от устройства, так как последние два сравниваемых бита в адресе устройства и в шаблоне не одинаковы.

СОВЕТ

Для простоты восприятия маску списка доступа можно рассматривать как реверсивную маску подсети для протокола IP. В то время как маска списка доступа начинается с серии нулей и заканчивается серией единиц (например, 0.0.255.255), маска подсети начинается с единичных битов, а заканчивается нулевыми (255.255.0.0).

Как уже не раз отмечалось, список управления доступом формируется на основе правил, и если хотя бы одно из них охватывает все потребности в фильтрации трафика (как в рассматриваемых примерах), то на этом формирование списка доступа можно считать завершенным. Однако довольно часто одним правилом процесс формирования списка доступа не ограничивается — могут потребоваться еще несколько дополнительных соглашений, пропускающих или отбрасывающих пакеты (например, для нескольких классов адресов). Для добавления правил к списку доступа администратор просто продолжает вводить команды, соответствующие правилам, с тем же самым номером списка доступа. Например, если продолжить рассмотрение примера списка доступа с номером 1 (access list 1), то можно добавить к нему новые правила.

```
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#access-list 1 permit 192.168.3.2 0.0.0.0
Router(config)#access-list 1 permit 192.168.3.101 0.0.0.0
```

После того как формирование списка доступа с номером 1 полностью завершено, можно выполнить команду show running-config для просмотра правил. Интересной особенностью является то, что администратор сети вправе создавать сколько угодно правил, не тревожась о времени простоя, так как маршрутизатор не будет фильтровать пакеты до тех пор, пока списки доступа не будут применены к выбранным интерфейсам.

```
Cisco805#show running-config
Building configuration...
...
access-list 1 permit 192.168.3.101
access-list 1 permit 192.168.3.2
...
```

Дополнительно существует более удобный способ проверки созданных списков доступа. Он реализуется командой show access-lists с указанием номера сформированного списка. Следует отметить, что операционная система опускает маску списка доступа при выводе на экран, если она была установлена в 0.0.0.0. Кроме того, администратор, создавая список доступа, работающий только с одним устройством, может также не вводить маску, состоящую из одних нулей.

```
Cisco805#show access-lists 1
Standard IP access list 1
      permit 192.168.3.101
      permit 192.168.3.2
```

Если необходимо создать правило, работающее только с конкретным адресом, то администратор вправе не вводить маску вообще, а включить в ком

ключевое слово host, которое указывает маршрутизатору, что следующий за этим словом параметр будет соответствовать адресу отправителя, с которым будет работать данное правило.

```
Cisco805(config)#access-list 1 permit host ?
   Hostname or A.B.C.D  Host address

Cisco805(config)#access-list 1 permit host i-mk
Translating "i-mk"...domain server (192.168.3.252) [OK]
```

Как видно из примера, можно вместо конкретного адреса ввести имя хоста. Маршрутизатор, на котором настроена служба разрешения имен, автоматически преобразует имя в соответствующий адрес.

Помимо правил с указанием определенных адресов или их диапазонов, во власти администратора так называемое «разрешающее все» правило. Для этого в качестве шаблона следует указать 0.0.0.0, а маску установить равной 255.255.255.255. Данное правило будет предписывать маршрутизатору игнорировать все 32 бита в адресах отправителей поступающих пакетов. Для упрощения ввода подобной команды операционная система IOS позволяет использовать ключевое слово any, которое не требует после себя никаких параметров в виде адресов.

```
Cisco805(config)#access-list 4 permit any
```

На практике чаще всего сначала настраиваются правила, ограничивающие доступ (deny), которые заканчиваются разрешающим правилом (permit). Иными словами, обычно формируются списки доступа, которые запрещают доступ определенным устройствам, но разрешают всем другим.

Как уже говорилось, до тех пор пока администратор не применил созданный список доступа на нужном интерфейсе, маршрутизатор не будет фильтровать трафик. Для применения созданного списка доступа на интерфейсе можно воспользоваться командой ip access-group, выполняемой в режиме конфигурирования интерфейса и имеющей формат:

ip access-group {<номер списка доступа | имя>} {in | out}

В следующем примере на интерфейсе Ethernet0 маршрутизатора команда ip access-group применяется к списку доступа с номером 4, для которого разрешено прохождение любого трафика (правило permit any).

```
Cisco805#conf term
Enter configuration commands. one per line. End with CNTL/Z.
Cisco805(config)#interface ethernet 0
Cisco805(config-if)#ip access-group 4 in
```

Как видим, сначала выполняется переход в режим настройки интерфейса, причем интерфейс выбирается исходя из того, что именно на нем будет затем проверяться список доступа. В качестве параметра команды ip access-group указывается одно из ключевых слов — in или out, указывающих на направление фильтрации трафика. В нашем случае маршрутизатор будет фильтровать согласно списку доступа входящий на интерфейс Ethernet0 трафик. Соответственно, с помощью ключа out можно вынудить маршрутизатор фильтровать покидающий интерфейс трафик.

После создания и применения списка доступа разумно проверить корректность результата. Для этого воспользуемся командой show ip interface с указанием имени и номера интерфейса, к которому был применен список доступа.

```
Cisco805#sh ip interface eth0
Ethernet0 is up, line protocol is up
  Internet address is 192.168.101.254/24
...
  Outgoing access list is not set
  Inbound  access list is 4
...
```

Есть несколько важных моментов, которые следует помнить при разработке списков доступа. Прежде всего это то, что правила в списке доступа обрабатываются последовательно, то есть маршрутизатор сравнивает пакеты согласно правилам от начала списка доступа, и если пакет не удовлетворяет условию, маршрутизатор продолжает сравнение с другими правилами, до тех пор пока не найдет подходящее. При этом первое единственное правило, которое оказалось действительно, и определит дальнейшую судьбу пакета — маршрутизатор либо пропустит его, либо отбросит.

После того как определенные в правиле действия с пакетом выполнены, маршрутизатор прекращает обработку списка доступа для данного пакета и не проверяет оставшуюся часть. Однако, когда поступает следующий пакет, весь процесс проверки на соответствие правилам начинается заново, с самого первого правила.

Очевидно, что при довольно интенсивном трафике маршрутизатор будет тратить значительную часть своих ресурсов на контроль соответствия пакетов спискам доступа. Отсюда следует рекомендация формировать правила в списке доступа таким образом, чтобы большая часть трафика требовала сравнения только с первыми несколькими правилами в списке доступа. В этом случае большинство пакетов будет быстро обработано, а меньшее их количество будет сравниваться со всеми объявленными правилами. Такой подход позволит значительно ускорить обработку списков доступа на маршрутизаторе.

Следует отметить, что при составлении списка доступа каждое правило вставляется в конец списка в порядке своего добавления. Администратор не может вставить правило произвольно между двумя другими или в начало существующего списка. Для этого потребуется удалить весь список доступа и создать его заново.

СОВЕТ

При написании списка доступа не вводите его сразу в маршрутизатор. Воспользуйтесь любым текстовым редактором и формируйте список сначала там. После того как все задачи, задуманные для данного списка доступа, определены, можно простым копированием текстовых строк через буфер обмена операционной системы перенести список в маршрутизатор. Это также упростит задачу добавления новых правил не в конец списка. Полученный текстовый файл имеет смысл сохранить для дальнейшего использования, причем лучше его хранить в безопасном месте, так как фактически его можно рассматривать как настройку защитного экрана. Для удаления списка доступа из конфигурации маршрутизатора достаточно в глобальном режиме выполнить команду no access-list с указанием номера списка.

Операционная система Cisco IOS не позволяет удалять правила, уже существующие в списке. Как и в ситуации со вставкой правил, для удаления конкретных правил из списка потребуется аннулировать список целиком и создать заново с нужными исправлениями. Также следует обратить внимание на то, что при создании правила, отбрасывающего определенные пакеты, нужно убедиться, что среди ранее созданных (находятся ближе к началу списка) нет такого, которое по ошибке разрешает прохождение пакетов, так как будет использоваться первое правило, совпадающее с пакетом.

Очень важной особенностью является то, что все пакеты будут совпадать по крайней с мере одним правилом в списке доступа, поскольку это последнее правило в списке существует по умолчанию и оно невидимо. Данное правило заставляет отбрасывать все пакеты; оно не отображается при выполнении команд show access-list и show running-config, но оно есть данность и отключить его невозможно. Последнее правило будет отбрасывать все пакеты-претенденты, которое прошли через цепочку правил и не нашли своего совпадения. Применительно к ранее показанному примеру (access list 1) фактический список доступа будет выглядеть следующим образом:

```
access-list 1 permit 192.168.3.2 0.0.0.0
access-list 1 permit 192.168.3.101 0.0.0.0
(access-list 1 deny any)
```

Здесь невидимое правило access-list 1 deny any (что подразумевают скобки) автоматически вставляется в конец каждого списка доступа, включая расширенные списки доступа, о которых будет сказано немного ниже. Это делается с целью поддержания дополнительных мер предосторожности, чтобы исключить бреши в системе безопасности, если при создании списка доступа были допущены ошибки.

ИЗ ОПЫТА

При формировании списка доступа не забывайте про служебный трафик, которым обмениваются маршрутизаторы (например, протокол RIP). Если подобный трафик не учесть при создании списка, то он будет блокирован невидимым правилом в конце списка.

Важно помнить, что списки доступа являются однонаправленными. Когда администратор применяет список доступа к интерфейсу, он должен указать направление передачи пакетов. В этом случае маршрутизатор будет сравнивать соответственно входящие или покидающие интерфейс пакеты. Когда же направление не указано, подразумевается исходящий трафик (out). Если необходимо фильтровать как входящий, так и исходящий потоки, следует создать и применить два отдельных списка доступа, или же использовать один список доступа, но для двух направлений.

СОВЕТ

Не делайте список слишком большим, так как при большом количестве правил производительность маршрутизатора может быть резко снижена — он будет вынужден сравнивать всякий поступающий пакет с каждым правилом в списке доступа. Естественно, что длина списка доступа зависит от производительности конкретной модели маршрутизатора и пропускной способности, которую нужно поддерживать на канале связи. Лучше находить разумный компромисс между безопасностью и скоростью работы.

Расширенные списки доступа позволяют администратору формировать более точные правила для фильтрации трафика, чем это позволяют делать стандартные списки. В то время как стандартные списки при фильтрации опираются только на адрес отправителя, расширенные списки доступа могут сравнивать: адреса отправителя и получателя, транспортные протоколы (ICMP, TCP, UDP, EIGRP, OSFP и т. п.), протоколы верхнего уровня (FTP, HTTP, SMTP и т. п.) и др. В следующем примере приведены протоколы, работающие поверх TCP, для которых при формировании расширенного списка доступа допускается использовать сокращения наименований.

```
Cisco1601(config)#access-list 101 permit tcp any any eq ?
  <0-65535>    Port number
  bgp          Border Gateway Protocol (179)
  chargen      Character generator (19)
  cmd          Remote commands (rcmd, 514)
  daytime      Daytime (13)
  discard      Discard (9)
  domain       Domain Name Service (53)
  echo         Echo (7)
  exec         Exec (rsh, 512)
  finger       Finger (79)
  ftp          File Transfer Protocol (21)
  ftp-data     FTP data connections (used infrequently, 20)
  gopher       Gopher (70)
  hostname     NIC hostname server (101)
  ident        Ident Protocol (113)
  irc          Internet Relay Chat (194)
  klogin       Kerberos login (543)
  kshell       Kerberos shell (544)
  login        Login (rlogin, 513)
  lpd          Printer service (515)
  nntp         Network News Transport Protocol (119)
  pim-auto-rp  PIM Auto-RP (496)
  pop2         Post Office Protocol v2 (109)
  pop3         Post Office Protocol v3 (110)
  smtp         Simple Mail Transport Protocol (25)
  sunrpc       Sun Remote Procedure Call (111)
  syslog       Syslog (514)
  tacacs       TAC Access Control System (49)
  talk         Talk (517)
  telnet       Telnet (23)
  time         Time (37)
  uucp         Unix-to-Unix Copy Program (540)
  whois        Nicname (43)
  www          World Wide Web (HTTP, 80)
```

Формат команды для работы с расширенным списком доступа приведен в табл. 9.5.

Таблица 9.5. Синтаксис команды создания расширенного списка доступа

Цель	Команда
Определить расширенный список доступа с указанием его номера и критериев фильтрации. Ключевое слово log позволяет вести журнал работы списка	access-list <номер списка доступа> {deny | permit} <имя протокола> <отправитель> <маска> <получатель> <маска> [precedence precedence] [tos tos] [established] [log]

И хотя синтаксис команды может показаться слишком сложным, ее суть остается такой же, как и для стандартного списка доступа. В данном случае предлагается много опций для достижения большей гибкости при создании правил. Рассмотрим следующий пример.

```
access-list 100 permit ip 192.168.3.0 0.0.0.255 192.168.25.0 0.0.0.31
```

Число 100 является номером расширенного списка доступа. Этот номер должен находиться в пределах от 100 до 199 включительно. Любой номер из диапазона 100–199 будет подходить к расширенному списку доступа, если он, конечно, не конфликтует с другими номерами. В примере команды запись 192.168.3.0 0.0.0.255 имеет такое же значение, как для стандартного списка доступа, и охватывает адреса отправителей. Напомним, что подобная запись говорит маршрутизатору о необходимости сравнивать первые три байта в шаблоне 192.168.3.0 с адресами отправителей поступающих пакетов. Последний байт не принимается в расчет, так как используется маска 0.0.0.255. Запись 192.168.25.0 0.0.0.31 является шаблоном для сравнения адресов получателей, границы которых определяются той же маской. Данное правило работает следующим образом: маршрутизатор будет пропускать дейтаграммы, адреса отправителей и получателей которых совпадут с определенными в правиле.

Для упрощения работы, как и в случае стандартного списка доступа, администратор может использовать ключевые слова any и host.

```
access-list 101 permit ip any host 192.168.25.10
```

Ввод подобной команды маршрутизатор воспримет как указание пропускать пакеты любого отправителя, адресованные устройству с адресом 192.168.25.10. Здесь не обязательно указывать маску, как уже оговаривалось при рассмотрении стандартных списков доступа (табл. 9.6).

Таблица 9.6. Синтаксис команды access-list с указанием сокращений протоколов

Цель	Команда
Определить расширенный список доступа используя сокращение для отправителя с маской, равной 0.0.0.0 255.255.255.255, и сокращение для получателя с аналогичной маской	access-list <номер списка доступа> {deny | permit}> <имя протокола> any
Определить расширенный список доступа используя сокращение для отправителя с маской, равной 0.0.0.0 (этот хост), и сокращение для получателя с аналогичной маской	access-list <номер списка доступа> {deny | permit} <имя протокола> host <отправитель> host <получатель>

Если администратор планирует осуществлять фильтрацию трафика, опираясь на информацию о протоколе, он также формирует список допустимых отправителей и получателей трафика и указывает тип протокола для сравнения. Например, можно предъявить требование маршрутизатору блокировать трафик, принадлежащий сеансу Telnet, следующими командами:

```
access-list 102 deny tcp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 eq Telnet
access-list 102 permit ip any any
```

Параметр tcp заставляет маршрутизатор пропускать только пакеты протокола TCP. Первый шаблон и маска (192.168.1.0 0.0.0.255) являются критериями для сравнения адресов отправителей пакетов, а критерием для сравнения адресов получателей является 192.168.2.0 0.0.0.255. Кроме того, указывается дополнительное условие для сравнения — eq Telnet, а именно дается указание на то, что порт на стороне получателя должен быть равен 23 (протокол Telnet). Ключевое слово eq является оператором сравнения, и в данном случае происходит проверка равенства. Администратор сети также может задействовать следующие операторы сравнения: neq (не равно), gt (больше чем), lt (меньше чем) и range (для указания диапазона номеров портов).

```
Cisco1601(config)#access-list 101 permit tcp any any ?
  Eq          Match only packets on a given port number
  Established Match established connections
  Gt          Match only packets with a greater port number
  Log         Log matches against this entry
  log-input   Log matches against this entry, including input interface
  lt          Match only packets with a lower port number
  neq         Match only packets not on a given port number
  precedence  Match packets with given precedence value
  range       Match only packets in the range of port numbers
  tos         Match packets with given TOS value
```

Если возвратиться к предыдущему примеру, то можно видеть, что в конце списка присутствует правило access-list 102 permit ip any any, которое разрешает прохождение любого трафика, противоречащего первому правилу списка доступа (ограничение протокола Telnet). Разрешающее условие необходимо, так как иначе невидимое правило автоматически блокирует любой трафик.

В итоге работу списка доступа 102 можно свести к следующим действиям:

- блокировать трафик, адресованный из сети 192.168.1.0 в сеть 192.168.2.0 (сравниваем первые 24 бита, согласно маске 0.0.0.255), где номер порта протокола TCP равен 23 (Telnet);
- весь остальной трафик протокола IP пропускаем (рис. 9.3).

Применение списков доступа с номером 102 на интерфейсе Ethernet1 маршрутизатора приведет к тому, что маршрутизатор будет блокировать сеансы Telnet, инициированные устройством в сети 192.168.1.0 и адресованные любому получателю в сети 192.168.2.0. Однако из последней сети сеансы Telnet блокироваться не будут. Чтобы запретить их, нужно применить список доступа также и в противоположном направлении.

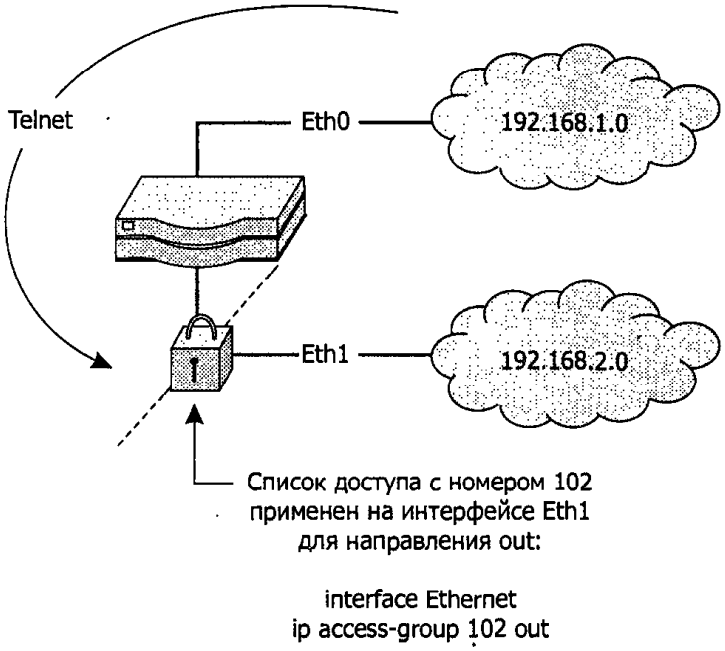


Рис. 9.3. Работа списка доступа

СОВЕТ

Администратор сети может использовать другой способ ограничения доступа к маршрутизатору по протоколу Telnet. Для этого нужно создать список доступа и применить его к виртуальным портам VTY, разрешая только доступ с определенных адресов. В данном примере доступ по протоколу Telnet ограничивается только с хостов, адреса которых принадлежат сети 192.168.3.0/24. Команда access-class 10 in применяет список доступа с номером 10 сразу на всех пяти виртуальных портах VTY (0–4):

```
access-list 10 permit 192.168.3.0 0.0.0.255
!
line vty 0 4
 access-class 10 in
```

Иногда возникает потребность в ведении журнала, фиксирующего факты отбрасывания маршрутизатором трафика, как определено расширенным списком доступа. Это особенно полезно в случае использования списков доступа для формирования политики безопасности и для намеренного блокирования сетевого взаимодействия, которое может представлять угрозу. Операционная система Cisco IOS позволяет вести журнал, указав ключевое слово log в правилах расширенного списка (для стандартных списков такая возможность отсутствует).

```
access-list 103 deny ip 192.168.6.254 0.0.0.0 192.168.3.254 log
access-list 103 permit ip any any
```

После активации журнала записываемые события доступны для просмотра с помощью команды show logging, или, если предварительно были сделаны соответствующие настройки, в журнале syslog-сервера.

СОВЕТ

В том случае, если в регистрируемых сообщениях отсутствуют временные отметки, можно настроить маршрутизатор на их использование с помощью команды service timestamps log datetime, выполняемой в глобальном режиме.

Маршрутизатор поддерживает счетчики пакетов для каждого задействованного правила. Такая возможность особенно полезна при решении проблем со списками доступа и для проверки их работы. Статистику по сравнению пакетов со списками доступа позволяет получить команда show access-list.

СОВЕТ

Если необходимо очистить счетчики, связанные со списками доступа, выполните команду clear counters, находясь в привилегированном режиме. Важно отметить, что статистика прохождения информации через интерфейсы локальных и глобальных сетей также будет сброшена, что не всегда желательно. Особенно в случае счетчиков на интерфейсах, например для подключенных сторонних компаний.

Защита от атак address spoofing

Остановимся более подробно на том, как использовать списки доступа для предотвращения атак типа address spoofing (подмена адресов), смысл которых заключается в том, что атакующий «втирается в доверие», используя адрес из списка допущенных к прохождению через маршрутизатор. Хотя это только один тип атаки из множества аналогичных, считается, что он является одним из самых популярных.

Наиболее часто списки доступа, блокирующие подобную атаку, настраиваются на маршрутизаторах, подключенных к Интернету. Опираясь на примеры, приведенные ниже, администратор может создавать и более изощренные списки доступа для защиты от сложных нападений, так как очевидно, что список доступа, эффективный против атаки spoofing, не сумеет обезопасить сеть от других атак, и, как правило, его следует использовать в сочетании с выделенным защитным экраном. Это может быть, например, Checkpoint Firewall-1 или маршрутизатор со специализированным программным обеспечением Cisco IOS Firewall Feature Set. Выделенные защитные экраны усиливают защиту от атаки address spoofing, реализуемую обычным списком доступа.

Рассмотрим пример, когда атакующий пытается выдать себя за хост, находящийся во внутренней сети с адресом 192.168.1.0/24, формируя, таким образом, атаку address spoofing (рис. 9.4).

Для предотвращения подобной атаки был создан список доступа с номером 101, который затем был применен на интерфейсе Serial0 для направления in (фильтруется входящий трафик).

```
Interface Serial0
 ip address 193.125.45.34
 ip access-group 101 in

access-list 101 deny ip 192.168.1.0 0.0.0.255 any
access-list 101 deny ip 192.168.2.0 0.0.0.255 any
access-list 101 deny ip 127.0.0.0 0.255.255.255 any
access-list 101 permit ip any any
```

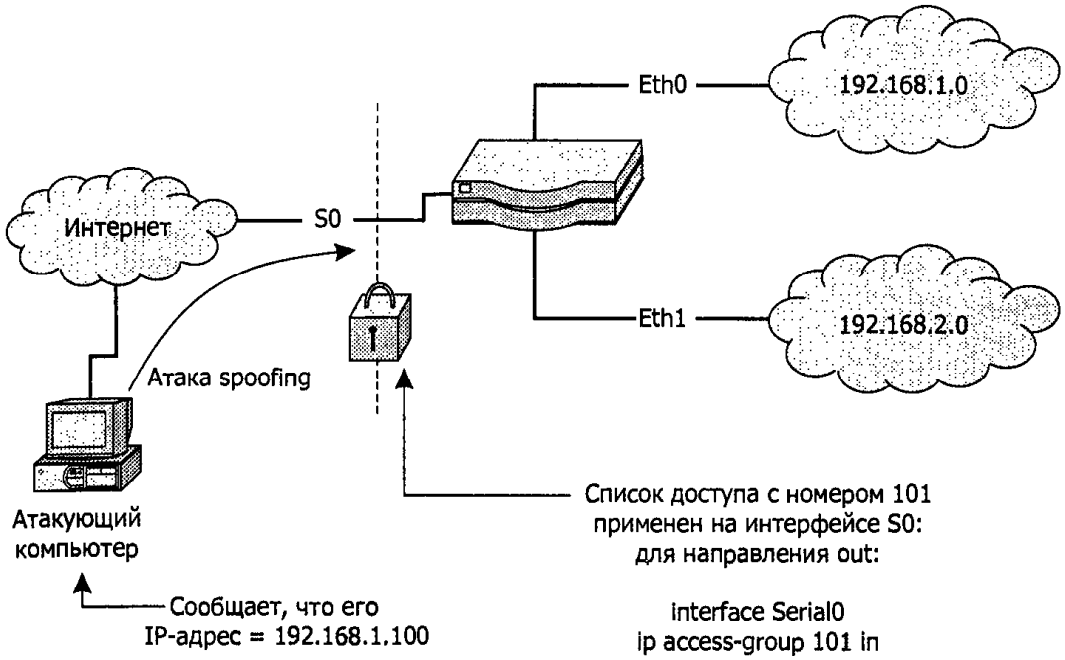


Рис. 9.4. Защита от атаки address spoofing

Этот список доступа «просеивает» входящий трафик из Интернета на интерфейсе Serial0. Первые два правила в списке доступа заставляют маршрутизатор блокировать весь трафик из Интернета, который имеет адреса отправителей, принадлежащих внутренним сетям (192.168.1.0/24 и 192.168.2.0/24). Такой фильтр создается из-за того, что администратор сети не ожидает поступления пакетов с внутренними адресами на внешний адрес Serial0 маршрутизатора, подключенного к Интернету. Очевидно, что любой пакет, поступающий из Интернета с адресом отправителя из внутренней сети, будет подозрителен. В рассматриваемом примере существуют две внутренние сети класса C, для которых и работают первые два правила в списке доступа. Если сеть компании состоит из большего количества сетей, то для каждой в отдельности нужно будет создать соответствующее правило.

Третье правило в списке доступа (access-list 101 deny ip 127.0.0.0 0. 255.255.255 any) вынуждает маршрутизатор отбрасывать все дейтаграммы, у которых первый байт адреса равен 127 (это адреса обратной связи — loopback — для всех устройств в сети). Подобные пакеты никогда не должны появляться из Интернета и передаваться по внутренней сети. Некоторые подтипы атак address spoofing используют данные зарезервированные адреса как адреса отправителей.

Продолжая разговор о возможных атаках из Интернета, следует остановиться на атаке smurf, которая является относительно новой в категории атак на сетевом уровне и относится к типу DoS. В этом случае атакующий посылает большое количество запросов протокола ICMP echo (команда ping), адресованных широковещательно. Причем все запросы имеют ложный адрес отправителя (тут есть варианты — можно использовать адреса отправителей из внутренней сети для проникновения через списки доступа, случайные или существующие, как будет показано далее). Если граничный маршрутизатор, с помощью которого организа-

ция подключается к Интернету, поддерживает передачу трафика с широковеща-тельным адресом, выполняя трансляцию широковещания на уровне протокола IP в канальный уровень, то большинство компьютеров во внутренней сети будут принимать эти запросы и отвечать на каждый из них, генерируя большой суммарный трафик. Кстати, родственной атакой является fraggle, когда вместо протокола ICMP используется протокол UDP, точнее одноименный запрос echo, выполняющий те же функции.

Успех атаки зависит от способности граничного маршрутизатора, обслуживающего большую сеть, транслировать широковещание на уровне IP (например адрес 10.255.255.255) в широковещание на канальном уровне (например для сети Ethernet FF:FF:FF:FF:FF:FF). Если обратиться к документу RFC 1812, содержащему требования к маршрутизаторам для протокола IP, то там можно найти следующий текст:

«Маршрутизатор **может** поддерживать возможность, которая запрещает получение широковещательных пакетов (направленного вещания — network-prefix-directed broadcasts) на сетевом уровне, и **должен** предоставлять настройку для запрещения трансляции полученного широковещательного трафика. Маршрутизатор **по умолчанию обязан поддерживать** получение и трансляцию широковещательных пакетов сетевого уровня».

Естественно, что, настраивая маршрутизатор, администратор должен отключить рассматриваемую возможность, чтобы защитить сеть от атаки. Если в сети используются маршрутизаторы Cisco Systems, то рекомендуется на них всех выполнить команду no ip directed-broadcast. Обратите внимание, что команда доступна в режиме настройки интерфейсов, следовательно, рекомендуется выполнять ее на всех интерфейсах. Кроме того, на маршрутизаторах с операционной системой Cisco IOS версии 12.0 и старше данная команда срабатывает по умолчанию.

С целью защиты от рассматриваемой атаки можно настроить компьютеры на блокирование формирования ответов на запросы ICMP echo, если в качестве адреса получателя указывается широковещательный адрес. В документе RFC 1122, описывающем требования к конечным хостам, говорится: «Запросы протокола ICMP echo, адресованные широковещательно... **могут игнорироваться** без уведомления отправителя...».

Однако большинство реализаций стека TCP/IP были разработаны с учетом ответа на запросы ICMP echo, несмотря на риск пострадать от атаки smurf, здесь для конечных компьютеров предпочтительнее было бы просто игнорировать запросы. Администратору следует проверить наличие дополнительных пакетов для используемых в организации операционных систем, которые после установки нового программного обеспечения меняют поведение стека в этом аспекте.

Учитывая особенность атаки, можно выделить два сетевых объекта, подверженные риску в наибольшей степени: атакуемая система и система с адресами, которые были указаны в качестве адресов отправителя. Рассмотрим пример сети из 100 компьютеров, имеющей высокоскоростной канал подключения к Интернету. Теперь предположим, что атакующий генерирует постоянный поток запросов ICMP echo на скорости 768 Кбит/с с ложными адресами. Все 100 компьютеров, получая запросы, будут формировать ответы на них, причем в совокупности ответы могут занять полосу пропускания ориентировочно 78 Мбит/с, и весь это

совокупный трафик будет посылаться по указанному в запросах адресу. Таким образом, атакующий может опосредованно воздействовать на объект нападения. Можно отметить исследования Крейга Хьюгена, который показал, что для локальной сети, состоящей из 200 компьютеров, оказалось возможно сгенерировать трафик объемом 80 Мбайт/с, направленный на стороннюю организацию. Столь большие объемы легко объяснимы: при отправке потока запросов со скоростью 400 Кбит/с широковещательно ответы умножаются на количество компьютеров в сети. Обратите внимание, что атакующему не нужно иметь высокоскоростное подключение к Интернету — в рассматриваемом примере хватит скорости в 512 Кбит/с (тут уже можно судить об особенностях атаки — работая по модему, ее организовать затруднительно).

Здесь следует отметить одну интересную особенность, связанную со списками доступа на маршрутизаторах Cisco Systems. Блокирование трафика атаки smurf списком доступа выполняется относительно медленно, так как в некоторых случаях требуется посылка отправителю (точнее, по указанным адресам — предполагается, что атакующий редко будет демонстрировать свой реальный адрес) сообщений ICMP unreachable. Блокирование входящих запросов ICMP echo на граничном маршрутизаторе защитит внутреннюю сеть, но не поможет снизить нагрузку на канал подключения к Интернету, поскольку хотя запросы и будут блокироваться маршрутизатором, в ответ могут генерироваться сообщения, указывающие на недостижимость получателя. Учитывая это, администраторам сетей, подвергающимся рассматриваемой атаке, стоит посоветовать обратиться в группу технической поддержки провайдера с просьбой выполнять фильтрацию трафика на их маршрутизаторы (если это, конечно, возможно организационно и технически). Кстати, принцип больших объемов ответного трафика, лежащий в основе этой атаки, можно использовать, например, для «финансового давления». Часто при подключении по высокоскоростному каналу к Интернету организация платит за входящий трафик, поэтому атакующий, указывая в широковещательных запросах адреса отправителей этой организации, способен неприятно удивить последнюю крупными счетами от провайдера. Обнаружение атаки можно усложнить, если организовывать ее ночью в надежде, что у жертв выключены программные комплексы обнаружения атак. Днем большие объемы необъяснимого трафика обнаружить несложно простым наблюдением за индикаторами на интерфейсах маршрутизатора (конечно, это не самый удобный способ).

При использовании списков доступа для фильтрации входящего трафика администратор может включить опцию ведения журнала (ключевое слово log при формировании правил списка доступа). Однако ведение журнала требует задействования ресурсов на маршрутизаторе, и при массированной атаке это опасно тем, что центральный процессор будет заниматься только журналом. Более того, если сообщения перенаправляются на сервер syslog, то во время атаки их количество может превысить размер допустимого.

Если в сети используется система обнаружения атак (Intrusion Detection System — IDS), то факт осуществления атаки выявляется довольно просто — по повышенному количеству запросов и ответов протокола ICMP. Однако если в сети для административных целей посылается большое количество запросов ping, то это может привести к ложному срабатыванию системы обнаружения

атак. Другим критерием обнаружения может быть присутствие данных в запросе ICMP echo (для увеличения объема трафика).

Для противодействия атаке рекомендуется настроить граничный маршрутизатор или защитный экран на блокирование прохождения в сеть запросов ICMP echo. Кроме того, в настоящее время большинство производителей маршрутизаторов встраивают команду, позволяющую блокировать прохождение так называемых spoofing-пакетов во внутреннюю сеть, проверяя помимо адресов получателей также адреса отправителей, чтобы убедиться, что обратный путь пакета будет проходить через тот же интерфейс маршрутизатора, через который он был получен.

Следует отметить, что схожий принцип используется в атаке Ping Flooding (шторм запросов ping), когда атакующий пытается нарушить работу сети, формируя постоянный поток направленных запросов ICMP echo.

АТАКА CISCO IDENT

Некоторое оборудование, произведенное компанией Cisco Systems, поддерживает очень простой протокол идентификации, который годится для определения типа этого оборудования. Когда на таком маршрутизаторе устанавливается соединение протокола TCP с портом назначения 1999, маршрутизатор высылает сегмент TCP с установленным флагом RST (нормальное поведение), а в поле данных этот сегмент будет содержать слово cisco. Противодействие атаке может заключаться в блокировании порта 1999 для протокола TCP на граничном маршрутизаторе, что будет препятствовать атакующему опрашивать оборудование.

Продолжая рассмотрение вопросов защиты внутренней сети от внешних атак, следует обратить внимание на возможности дополнительного программного обеспечения маршрутизатора Cisco IOS Firewall Feature Set, которое приобретается отдельно и после установки добавляет к маршрутизатору функции защитного экрана. Использование этого программного обеспечения особенно привлекательно в тех ситуациях, когда в компании уже установлены маршрутизаторы Cisco Systems и ей требуется улучшить безопасность на внешних подключениях. Такое решение позволяет совместить функциональные возможности маршрутизатора и защитного экрана в одном устройстве.

АТАКА, ОСНОВАННАЯ НА ФРАГМЕНТАЦИИ ДЕЙТАГРАММ IP

Дейтаграмма протокола IP иногда может фрагментироваться при передаче по сети. Эти фрагменты будут собираться на стороне получателя и сформируют полную дейтаграмму. Некоторые маршрутизаторы, которые выполняют функции фильтрации трафика, основываясь на информации в заголовке протокола TCP, проверяют только первый поступивший фрагмент дейтаграммы, а остальные пропускают без проверки. Учитывая такой подход, атакующий может создать фрагменты таким образом, что они будут как бы перекрывать друг друга. В результате не исключена перезапись частей заголовка протокола TCP при сборке фрагментов получателем. При этом фильтрующий маршрутизатор будет считать, что пакет адресован разрешенному порту. Очевидно, что подобная атака в первую очередь ориентирована на прохождение через списки доступа на граничном маршрутизаторе или на защитном экране.

Если обратиться к практике, то, по опыту автора, администратор сети должен учитывать три основных составляющих защиты сети: установку защитного экрана, внедрение системы обнаружения атак и использование сканера безопа

ности. Хотя, пожалуй, подобные меры призваны в большей степени защитить от атак извне (чаще всего из Интернета), так как защита изнутри помимо технических мер опирается на организационные мероприятия, приучающие пользователей обращать повышенное внимание на вопросы безопасности. Действительно, если, например, конкурент задастся целью получить некую конфиденциальную информацию о компании (цены, списки дебиторов и кредиторов и т. п.), то гораздо проще, например, подкупить одного из сотрудников и с его помощью проникнуть в сеть изнутри, чем нанимать профессиональных взломщиков для организации проникновения снаружи. Тем более, что традиционно сотрудники технических отделов оказывают больше внимания защите от атак извне, и по этой причине взломать сеть будет достаточно сложно.

Однако следует учесть участившиеся факты простого вандализма, когда за попыткой вывести из рабочего состояния корпоративный web-сервер стоят не коммерческие, а «познавательные» цели. Именно для этого стоит обращать внимание на организацию грамотной защиты периметра сети. Сначала рекомендуется выбрать подходящий защитный экран, тем более что в настоящее время рынок переполнен предложениями для разных платформ и размеров сетей. Если средства не позволяют, то можно просто настроить граничный маршрутизатор на совместное использование технологии NAT и списков доступа. Этого бывает вполне достаточно, учитывая, что большинство программных защитных экранов, собственно говоря, и представляют собой такую комбинацию, интегрируясь в операционную систему и перехватывая пакеты на одном из нижележащих уровней стека TCP/IP в соответствии с определенными загодя правилами.

АТАКА PING OF DEATH

Эта атака относится к типу DoS и заключается в добавлении очень большого объема информации к запросу протокола ICMP echo, что может привести к сбою операционной системы. Следует обратить внимание, что хотя при нормальной работе сети и возможно появление сообщения ICMP echo с большим объемом дополнительных данных (более 4000 байт), на практике это очень редкая ситуация. Кроме того, если атакующий инициирует направленную атаку данного типа на защищенную систему, то ответы также будут относиться к типу Ping Of Death. Для противодействия можно настроить граничный маршрутизатор на блокирование входящих запросов ICMP echo. Однако здесь стоит обратить внимание на некоторые моменты, связанные с работой списков доступа, которые можно найти в описании принципов атаки smurf.

Другим важным элементом системы защиты является система обнаружения атак. Архитектура подобной системы часто строится из агентов и станции управления, на которую поступает информация. Причем типы агентов могут отличаться предполагаемым местом их установки. Так, компания ISS (Internet Security System — www.iss.net) предлагает продукт ISS RealSecure, который и призван обеспечить обнаружение и реагирование на атаки. Относительно атак из Интернета, по мнению автора, наиболее удачным будет использование агента Network Scanner, входящего в состав продукта. Этот агент устанавливается на выделенный компьютер, работающий, например, под управлением операционной системы Windows NT, и анализирует трафик на сегменте сети, к которому он подключен (рис. 9.5).

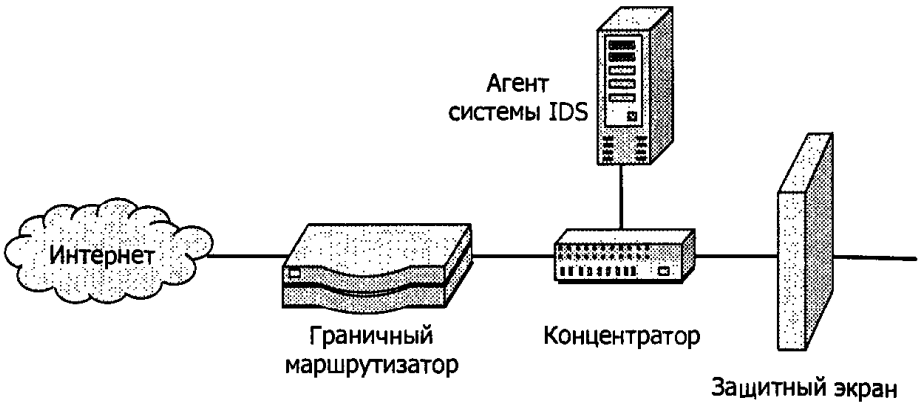


Рис. 9.5. Использование агента Network Scanner

В данном примере трафик, поступающий из Интернета, будет проходить через граничный маршрутизатор и защитный экран, причем последний будет либо отбрасывать его, либо пропускать. Учитывая, что связующим звеном выступает концентратор, который по характеру своей работы передает весь проходящий трафик на все свои порты, то агент системы обнаружения атак сможет анализировать не только трафик, направленный ему, но также и весь трафик, направленный в защищаемую сеть. В общих чертах работу агента можно охарактеризовать следующим образом: он выполняет сравнение трафика с так называемыми сигнатурами известных типов атак, и если находит совпадение, то производит определенные действия (например, посылает письмо администратору сети). В любом случае, агент ведет журнал событий, который является источником довольно ценной информации о предпринимаемых попытках воздействия на сеть.

АТАКА SYN FLOODING (ШТОРМ ЗАПРОСОВ TCP SYN)

Данный тип атаки был уже рассмотрен в главе 5, посвященной протоколу TCP. Обобщая, отметим, что атака относится к типу DoS и является ее классическим представителем (по крайней мере, если судить по количеству материала в Интернете). При использовании в сети системы IDS важно иметь в виду, что существование посещаемого WWW-сервера в компании может вызывать ложные срабатывания из-за большого количества сеансов протокола TCP, устанавливаемых с сервером за короткий интервал времени. Работая с системой ISS RealSecure, администратор сети при наличии фактов ложных срабатываний может настроить параметры агента. Роль потенциальной жертвы атаки в принципе способно играть любое устройство в сети, принимающее запросы протокола TCP и имеющее ограниченную очередь незавершенных запросов. Самым простым способом противодействия этой атаке является обычная перезагрузка атакуемого компьютера, так как это освобождает очередь запросов на установление соединения TCP. Если используется система IDS, ее рационально настроить на уничтожение неактивных соединений. Естественно, излишне говорить о необходимости установить последние обновления от производителя операционной системы, так как в настоящее время большинство популярных операционных систем (точнее, реализация стека TCP/IP в них) поддерживают механизмы, позволяющие в той или иной мере противодействовать атаке SYN Flooding.

Следует отметить, что в настоящее время некоторые возможности систем обнаружения атак встроены в защитные экраны. Так, например, защитный экран GuardianPro v5.02 обнаруживает и блокирует атаки address spoofing. При этом факт подобной атаки записывается в журнал. Также допустимо настроить

правку электронной почты. При настройке стратегии защиты на защитном экране администратор сети может включить защиту от атаки spoofing и указать, какие именно адреса следует защищать.

На рис. 9.6 показано окно настройки стратегии безопасности. Здесь администратор может настроить защитный экран на выполнение контроля атаки address spoofing (Enable Spoofing control) с прямым указанием охраняемых сетей (Protected addresses).

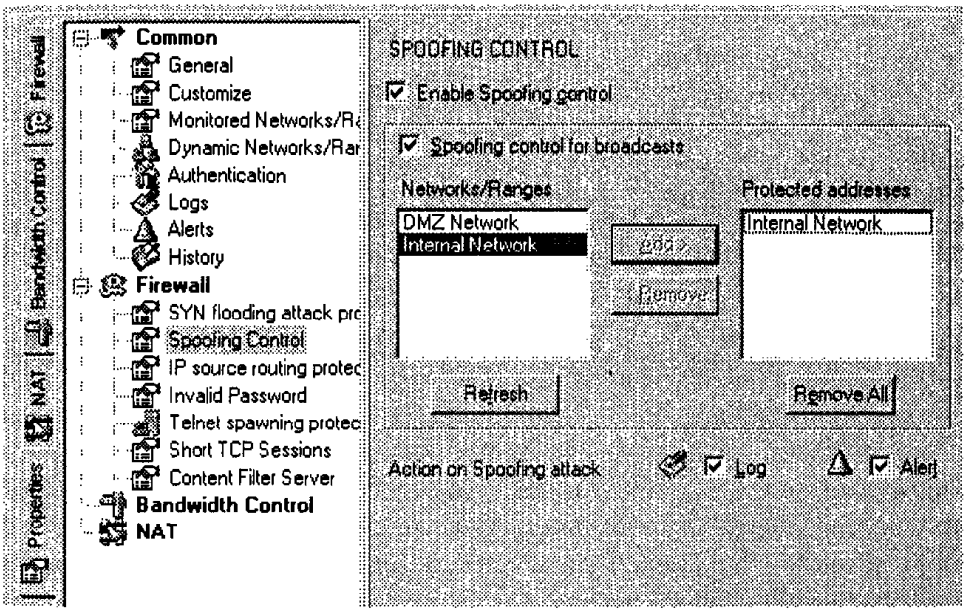


Рис. 9.6. Настройка GuardianPro на защиту от атаки address spoofing

Безопасность удаленного доступа

Продолжая разговор о базовой безопасности, следует отметить, что, как правило, любая крупная сеть предоставляет пользователям возможность работать с удаленным доступом. При этом наиболее популярным решением организации удаленного доступа является использование аналоговых модемов, подключенных к какому-либо серверу доступа. Сам выбор сервера доступа напрямую зависит от количества претендентов, желающих воспользоваться таким сервисом. Однако помимо организационной выгоды от предоставления удаленного доступа необходимо обратить пристальное внимание на такие вопросы, как аутентификация (authentication), авторизация (authorization) и учет (accounting). В англоязычной литературе эти меры часто обозначают как «строенное А» (tripple A — AAA).

Аутентификация заключается в проверке пользовательского имени и пароля. Данная задача и, собственно, ее реализация является центром любой системы безопасности удаленного доступа, и именно аутентификация отвечает за доступ пользователя в сеть, опираясь на тот факт, что система знает этого пользователя. Для последнего процесс аутентификации представляется проверкой его имени (login, username) и пароля (password).

Авторизация призвана ограничить действия пользователя в сети, обычно посредством добавления пользователя в какую-либо группу с наложенными ограничениями. Это могут быть ограничения, связанные со временем разрешенного входа в сеть (день, час), тип используемого протокола (IP, IPX), максимальное количество одновременных сеансов для одной учетной записи (точнее, для одной комбинации имени и пароля) и т. п.

С помощью учета, как можно понять из названия, отслеживается, что пользователь делает или делал в процессе работы с сетью, работая в режиме удаленного доступа. Обычно наибольшее внимание уделяется ведению журнала, содержащего информацию о начале и конце сеанса удаленного доступа, о количестве полученных и переданных байтов, о том, по какой линии передавались данные и т. п.

ПРИМЕЧАНИЕ

Применительно к маршрутизаторам и серверам доступа компании Cisco Systems технология AAA помимо управления доступом удаленных пользователей в сеть может также использоваться для контроля доступа технического персонала к интерфейсу командной строки IOS на сетевых устройствах.

Ключевым компонентом технологии AAA является централизованная база данных, располагаемая на выделенном сервере и содержащая такую информацию, как имена пользователей и соответствующие пароли, группы и связанные с ними правила и т. п. Централизованное хранение информации помогает значительно упростить администрирование сети, если администратор вынужден поддерживать множество точек доступа в сеть. Альтернативой централизованного доступа может быть поддержка локальных баз данных на каждом маршрутизаторе, однако последний подход не обеспечивает всей гибкости и возможностей централизованного управления.

Внедряя технологию AAA в сети, следует обратить особое внимание на механизм взаимодействия маршрутизаторов и сервера AAA — для этого были разработаны специализированные протоколы, наиболее популярными среди которых являются RADIUS (Remote Access Dial-In User Service) и TACACS+ (Terminal Access Controller Access Control System). Несмотря на некоторые технические расхождения в методе их работы, которые будут рассмотрены далее, основная цель остается неизменной — они обеспечивают передачу запросов и ответов между клиентом (в нашем случае маршрутизатором) и сервером AAA для выполнения аутентификации, авторизации и учета.

Поддержка протокола RADIUS была введена в операционную систему Cisco IOS начиная с версии 11.1, и это позволяет, задействуя оборудование компании Cisco, строить крупные центры обработки удаленных пользователей. Чаще всего подобное решение востребовано компаниями, предоставляющими удаленным пользователям доступ в Интернет.

Изначально протокол RADIUS был представлен компанией Livingston Enterprises как средство для аутентификации и учета. Затем, поддержанный различными производителями оборудования для удаленного доступа, он был принят в качестве стандарта. В 1996 году спецификация протокола была передана на утверждение организации IETF, и в настоящее время полное описание стандарта можно найти в документах RFC 2058 и RFC 2059.

Работа самого протокола опирается на популярную модель «клиент-сервер». При этом сервер доступа (например, Cisco 2511)[1] с точки зрения протокола выступает клиентом и отвечает за передачу пользовательской информации к выделенному серверу и затем принимает дальнейшее решение на основе ответа от сервера. Сервер RADIUS может предоставлять функции аутентификации и учета для множества клиентов, и он отвечает за получение запроса на подключение удаленного пользователя, его аутентификацию и отправку клиенту информации, которая необходима для предоставления требуемого уровня сервиса.

В настоящий момент существует множество программных средств разных производителей, реализующих сервер RADIUS, причем для разных платформ. Компания Cisco Systems предлагает свое собственное решение CiscoSecure, которое также поддерживается для платформ UNIX и Microsoft Windows. Хочется отметить, что, по опыту автора, наиболее удобной является реализация, поддерживающая различные методы хранения информации о пользователях, так как при относительно большом количестве клиентов встает вопрос о введение некой схемы оплаты предоставляемых услуг. При этом возможность, например, размещать информацию имени пользователя и его пароле в базе данных SQL часто является решающим фактором при выборе программного обеспечения с функциональностью сервера RADIUS (сказанное также относится и к серверу TACACS+).

Взаимодействие между клиентом и сервером RADIUS опирается на протокол UDP. Выбор именно этого протокола в качестве базового был продиктован тем, что вопросы повторной передачи утерянных пакетов, обработка временных задержек и т. п. обеспечиваются клиентом и сервером протокола RADIUS. Кроме того, наиболее популярной конфигурацией является ситуация, когда клиент и сервер RADIUS связаны надежной локальной сетью. Хотя, конечно, нет никаких ограничений на расположение клиента и сервера в разных частях распределенной сети.

Говоря обобщенно, проверка имени и пароля пользователя состоит в отправке запроса Access-Request (запрос на доступ) от клиента к серверу RADIUS и последующих ответов Access-Accept (доступ разрешен) или Access-Reject (доступ запрещен) от сервера. Пакет Access-Request содержит в себе имя пользователя, зашифрованный пароль, IP-адрес клиента RADIUS (сервера доступа) и номер порта, на который обратился пользователь. Кроме того, в данный пакет включена служебная информация о типе сеанса, которую запрашивает пользователь: например, если подключение выполняется по протоколу PPP, то поле Service-Type (тип сервиса) устанавливается в значение Framed-User, а поле Framed-Type — в значение PPP.

Когда сервер получает запрос от клиента на проверку пользователя, он проверяет наличие последнего в своей базе данных учетных записей. Если указан-

[1] Автор не нашел четкого разделения между терминами «сервер доступа» и «маршрутизатор». Так, устройство Cisco 2511, имея 16 асинхронных портов, также снабжается портом Serial для подключения к высокоскоростной сети и будет работать как маршрутизатор, принимая трафик от асинхронных портов и порта Ethernet и маршрутизируя его, например в Интернет, через интерфейс Serial.

ная комбинация имени и пароля не присутствует в базе данных, то сервер сразу посылает сообщение Access-Reject, которое может также сопровождаться дополнительным текстовым сообщением, объясняющим причину отказа в доступе. В том случае, если имя пользователя найдено и его пароль корректен, сервер возвращает сообщение Access-Accept, включающее в себя список атрибутов, которые определяют параметры для данного сеанса. Обычные параметры — это тип протокола, адрес IP для пользователя (статический или динамический), список доступа для применения к этому пользователю или статический маршрут, добавляемый к таблице маршрутизации сервера доступа.

Аутентификация пользователя является наиболее сложным аспектом обеспечения безопасности. Стандартная реализация протокола RADIUS поддерживает разные методы аутентификации, включая протоколы PAP и CHAP. Однако решения различных производителей могут добавлять поддержку других протоколов.

ПРИМЕЧАНИЕ

Компонент IAS компании Microsoft, входящий в состав операционной системы Windows 2000 Server, поддерживает следующие протоколы: PAP, CHAP, MS-CHAP версий 1 и 2 и EAP (Extensible Authentication Protocol). Протокол EAP предоставляет организации целую инфраструктуру обеспечения безопасности, поддерживая такие возможности, как пластиковые карты, клиентские сертификаты и т. д.

Протокол RADIUS является только одним из вариантов обеспечения функций аутентификации и учета. Другим достаточно широко распространенным протоколом является TACACS+, разработанный компанией Cisco Systems. Несмотря на общую цель, существуют существенные расхождения в деталях работы этих протоколов. Фундаментальное различие заключается в используемом транспортном протоколе: RADIUS опирается на протокол UDP, в то время как TACACS+ использует TCP (порт 43). Учитывая, что протокол TCP самостоятельно обеспечивает гарантированную доставку и другие возможности, связанные с работой протокола, ориентированного на установление соединения между абонентами, протоколу TACACS+ нет необходимости самостоятельно их обеспечивать, в отличие от RADIUS.

Другое отличие заключается в том, что в сообщении Access-Request, передаваемом от клиента к серверу, протокол RADIUS шифрует только пароль пользователя, оставляя остальную часть пакета незащищенной, в том числе имя пользователя. По утверждению компании Cisco Systems, это делает протокол уязвимым для перехвата информации. Однако, по мнению автора, говоря об уязвимости, следует уточнять конкретную сетевую конфигурацию. Если сервер доступа установлен за защитным экраном с одной стороны, а сервер RADIUS с другой и в качестве сетевых устройств используются коммутаторы, то вопрос об уязвимости можно подвергнуть сомнению. Вместе с тем, действительно, если на одном из компьютеров в локальной сети установить анализатор пакетов, то при использовании концентраторов, передающих трафик на все порты, перехват и получения информации об имени пользователя становится несложным делом.

В противоположность протоколу RADIUS, TACACS+ шифрует весь пакет, оставляя в открытом виде только заголовок, в котором присутствует поле, указывающее на то, зашифровано или нет тело сообщения. При нормальной работе тело сообщения шифруется полностью.

Практическая настройка технологии Cisco AAA

Для того чтобы аутентифицировать пользователя, администратор сначала должен определить так называемый аутентификационный список, формирующий последовательность действий, которые обязан предпринять маршрутизатор для проверки пользовательского имени и пароля. Например, список может указать сначала попробовать обратиться к протоколу RADIUS, а если попытка его применения не будет успешной, проверить локальную базу имен и паролей. После того как список создан, он применяется на интерфейсах маршрутизатора, на которые обращаются пользователи.

Для того чтобы включить поддержку технологии AAA на маршрутизаторе, администратору следует ввести в глобальном режиме команду **aaa new-model**. Далее нужно создать именованный список аутентификации с помощью команды **aaa authentication**.

```
Cisco2511RJ#conf terminal
Enter configuration commands, one per line. End with CNTL/Z.
Cisco2511RJ(config)#aaa new-model
Cisco2511RJ(config)#aaa authentication ppp L1 radius local
```

В приведенном примере создается новый список с именем L1, указывающий, что в качестве первого метода аутентификации необходимо выбрать протокол RADIUS (точнее, сервер безопасности, поддерживающий протокол RADIUS), а если сервер не удастся использовать (например, он выключен), то маршрутизатор будет пытаться проверить имя пользователя и пароль в своей локальной базе (ключевое слово **local**). Следует обратить внимание, что список создается для проверки пользователей, подключившихся по протоколу PPP (Point-to-Point Protocol), который в настоящее время наиболее популярен для подключения удаленных пользователей и офисов компании через коммутируемые соединения.

Создав список, его следует применить на интерфейсе, который будет принимать входящие звонки. Для асинхронного интерфейса последовательность команд может быть следующей:

```
Cisco2511RJ(config)#int async 16
Cisco2511RJ(config-if)#ppp authentication chap L1
```

В команде ключевое слово **chap** указывает на задействуемый протокол передачи имени и пароля от удаленного пользователя к маршрутизатору. В нашем случае таким будет одноименный протокол CHAP (Challenge Handshake Authentication Protocol), который шифрует передаваемую информацию. В распоряжении администратора также есть протокол PAP (Password Authentication Protocol), выполняющий те же функции, что и CHAP, за исключением шифро-

вания передаваемых данных (пароль передается открытым текстом). Список доступных для использования протоколов можно получить с помощью команды ppp authentication ?.

В примере выше созданный список аутентификации применялся только на одном интерфейсе (номер 16), в то время как на практике чаще всего приходится работать с группой асинхронных интерфейсов, что позволяет снизить затраты по управлению. Так, на маршрутизаторе может быть создан логический интерфейс group-Async 1, который объединяет все 16 физических асинхронных интерфейсов. Это осуществляется с помощью команды group-range с указанием начального и конечного номеров объединяемых интерфейсов.

```
Cisco2511RJ(config)#int group-Async 1
Cisco2511RJ(config-if)#group-range 1 16
Cisco2511RJ(config-if)#ppp authentication chap L1
```

Вместо создания именованных списков аутентификации администратор вправе определить список по умолчанию (default authentication list), который идентифицируется с помощью ключевого слова default. Маршрутизаторы Cisco Systems будут использовать список по умолчанию, если администратор не определил именованных списков. Следующий пример показывает настройку списка аутентификации по умолчанию с двумя методами проверки пользователей (первый с помощью протокола RADIUS, второй с помощью локальной базы пользователей) и включает поддержку протокола CHAP на асинхронном интерфейсе.

```
Router#conf terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#aaa new-model
Router(config)#aaa authentication ppp default radius local
Router(config)#int async 1
Router(config-if)#ppp authentication chap
```

Следует отметить, что AAA — это не только способ контроля подключения удаленных пользователей к сети. Администратор сети вправе применить эту технологию также и для контроля доступа к маршрутизаторам. Если сеть обслуживается силами одного специалиста, то такой подход может быть избыточным, однако все его достоинства проявляются, если существует группа специалистов, занятых управлением и настройками маршрутизаторов. С помощью технологии AAA можно проверять каждого отдельного технического специалиста и вести журнал изменений. Даже если задачи ведения журнала не очень важны для организации, сам факт предоставления каждому пользователю имени и пароля доступа является хорошей политикой безопасности по сравнению с поддержкой одного пароля для команды enable. Кроме того, внедрение AAA позволит выполнять периодическое централизованное изменение паролей доступа.

Контроль доступа к командной строке маршрутизатора аналогичен задачам, решаемым при контроле доступа с помощью протокола PPP для удаленных пользователей: необходимо создать список аутентификации и применить его на консольном порте (console) и виртуальных портах VTY. Следующ

пример показывает настройку списка и применение его к виртуальным портам с номерами 0–4.

```
Router#conf terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#aaa new-model
Router(config)#aaa authentication login L3 radius line enable
Router(config)#line vty 0 4
Router(config-line)#login authentication L3
```

Здесь команда aaa authentication login L3 radius line enable создает именованный список аутентификации L3 для проверки доступа к командной строке маршрутизатора. Созданный список заставляет маршрутизатор проверять имя пользователя и пароль с помощью протокола RADIUS (если указать ключевое слово tacacs+, то маршрутизатор будет использовать сервер TACACS+), а если проверка будет неуспешной по любым причинам, то предписывает проверить пароль, присвоенный виртуальным портам VTY (доступ с помощью протокола Telnet). Если же пароль доступа не был настроен, то маршрутизатор будет пытаться использовать пароль enable. Важно отметить, что использование пароля enable в качестве «последней инстанции» очень рекомендуется, так как помогает предотвратить ситуацию, когда доступ к маршрутизатору станет невозможным ввиду проблем с сервером RADIUS (или TACACS+).

Если администратор по каким-либо причинам не может задействовать протоколы RADIUS/TACACS+, то в его распоряжении есть возможность поддерживать базу данных пользователей локально на маршрутизаторе (так называемая local username database). Такая возможность полезна в ситуации, если необходимо обслуживать небольшое количество пользователей, так как в этом случае все удобства централизованной базы будут отсутствовать. Важно отметить, что локальная база может работать как резервный источник аутентификации. Для этого нужно настроить список аутентификации с ключевым словом local, следующим за ключевыми словами radius или tacacs+.

```
Router(config)#aaa authentication ppp L4 radius local
```

Для создания локальной базы пользовательских имен и паролей следует выполнить команду username, вводимую в глобальном режиме. В качестве параметра команды указываются имя и пароль (непосредственно пароль после ключевого слова password).

```
Router(config)#username user1 password pass1
Router(config)#username user2 password pass2
```

Для того чтобы введенные пароли хранились в зашифрованном виде, предусмотрена команда service password-encryption, также вводимая в глобальном режиме.

Как уже отмечалось, задача авторизации (authorization) в технологии AAA заключается в определении тех действий, которые пользователю позволено выполнять во время сеанса. Администратор вправе контролировать доступ, например, в соответствии со временем и днем недели, используемым протоколом

и т. д. в зависимости от возможностей сервера AAA. При этом сервер доступа, работая клиентом по протоколу RADIUS/TACACS+, будет получать информацию об авторизации и назначать указанные привилегии пользователю.

Чтобы настроить маршрутизатор на опрос сервера AAA для авторизации пользователей, работающих по протоколу PPP, следует выполнить команду aaa authorization network, как показано в следующем примере (если протоколом авторизации является TACACS+, нужно ввести ключевое слово tacacs+ вместо radius).

```
Router(config)#aaa authorization network radius none
```

В этой команде ключевое слово none указывает маршрутизатору на необходимость успешно заканчивать авторизацию даже в том случае, если сервер RADIUS недоступен. Аналогично, для настройки авторизации доступа к командной строке маршрутизатора следует вместо network указать ключевое слово exec.

```
Router(config)#aaa authorization exec radius none
```

Как и при авторизации, возможности третьего A — учета (accounting) значительно зависят от реализации сервера AAA. В любом случае, как и для авторизации, администратор должен напрямую указать маршрутизатору использовать команды учета протоколов RADIUS или TACACS+. Операционная система Cisco IOS поддерживает три метода ведения учета.

Первый метод (stop-only) предполагает, что маршрутизатор посылает учетную информацию (accounting records) серверу AAA по протоколу RADIUS или TACACS+ только в том случае, когда пользовательский сеанс завершается. Очевидно, что этот метод предоставляет минимальные возможности для ведения учета. Второй метод (start-stop) подразумевает, что учетная информация посылается в начале и конце пользовательского сеанса. Третий метод (wait-start) предполагает, что записи посылаются серверу AAA в начале сеанса, а затем маршрутизатор ожидает подтверждения от сервера для начала обслуживания пользователя. Учетная информация также посылается при завершении удаленного сеанса.

Сама учетная информация передается в служебных пакетах протоколов RADIUS/TACACS+ и включает в себя время начала и конца сеанса, его длительность, количество переданных и полученных байтов, вводимые команды и т. п.

Для настройки функции учета на маршрутизаторе администратору следует выполнить команду aaa accounting, вводимую в глобальном режиме. Следующий пример показывает настройку учета на основе протокола RADIUS для пользователей, работающих по протоколу PPP.

```
Router(config)#aaa accounting network start-stop radius
```

Ключевое слово start-stop указывает маршрутизатору посылать серверу RADIUS информацию о начале и конце сеанса пользователя, как было отмечено ранее. Если следует вести учет для пользователей, работающих с командной строкой маршрутизатора, следует вместо network указать ключевое слово exec.

После того как администратор настроил все три элемента технологии AAA, ему остается информировать маршрутизатор, какой сервер (и протокол) дей

вителей при проверке пользователей. Для этого нужно указать IP-адрес сервера и специальный закрытый ключ (security key) для обеспечения безопасности взаимодействия. Без этих двух параметров маршрутизатор не сможет работать с сервером по протоколу RADIUS или TACACS+. Закрытый ключ предназначен для шифрования передаваемых данных и для аутентификации клиента (маршрутизатора) на сервере. Следующий пример показывает настройку рассмотренных параметров на маршрутизаторе.

```
Router(config)#radius-server host 192.168.3.253
Router(config)#radius-server key dybvfybt
```

При вводе команды radius-server host можно вместо адреса ввести имя хоста, и если на маршрутизаторе настроена поддержка DNS, он выполнит преобразование имени в адрес и подставит последний. Вторая команда указывает закрытый ключ (в нашем примере dybvfybt). Если администратор склонился в сторону протокола TACACS+, то для него нужно ввести аналогичные команды, только вместо ключевого слова radius-server указать tacacs-server.

10 Построение виртуальных частных сетей

Введение

В данной главе детально рассматриваются вопросы построения виртуальных ча
ных сетей на базе протокола PPTP (Point-to-Point Tunneling Protocol — проток
туннелирования «точка-точка»), поддержка которого входит в задачи операцио
ной системы Microsoft Windows 2000 Server (как, впрочем, и в маршрутизато
Cisco Systems). Необходимость применения технологии виртуальной частно
сети (Virtual Private Network — VPN) наиболее часто возникает, когда необх
димо подключить один или несколько удаленных объектов, которые невозмо
но (или нецелесообразно экономически) подключать традиционными метода
(например, привлекая провайдера услуг связи). Основным материалом для да
ной главы послужила практика подключения одного из удаленных объект
к корпоративной сети. Кроме того, материал главы будет отталкиваться от опы
автора по построению сети на основе самостоятельного формирования канал
связи по прямым медным проводам (в масштабах Санкт-Петербурга). Читате
может показаться, что связи между прямыми (от объекта до объекта) прово
ми и технологией VPN нет, но как будет показано далее, одно вытекает из др
гого, особенно если было принято решение строить сеть своими силами.

При создании распределенной компьютерной сети организация высокоск
ростной связи является наиболее сложной частью общей задачи. При этом ре
идет только о высокоскоростных каналах, а не о связи с использованием ана
говых модемов. Конечно, как вариант, в некоторых случаях и аналоговые мо
мы могут обеспечить соединение, но при этом следует помнить о том, что
минимум необходим свободный телефонный номер на удаленном объекте,
и скорость и качество соединения будут невысокими. Естественной альтерна
вой является цифровая связь, которая, в свою очередь, имеет альтернативн
решения. Например, для связи объектов можно использовать такие технолог
как ISDN, Frame Relay или даже ATM. Выбор конкретной технологии зави
от потребностей компании и от ее финансовых возможностей по оплате п
ключения и ежемесячной оплате услуг.

Первоначально при создании сети, некоторые решения из которой ле
в основу материала данной книги, использовалась технология Frame R

(см. главу 8), услуги которой в Санкт-Петербурге предлагали несколько компаний. Эта технология к тому времени была довольно надежной и с достаточно развитой структурой, что позволяло надеяться на работу без проблем. Но внедрение этой технологии предполагает достаточно серьезные финансовые затраты. Причем уровень затрат напрямую связан с требуемой скоростью передачи данных. Однако, в виду ценовой политики провайдера, компания была вынуждена ограничить максимальную скорость передачи 32–64 Кбит/с, что не всегда соответствовало запросам. Самое главное — это препятствовало внедрению идеи передачи голоса по сети IP. С учетом таких скоростных и ценовых ограничений, а также того факта, что ПТС (Петербургская телефонная сеть) предлагала услугу формирования прямых медных проводов в городе, было принято решение приступить к работам по постепенному переходу от технологии Frame Relay к технологии xDSL. Кстати, если говорить о затратах, то еще одним доводом в пользу xDSL может быть тот факт, что существуют модемы для физических линий, интегрирующие в себе функциональность моста и поставляемые с портом 10Base-T — идеальное решение для связи двух локальных сетей, если не требуется более гибкое управление трафиком. В последнем случае лучше остановиться на моделях с портом V.35 и подключить их к маршрутизатору (например, Cisco Systems).

В специализированной литературе технология xDSL описана достаточно подробно, поэтому ее детальное рассмотрение останется за рамками книги. Технология действительно удивительная — она позволяет при помощи одной или двух пар прямых проводов между двумя объектами добиться пропускной способности до 2 Мбит/с (уровень достигаемой скорости существенно зависит от качества медной пары).

Для организации соединения xDSL между двумя объектами (как правило, эта технология работает для двухточечной связи) оформляется заявка в местную телефонную сеть, в которой указаны адреса объектов и номера городских телефонов (описывается процедура применительно к городу Санкт-Петербург). В заявке предлагается организовать одну или две пары прямого провода между объектами. Опыт показывает, что лучше также дать дополнительную информацию, например перечислить телефонные коробки, установленные в здании (естественно только те, к которым есть доступ). Такая информация может уменьшить время проверки возможности выделения проводов. Через некоторое время приходит ответ. Если поставщик услуг согласен, то необходимо оформить договор и после его оплаты на кросс-узлах промежуточных АТС формируют прямой провод (часто под термином «прямой провод» понимают одну медную пару проводов). Естественно, описан идеальный вариант развития событий, в то время как на практике часто появляются проблемы. Если прямой возможности нет, то можно провести дооборудование объекта или, иными словами, расширить за свой счет кабельные емкости оператора связи по его указанию. Автор сталкивался с необходимостью провести дооборудование нескольких объектов — были проложены десятипарные телефонные кабели от коммуникационных шкафов, включая установку телефонных коробок. Только после этого были выделены прямые провода.

Выделенный в ваше распоряжение прямой провод необходимо проверить, так как его качество будет очень существенно влиять на надежность и скорость канала связи. Простую проверку можно провести с помощью обычного тестера. При этом с одной стороны замыкают концы у выделенного провода, а на другом измеряют суммарное сопротивление. Естественно, тестер не должен показывать «бесконечность». Если все в порядке, то стрелка должна остановиться в интервале между 100 и 1500 Ом. Опыт показывает, что в большинстве случаев, если сопротивление проводов больше 1500 Ом, модемы работать не будут. На практике есть еще масса характеристик провода, влияющих на работоспособность. К ним относятся: диаметр жил, наличие переходов между сечениями, емкостные составляющие и т. п.

Теперь можно взять пару модемов на пробу. Хорошо бы уделить проверке хотя бы два дня (вообще говоря, чем дольше, тем лучше) и за это время провести мониторинг. Следует помнить, что периодически, особенно на плохих проводах, модемы могут терять синхронизацию и канал связи «падает». После разрыва связи модемы будут пытаться восстановить соединение. Длительность таких перерывов зависит от модели модема. Иногда бывает так, что модемы восстанавливают связь, но через пару секунд теряют несущую, затем цикл повторяется вновь. Учитывая, что на большинстве моделей модемов есть визуальная индикация, то этот процесс легко наблюдать. Что касается рекомендаций о применении того или иного типа модемов, то дать ее трудно, так как не все типы модемов были доступны для тестирования и по этой причине рекомендация будет не совсем объективна. Так, хорошо показали себя на «сложной» линии модемы FlexDSL компании «Натекс» с модуляцией PAM, хотя была ситуация, когда после очередной потери синхронизации они не смогли восстановить связь без принудительного выключения питания на обоих концах. Так же очень хорошо работали модемы компании SCHMID TELECOM AG серии Watson и модемы PairGain (ADC) Megabit Modem MM300S.

Модемы могут иметь различные типы интерфейсов для подключения к ним сетевого оборудования, но чаще всего для связи локальных сетей используются интерфейсы V.35 или 10BaseT. В последнем случае модем работает либо как мост, что бывает чаще, либо как малофункциональный маршрутизатор, возможностей которого, однако, достаточно для подключения, например, к сети Интернет. Существуют модемы с портами Ethernet, которые имеют свой собственный IP-адрес для дистанционного управления ими через Telnet. Хочется особо отметить удобство использования модемов с портом 10BaseT, работающих в режиме моста. В рассматриваемой сети работает модем PairGain (ADC) Megabit Modem MM300S, который, несмотря на невысокую стоимость (порядка $1200 за две штуки), обеспечивает качественное соединение на скоростях до 2 Мбит/с.

То есть в простейшем варианте, когда нужно связать две территориально разнесенные локальные сети, модемов с портами 10Base-T будет вполне достаточно. Кроме того, как показывает опыт, даже при числе объектов более десяти модемы с функциональностью моста не создают никаких проблем, хотя при планировании сети были сомнения на случай появления широковещательного трафика с присущим ему большим объемом. Однако в действительности уровень широковещания в сети не превышал допустимых пределов. На что, в ч

ности, повлияли коммутаторы компании Hewlett-Packard, которые поддерживают технологию ABC (Automatic Broadcast Control — автоматический контроль широковещательных пакетов), позволяющую снизить уровень общего трафика.

В подобных модемах не предусмотрено никакого внешнего управления через компьютер, которое, например, доступно для модели FlexDSL, но они позволяют проводить регулировку скорости с помощью специальных переключателей на задней панели. Существуют ситуации, когда такой регулировкой приходится пользоваться. Например, в результате ухудшения качества линии (а это может произойти после дождя) модемы постоянно теряют синхронизацию. Для более устойчивой их работы можно позвонить на удаленный объект и попросить любого сотрудника с помощью отвертки установить указанную вами скорость. Более «продвинутые» модели модемов позволяют провести корректировку скорости только на одном из них — на ведущем (master), а второй удаленный модем (slave или ведомый) должен автоматически утвердить предложенную скорость или же не принять ее, если качество провода не позволяет. Как правило, управление подобными модемами осуществляется через консольный (последовательный) порт (RS-232) компьютера с помощью программы терминала. В этом месте хочу несколько отвлечься от темы и отметить, что были ситуации, когда для управления модемами компании «Натекс» пришлось своими руками монтировать кабель для подключения модема к компьютеру — все имеющиеся у нас кабели не обладали соответствующий распайкой. Впрочем, это не должно вызывать больших трудностей, так как схема нужного кабеля приводится в документации на модем.

Переход от технологии Frame Relay выполнялся постепенно. В общей сложности, процесс перехода занял более двух лет, хотя, конечно, можно было сделать это быстрее. Сложности возникли с самым удаленным объектом, который находился на расстоянии 30 километров от ближайшего объекта, к которому можно было доступиться по прямому проводу. В конечном итоге и эта удаленная точка была подключена по технологии VPN (виртуальных частных сетей). Но это стало возможным после появления провайдера услуги ADSL (асимметричная цифровая абонентская линия — Asymmetric Digital Subscriber Line, провайдер WebPlus — http://www.wplus.net). Кстати, следует отметить, что подключение по ADSL со скоростью 64 Кбит/с (входящий поток) и 16 Кбит/с (исходящий поток) с последующей шифрацией трафика в туннеле по протоколу PPTP ничуть не хуже, чем виртуальное соединение Frame Relay со скоростью подключения 64 Кбит/с (Access Rate, AR) и гарантированным CIR (Committed Information Rate) в 16 Кбит/с. Но это если не брать во внимание вопросы надежности — туннель PPTP, работающий через сеть Интернет, иногда имеет свойство отказывать 2–3 раза в день, хотя время восстановления составляет всего пару минут. Более того, иногда при управлении через терминал удаленным сервером, работающим под Windows 2000, кажется, что стало даже лучше, быстрее.

Сначала выбор был сделан в пользу модемов с портом V.35 для того, чтобы можно было задействовать имеющиеся маршрутизаторы, которые использовались с технологией Frame Relay. После того как все маршрутизаторы оказались в связке с модемами с портом V.35, мы стали приобретать модемы с портами

10Base-T. Хочу сразу отметить их недостаток — сложно отслеживать проходящий через канал объем трафика. Если используются маршрутизаторы, то достаточно подключиться через сеанс Telnet и вывести статистику по интерфейсу (для маршрутизатора Cisco Systems с помощью команды show interface ...) или запустить инструменты сбора статистики по протоколу управления SNMP. Многие администраторы предпочитают программу MRTG (http://people.ee.ethz.ch/~oetiker/webtools/mrtg), отображающую в графическом виде уровень проходящего трафика. Для модемов с портами 10Base-T это невозможно. Кроме того, при использовании модемов с функцией моста, в том случае если IP-адрес им не назначается, сложно провести мониторинг работоспособности удаленного канала связи, особенно когда на другом конце нет постоянно работающих устройств с фиксированным IP-адресом. В результате, в распределенной сети получилась некоторая «солянка» из разных типов модемов. Однако постепенно по мере реализации идеи передачи голоса поверх сети IP, модемы с портами Ethernet стали замещаться на связку из маршрутизатора (Cisco Systems) и модема с портом V.35. Основная причина замены не в том, что модемы с портами V.35 лучше, чем модемы с портами Ethernet, а в том, что передача речи очень критична к задержкам, и если по одному каналу одновременно передаются и голос и данные, то крайне необходимо голосовому трафику назначать больший приоритет при передаче. С такой задачей могут справиться маршрутизаторы, но не в силах совладать модемы. Правда и стоимость организации связи увеличивается на цену двух маршрутизаторов, хотя, как вариант, можно приобрести бывшие в употреблении устройства (рекомендую посетить сайт www.usedcisco.ru).

Переход на выделенные каналы xDSL ознаменовался для компании высокими скоростями (от 64 Кбит/с до 2 Мбит/с) и резким снижением затрат — абонентская плата за одну пару прямого провода составила примерно 500 рублей в месяц (хочу сразу отметить, что приводимые в книги денежные величины верны только на момент написания книги и, естественно, могут меняться со временем). А затраты на связь всех объектов по городу с выходом в сеть Интернет составили приблизительно 5000 рублей в месяц. Помимо экономической целесообразности отказа от услуг Frame Relay существует и скоростная составляющая — связь со скоростью менее чем 64 Кбит/с не проводится ни с одним из объектов. Сегодня руководство компании проявляет все больший интерес к внедрению местной голосовой связи с удаленными объектами с применением технологии VoIP. И очень отрадно, что существующая инфраструктура сети благодаря своим высоким скоростям позволяет провести внедрение этой технологии без больших затрат.

Для этого существует несколько решений. Например, компания Cisco Systems предлагает модели телефонов, которые подключаются напрямую к локальной сети. В этом случае необходимо провести следующие работы: приобрести шлюз, который с одной стороны включается в центральную офисную телефонную станцию головного офиса (УАТС), а с другой стороны — в локальную сеть; на каждый удаленный объект приобретается необходимое количество IP-телефонов; устанавливается сервер с работающим программным обеспечением Cisco CallManager (или аналог), выполняющий задачи телефонной станции, но только для IP-сети. После этого на удаленных объектах пользователи могут принимать внеш

и внутренние звонки и выполнять их сами. Причем высокоскоростные каналы связи в большой степени и определяют успех проекта — в зависимости от используемых кодеков один телефонный разговор может занимать разную полосу пропускания (например, 32 Кбит/с), и чем выше скорость каналов, тем больше одновременных разговоров будет поддерживаться (ну и, разумеется, тем будет выше качество голосового соединения). Если у компании нет возможности приобрести подобные модели телефонов, то можно предложить другой вариант — использовать предложенную компанией Cisco Systems модель Cisco ATA 186, которая позволяет подключать к сети IP обычные аналоговые телефонные аппараты.

Ранее говорилось о достоинствах создания каналов связи своими силами, теперь необходимо отметить и недостатки такого пути. Самый существенный из них — низкая надежность, особенно при работе на критических расстояниях. Модемы могут под воздействием внешних факторов потерять контакт друг с другом, после этого нужна как минимум пара минут для восстановления связи. Зачастую проблемы со связью возникают из-за ремонтных работ, которые проводились в телефонных колодцах. В таких случаях время простоя может достигать нескольких дней. То есть если надежность связи является строго обязательным условием, то настоятельно рекомендуется обратиться к провайдеру и заказать оптоволоконный канал для организации связи.

Могут возникнуть и вовсе мистические неполадки. Такая ситуация в течение длительного периода времени наблюдалась на одной из удаленных точек. Постоянно связь с ней была очень неустойчива — модемы могли проработать 2–3 часа, а затем потерять синхронизацию на 5–6 часов. Неоднократные вызовы кабельщиков-монтажников решали проблему только временно и перебои со связью повторялись регулярно — проблема заключалась в том, что вносились сильные помехи через проложенный рядом силовой кабель. Ко всему прочему, в один прекрасный момент охотники за цветными металлами вырезали кусок кабеля, проложенного внутри здания до телефонной коробки. Поставленный взамен новый кабель погрызли крысы и т. п. Но бывает и иначе, когда провод работает месяцами даже без намека на сбои, а если они и случаются, то очень редко, и модемы сами восстанавливают синхронизацию. Периодические проблемы со связью становятся еще более ощутимыми, если в компании планируется внедрить IP-телефонию. Как показывает практика, построение распределенной сети своими силами является оптимальным выходом только на определенном этапе ее развития. По мере того, как в сети внедряются все новые решения, которые подразумевают повышенные требования к надежности и качеству, пользователи могут все серьезнее ощущать на себе недостатки медных проводов.

Хочется отметить, что к моменту появления необходимости подключения первого территориально-удаленного объекта (который было невозможно подключить с помощью прямого провода из-за значительного расстояния) все другие точки уже были связаны между собой с помощью прямых проводов. Опыт создания виртуальной сети оказался столь успешным, что в последующем он применялся для подключения даже тех объектов, которые можно было бы просто замыкать на сеть центрального офиса. Почему? Дело в том, что всегда было желание сделать удаленные объекты максимально автономными и независимыми от перебоев со связью. Ведь прямые провода, как и любые каналы связи, перио-

дически перестают работать. И чем независимее точка, тем меньше будет пр блем с пользователями.

Так вот, успех с первым объектом подтолкнул к пересмотру схемы всей распр деленной сети — теперь объекты представляют собой некие «островки», связан ные между собой туннелями PPTP, и при этом остаются автономными. А межд островками «штормит океан» сети Интернет. Безусловно, у такого подхода ест и недостатки: необходимо устанавливать защитные экраны (или другие фильтру щие устройства), настраивать туннели, планировать и поддерживать политик удаленного доступа и т. п. К достоинствам же можно отнести уже отмеченну автономность объектов, контроль доступа из удаленного объекта в центральну сеть офиса (можно настраивать ограничения на доступ для удаленных польз вателей), контроль доступа в Интернет для каждого объекта и, как следстви отдельный учет полученного и переданного трафика (как в сети Интернет, т и между туннелями), возможность быстро и безболезненно разорвать туннел между объектами в любой момент времени и т. п. Несмотря на то, что туннел подразумевают некоторое административное вмешательство, но по сути, од раз настроив их, можно практически забыть об их существовании.

Оборудование удаленных объектов

На удаленных объектах с небольшим количеством компьютеров не всегда цел сообразно выделять место под серверную комнату. Во-первых, зачастую ме является дефицитом (скажем, ваша компания открыла магазин и для этих ц лей арендует помещение — как правило, все пространство тотчас же будет за то под упаковку и т. п.), во-вторых, на небольшое количество пользователей н нужды создавать емкую сетевую инфраструктуру.

Те специалисты, кто сталкивался с подобной задачей, согласятся, что наилу шим решением будет установка оборудования в коммуникационные шкафы, к торые имеют защиту от пыли и могут закрываться на ключ. Шкафы помога защитить оборудование от внешнего воздействия, не говоря уже и о несанкц нированном доступе. Если важность их использования очевидна, то интерес остановиться на том, какие именно шкафы нужны. По мнению автора, закуп дорогих фирменных стоек не всегда оправданна, особенно если не выдвигаю очень жесткие требования по защите оборудования от окружающей среды. В бо шинстве случаев таких требований нет, и все что нужно, это укрыть аппарату от любопытных глаз, пыли, грызунов и насекомых (как правило, от тарака и крыс, если последние присутствуют, конечно). Например, тараканы имеют вр ную привычку забираться внутрь оборудования и застревать между лопаст охлаждающих вентиляторов. Защита от пыли также важна, особенно если об территориально удален и туда нет возможности ездить каждый месяц, чтоб прибрать помещение. Одним из возможных решений может быть покупка ме лического шкафа, или антресоли, с шириной и высотой от 1000 мм, а глуби от 400 мм. По крайней мере, в Санкт-Петербурге были найдены именно шкафы, которые хорошо подходят для размещения двух серверов, одного М

тора и другого небольшого по размеру сетевого оборудования. В этот шкаф обязательно нужно поставить вентилятор (вытяжной), питаемый от обычной сети 220 В. В противном случае температура внутри шкафа может достигнуть критической для оборудования отметки. Щели в шкафу желательно залить герметиком, а на дверцу навесить замок. Подобное решение обойдется гораздо дешевле, чем покупать фирменный коммуникационный шкаф, который иногда по стоимости сопоставим со стоимостью сервера. Обратите внимание на то, что если шкаф будет стоять рядом с местом «дислокации» местных работников, то они его обязательно приспособят под чайный столик и несколько раз обольют кофе. Поэтому желательно шкаф размещать так, чтобы кофе на нем пить было дискомфортно, пусть даже в ущерб собственным удобствам. При нормально работающем оборудовании профилактику достаточно делать 1–2 раза в год, приезжая с пылесосом и мокрой тряпкой. Для отвода кабелей (силовых, компьютерной сети и т. п.) в шкафу можно сделать несколько отверстий и установить специализированные вводы кабелей с защитой от пыли. Также неплохо иногда проверять температуру внутри сервера (многие серверные материнские платы позволяют делать это удаленно), чтобы «отловить» момент, когда вентилятор шкафа перестает работать.

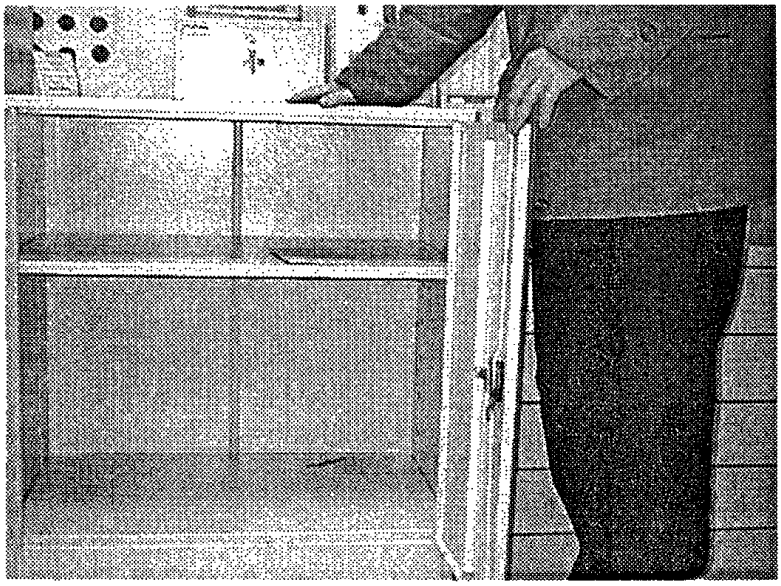


Рис. 10.1. Пример ящика под оборудование

Обратите внимание, что размер ящика под оборудование зависит от планируемого количества устройств, которые будут там установлены. Например, если применяется шкаф, показанный на рис. 10.1, то на нижнюю полку можно поставить серверы (так получалось разместить до двух небольших серверов), а на верхнюю — небольшие источники бесперебойного питания и другое некрупногабаритное сетевое оборудование (концентраторы и т. п.). В этом варианте компоновки предполагается, что монитор будет стоять на самом ящике. Если количество серверов в ящике будут больше 1, то рекомендую приобрести переключатель «монитор-клавиатура-мышь» на 2 или 4 порта. Это позволит управлять серверами, не

открывая сам ящик — просто переключаясь на нужный порт. Как правило, если принято решение размещать оборудование в подобной мебели, то предполагается, что оно будет обслуживаться довольно редко. Тогда, как уже было отмечено, необходимо монтировать выдувной вентилятор (а лучше 2 — были случаи, когда вентилятор выходил из строя и температура в ящике достигала критичного максимума) и накрывать монитор полиэтиленовым мешком (обычно новые мониторы поставляются уже в таких упаковочных мешках). Это некоторой степени будет способствовать борьбе с пылью. Отмечу, что не всегда размещение монитора на самом ящике будет удачным решением — на практике автор сталкивался с ситуациями, когда не следовало привлекать внимание к факту, что на объекте есть иное компьютерное оборудование, за исключением офисных ПК. В таком случае можно приобрести аналогичный ящик просто больших размеров.

Технологии виртуальных частных сетей

Автор не ставил задачей в полном объеме осветить все теоретические вопросы организации виртуальных частных сетей, для этого вам достаточно просмотреть архив соответствующих публикаций по сетевой тематике. Самое интересное, на мой взгляд, это «спуститься» на сетевой уровень и взглянуть на примеры реальных перехватов трафика. Тогда такие вопросы, что же на самом деле представляет собой туннель, не будут вызвать недопонимания — многие нюансы станут полностью ясны. Отмечу, что в этой главе рассматривается только протокол PPTP, хотя в последнее время все больше говорят об IPSec. Последний считается более надежным и современным, хотя на взгляд автора этот протокол гораздо более сложен в реализации, чем протокол PPTP.

Технически реализация защищенных виртуальных сетей стала возможной уже достаточно давно. Инкапсуляция использовалась раньше и применяется сейчас для передачи немаршрутизируемого трафика через маршрутизируемые сети, а также для ограничения многопротокольного трафика одним протоколом. Технологии шифрования также появились задолго до широкого внедрения глобальных сетевых технологий. Однако общепринятые протоколы для создания защищенных виртуальных сетей разработаны недавно и сейчас продолжается работа над их совершенствованием и расширением. Они являются открытыми или, иными словами, свободными для распространения и реализации.

Для независимости от прикладных протоколов и приложений защищенные виртуальные сети формируются на одном из более низких (по сравнению с прикладным) уровней модели OSI — канальном, сетевом или сеансовом. Канальному (второму) уровню соответствуют такие протоколы, как PPTP (о нем, собственно, и пойдет речь в этой главе), L2F (Layer-2 Forwarding — предлагался компанией Cisco Systems и рядом других производителей) и L2TP (Layer-2 Tunneling Protocol — протокол туннелирования 2-го уровня; был предложен как компромиссное решение между двумя первыми предложениями); сетевому (третьему) уровню — IPSec (IP Security — безопасный протокол IP; в последнее время получает все большую популярность в связи с увеличением количества обо

дования и программного обеспечения, его поддерживающего), а сеансовому (пятому) уровню — SSL/TLS и SOCKS.

Чем ниже уровень эталонной модели, на котором реализуется защита, тем она прозрачнее для приложений и незаметнее для пользователей. Однако при снижении этого уровня уменьшается набор реализуемых услуг безопасности и становится сложнее организация управления. Чем выше защитный уровень в соответствии с моделью OSI, тем шире набор услуг безопасности, надежнее контроль доступа и проще конфигурирование системы защиты. Однако в этом случае усиливается зависимость от используемых протоколов обмена и приложений.

Для стандартного формирования криптозащищенных туннелей на канальном уровне модели OSI компанией Microsoft при поддержке компаний Ascend Communications, 3Com/Primary Access, ECI-Telematics и US Robotics был разработан протокол туннелирования PPTP, представляющий собой расширение протокола PPP (Point-to-Point Protocol — протокол связи «точка-точка»). В протоколе PPTP не специфицируются конкретные методы аутентификации и шифрования передаваемых данных. Клиенты удаленного доступа в операционных системах Windows NT 4.0 и Windows 98 (с компонентом Dial-Up Networking) поставляются с версией шифрования DES компании RSA Data Security, получившей название «шифрование Microsoft для канала "точка-точка"» (Microsoft Point-to-Point Encryption — MPPE).

Канальному (второму) уровню модели OSI соответствует также протокол туннелирования L2F (Layer-2 Forwarding), разработанный компанией Cisco Systems при поддержке компаний Shiva и Northern Telecom. В данном протоколе также не специфицируются конкретные методы аутентификации и шифрования. В отличие от протокола PPTP протокол L2F позволяет использовать для удаленного доступа к провайдеру не только протокол PPP, но и другие протоколы, например SLIP (на настоящее время устарел и практически не встречается). При формировании защищенных каналов по глобальной сети поставщикам услуг Интернета не нужно выполнять конфигурирование адресов и аутентификацию. Кроме того, для переноса данных через защищенный туннель могут использоваться различные протоколы сетевого уровня, а не только IP, как в протоколе PPTP.

Протоколы PPTP и L2F были представлены в организацию Internet Engineering Task Force (IETF) и в 1996 году соответствующие комитеты решили их объединить. Получившийся в результате протокол, вобравший в себя все лучшее из PPTP и L2F, был назван протоколом туннелирования второго уровня (Layer-2 Tunneling Protocol — L2TP). Его поддерживают компании Cisco Systems, Microsoft, 3Com, Ascend и многие другие производители оборудования и программ. Как и предшествующие протоколы канального уровня, спецификация L2TP не описывает методы аутентификации и шифрования. Протокол L2TP является расширением PPP на канальном уровне и может поддерживать любые высокоуровневые протоколы.

Спецификацией, где описаны стандартные методы для всех компонентов и функций защищенных виртуальных сетей, является протокол Internet Protocol Security (IPSec), соответствующий сетевому уровню модели OSI и входящий в состав новой версии протокола IP — IPv6. Протокол IPSec иногда еще называют

протоколом туннелирования третьего уровня (Layer-3 Tunneling). IPSec предусматривает стандартные методы аутентификации пользователей или компьютеров при инициализации туннеля, стандартные способы шифрования конечными точками туннеля, формирования и проверки цифровой подписи, а также типовые методы обмена и управления криптографическими ключами между конечными точками. Этот гибкий стандарт предлагает несколько способов для выполнения каждой задачи. Выбранные средства для достижения одной цели обычно не зависят от методов реализации других задач. Для функций аутентификации IPSec поддерживает цифровые сертификаты популярного стандарта X.509.

Практика подключения объекта

Очевидно, что первым этапом создания виртуальной частной сети является подключение объекта (или нескольких объектов) к сети Интернет. При этом вариантов подключения имеется множество. Остановимся на одном из решений, которое предлагает при небольших затратах обеспечить относительно высокие скорости — технологии ADSL. Само подключение по технологии ADSL осуществляется с помощью провайдера услуг сети Интернет, который имеет оборудование на ближайшей к объекту АТС. Физически подключение сводится к двум шагам (рассмотрим на практическом примере подключения одного из объектов) — формированию прямого провода от объекта до кросс-узла указанной провайдером АТС и последующей проверке этого провода на качество с помощью тестера. Следует отметить, что технология ADSL может работать и по медной паре, к которой подключаются обычные городские телефоны (кстати, именно такое решение наиболее распространено). Отдельное формирование прямого провода для работы технологии ADSL может быть оправдано, если, например, ваш телефон параллельно используется для сигнализации во вневедомственную охрану или установлено оборудование уплотнения.

Здесь специалисты провайдера гарантировали, что их оборудование ADSL компании Siemens будет вполне стабильно работать при величине сопротивления провода до одного килоома. После этого приехал специалист от провайдера и поставил модем ADSL. Модем имел два порта — один порт RJ12 для подключения в линию, а второй, RJ45 — порт Ethernet для соединения с локальной сетью. Этот порт напрямую включался во внешний интерфейс защитного экрана с помощью так называемого кросс-кабеля, причем этот кабель был предоставлен провайдером. В нашем случае модем выполнял еще и роль маршрутизатора по умолчанию, на который должен передаваться весь трафик. Поэтому ему был присвоен IP-адрес. Причем адрес назначался провайдером и настраивался на модеме специалистами его службы технической поддержки. Скорость передачи по каналу ADSL составляла 64 Кбит/с в локальную сеть (направление «к клиенту») и 16 Кбит/с в сеть Интернет (направление «от клиента»). Как видим, асинхронность этой технологии очевидна — пользователь получает информацию из сети Интернет в большем объеме, чем передает, и по этой причине скорость к нему больше, чем от него. Это правило очевидно, когда некая локальная сеть просто подключается к Интернету и не подразумевается наличие туннеля с

гой удаленной сетью. В таком варианте характер распределения трафика может быть кардинально другим — все зависит от используемых приложений.

Естественно, что как только было выполнено подключение объекта к сети Интернет, сразу появилась необходимость в установке защитного экрана (в принципе, можно обойтись и без защитного экрана, если, например, на удаленном объекте работает только один компьютер). Причем важно отметить, что защитный экран должен иметь либо встроенную поддержку организации сетей VPN, либо просто иметь возможность поддерживать необходимые протоколы. В нашем случае для организации VPN используется протокол PPTP компании Microsoft. В качестве защитного экрана тестировался продукт Kerio WinRoute Pro (более подробную информацию о котором можно получить по адресу www.kerio.com), который без проблем поддерживает протокол PPTP. Но на момент подготовки материала главы было одно ограничение — нельзя было создавать туннель с того же сервера, на котором работал продукт WinRoute Pro, из-за особенностей реализации последнего. Если же в сети присутствует другой защитный экран, который обеспечивает создание туннеля прямо с него (то есть он должен быть конечной точкой туннеля), то такое решение является еще более привлекательным. В рассматриваемом примере полагаем, что сеть защищается отдельным защитным экраном (это, разумеется, может быть как программное решение, так и специализированное устройство, например Cisco PIX Firewall), а для организации виртуальной частной сети используется сервер VPN, находящийся внутри защищаемой сети (рис. 10.2).

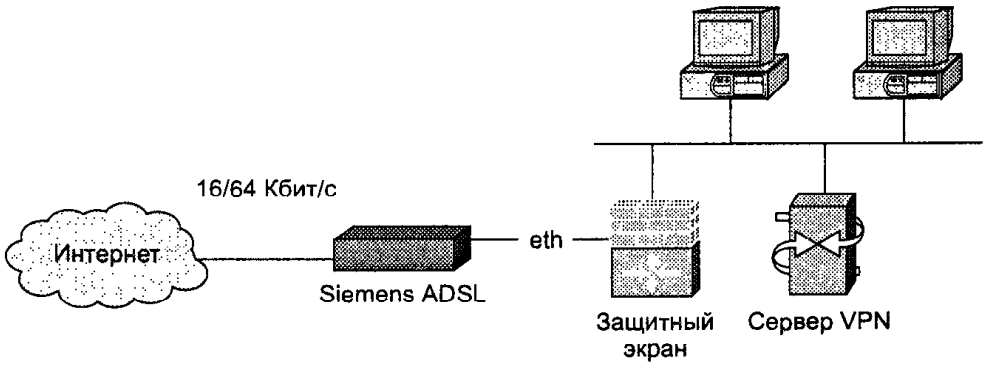


Рис. 10.2. Пример подключения объекта к сети Интернет

Существующие методы и подходы по защите сетей от несанкционированного доступа подробно описаны в большом количестве источников. По этой причине подробно на общих подходах останавливаться нет необходимости. Следует отметить только, что, на взгляд автора, самый «навороченный» защитный экран не обязательно будет работать лучше, чем простое и дешевое решение. Кроме того, опыт проведения работ по защите сети показал, что чем понятнее внутренняя логика работы защитного экрана, тем, в конечном счете, защита оказывается крепче. В конце концов, настройку экрана выполняет специалист, который должен досконально разбираться в вопросах его функционирования. Автор пробовал работать с программными продуктами NetGuard Guardian Pro 5.0x, Microsoft ISA Server и Kerio WinRoute Pro. Под термином «работать» имеется в виду непо-

средственная эксплуатация этих продуктов в сети в течение определенного времени. Несмотря на то что все протестированные продукты функционировали великолепно и нареканий не вызывали, больше всего симпатий вызвал продукт Kerio WinRoute Pro (на момент написания главы предлагалась версия 4.2.2, которая, кстати, поставляется на различных языках, включая русский). Такое отношение проистекает из-за прозрачной и понятной логики его работы. Сам продукт умещается на дискету, интуитивно понятен в использовании и очень производителен. И именно на этом продукте я и остановился. Естественно, есть у него и определенные недостатки. На взгляд автора, самым серьезным можно считать отсутствие удобных средств для анализа журналов работы пользователей. Иногда наличие таких средств просто необходимо. Например, если в компании проводится политика, согласно которой за потребление ресурсов сети Интернет раздельно платят разные отделы компании в соответствующей пропорции. Такая политика начинает работать, когда объем потребляемой информации превышает какой-то минимум, а в компании есть несколько отделов, в разной степени пользующихся ресурсами сети Интернет.

Продолжая разговор о продукте Kerio WinRoute Pro, хочется отметить и другую проблему, связанную с построением виртуальных сетей на базе операционной системы Microsoft Windows 2000. Учитывая, что защитный экран размещается на границе сети, вполне понятно желание на его основе построить и туннели VPN с удаленными объектами, а точнее с аналогичными защитными экранами. В случае Microsoft ISA Server это возможно и более того, есть мастер (wizard), который поэтапно и с подсказками поможет это сделать. Но при использовании продукта Kerio WinRoute Pro в качестве защитного экрана такое решение не представляется невозможным, по крайней мере, в обсуждаемой версии — 4.2.2.

В версии 4.2.4 появилась поддержка транзитного трафика IPSec. Этот факт прояснился после отправки запроса в службу технической поддержки. Как оказалось, существует конфликт между службой RRAS и Kerio WinRoute Pro (точнее с возможностью dial-on-demand службы RRAS). По этой причине и было предложено устанавливать серверы VPN за защитным экраном. Последнее решение было протестировано и доказало свою полную работоспособность. К слову, техническая поддержка пользователей вызывает только восхищение — по меньшей мере ни один из моих вопросов, отправленных в службу технической поддержки, не остался без внимания. Так, например, по ходу тестирования этого защитного экрана было выявлено его странное поведение в случае, если объем проходящих через него небольших дейтаграмм протокола UDP достигал некоторого порогового значения. Это, кстати, вполне нормальная ситуация для большинства компьютерных игр, предлагаемых на серверах в сети Интернет (в частности Counter-Strike). Так вот, после отправки запроса в службу технической поддержки с описанием проблемы ответ пришел на следующий день с четким указанием как и что нужно делать.

Одним из возможных вариантов расширения функциональности защитного экрана может быть установка на сервере программного продукта Microsoft ISA Server, который настроен только на выполнения кэширования запросов к сайтам (один из вариантов установки — в другом режиме он будет сам работать как защитный экран). Этот компонент помогает решить две задачи — контроль до-

ступа и ускорение доступа к сайтам Интернета за счет хранения последних загруженных страниц в памяти и дисковом кэше.

Контроль доступа достигается следующим образом. Сервер, выполняющий роль защитного экрана, включается в домен (хотя это и может противоречить основным рекомендациям по организации защитных экранов), а затем при настройке сервера Microsoft ISA Server указывается, что запросы могут проходить только от определенной группы пользователей домена (например, домена Windows 2000 Active Directory). Далее, пользователи указывают в настройках своих браузеров (например, Internet Explorer), что следует работать через прокси-сервер (proxy server, часто также встречается термин сервер-посредник), и указывают номер порта (по умолчанию для сервера Microsoft ISA Server номер порта — 8080). Впоследствии все запросы на просмотр страниц будут перенаправляться прокси-серверу, который либо вернет данные из кэша, либо самостоятельно получит их из сети Интернет. Для пользователей подобная схема работы абсолютно прозрачна. И после всего на защитном экране настраиваются правила фильтрации, в соответствии с которыми запрос протокола TCP на порт назначения с номером 80 (протокол HTTP) могут проходить только от прокси-сервера, который работает на самом защитном экране или на отдельно стоящем сервере.

Прозрачность для пользователей достигается тем, что при отправке запроса на доступ к определенной HTML-странице на выбранном сайте браузер в ответ на запрос получает от настроенного на нем прокси-сервера информацию о том, что требуется аутентификация. После того как прокси-сервер проверил права пользователя и убедился в том, что последнему позволено воспользоваться его услугами, он начинает работать как бы за него, отправляя приходящие от клиента запросы HTTP в сеть Интернет. Безусловно, подобная организация доступа в сеть Интернет позволит контролировать доступ только для протоколов HTTP и FTP, которые являются наиболее популярными у пользователей. Для других протоколов в рассматриваемой схеме контроль невозможен, и если он необходим, то администратору придется ориентироваться исключительно на адреса IP. Основным преимуществом программного продукта Microsoft ISA Server является возможность проверять принадлежность пользователя к определенной группе перед тем, как отправить его запрос в сеть Интернет. Причем параллельно ведется полный журнал работы, что позволит далее выполнять аналитический разбор трафика.

Кроме установки перечисленных программных продуктов на защитный экран, можно порекомендовать обратить внимание на средства, контролирующие задачи, которые пользователи выполняют, работая в сети Интернет. В частности, довольно полезной является возможность проводить классификацию сайтов, посещаемых пользователем, с последующей генерацией отчетов посещений. Автор тестировал продукт компании SurfControl (http://www.suRFControl.com), который интегрируется с Microsoft ISA Server. Компания SurfControl выпускает несколько версий своего продукта для работы с разными защитными экранами (в частности есть версия для CheckPoint Firewall-1). В общем случае схема настройки защитного экрана очень проста — вслед за установкой операционной системы Windows 2000 Server устанавливается продукт ISA Server, а затем SuperScoute for ISA Server. Далее все запросы пользователей, проходящие через прокси-сервер,

минуют так называемый подключаемый модуль (plug-in) системы SuperScout, и автоматически причисляются к той или иной категории. Для этого ведется специальный файл (формат Microsoft Access), на основе которого затем можно получать очень удобные отчеты. Кроме того, к несомненным достоинствам данного продукта следует отнести возможность создания групп пользователей с дальнейшей генерацией отчетов по созданным группам. Например, руководство для минимизации затрат имеет право поинтересоваться, сколько времени провели в Интернете отдельные пользователи и какой объем трафика (за который, как правило, приходиться платить) был получен из сети Интернет разными отделами. Кроме того, довольно интересна возможность классифицировать посещаемые пользователями сайты по разным категориям, например, таким как «сайты для взрослых», «развлечения», «спорт» и т. п. Дополнительно с продуктом поставляется большая база уже классифицированных сайтов, доступная для обновления, а также можно воспользоваться другим продуктом, который самостоятельно пытается приписать неизвестные сайты к той или иной категории (на основе анализа набора ключевых слов в ответах протокола HTTP — правда, с русским языком могут быть проблемы).

Шаги настройки защитного экрана, если были выбраны описанные продукты, следующие. Сначала на компьютер под управлением Windows 2000 Server устанавливается Microsoft ISA Server 2000. При его установке указывается режим «только кэшировать» (cache-only — в этом режиме отключается функциональность защитного экрана программного продукта Microsoft ISA Server 2000, и он будет работать только как сервер-посредник для запросов протоколов HTTP и FTP). Это важно, так как сам продукт имеет функциональность защитного экрана и его можно использовать, если администратор не выбрал другой продукт. В противном случае, во избежание проблем несовместимости между двумя защитными экранами, следует устанавливать продукт с функциональностью сервера-посредника. Далее, если необходимо, можно установить программное обеспечение учета работы пользователей в сети Интернет, а затем прикрыть все сверху защитным экраном (скажем, Kerio WinRoute Pro). Отметим, что подобная схема установки была проверена только для защитного экрана Kerio WinRoute Pro и при использовании других решений она может претерпеть изменения.

Если для построения виртуальной частной сети был сделан выбор в пользу решения PPTP от компании Microsoft, то на удаленном объекте может потребоваться установить сервер, который будет выполнять задачи шифрации и дешифрации трафика (далее по тексту он будет называться сервером VPN). Опыт показал, что сервер в состоянии выполнять более широкие функции, чем просто обслуживание туннеля PPTP. Так, он может быть контроллером домена Active Directory, сервером почты Microsoft Exchange 2000, сервером баз данных Microsoft SQL, а также поддерживать другие, менее ресурсоемкие службы, такие как DNS, DHCP и т. п. Учитывая, что количество пользователей на удаленном объекте, как правило, невелико (обычно порядка 10–15), то один сервер вполне справится с указанной нагрузкой. Можно посоветовать в этой ситуации устанавливать память в компьютер-сервер 512 Мбайт и больше, так как при решении на сервере такого количества задач резко повышаются требования к памяти, в то время как процессор занят незначительно: на 7–15%. Кроме перечисленных задач сервер мож

взять на себя функции сервера VPN, и для этих целей на нем должна быть запущена служба RRAS (Routing and Remote Access — маршрутизация и удаленный доступ), входящая в состав операционной системы Windows 2000 Server. Служба позволяет выполнять множество важных задач, таких как маршрутизация трафика между интерфейсами, поддержка протоколов динамической маршрутизации (например, RIP), поддержка организации виртуальных частных сетей и т. п.

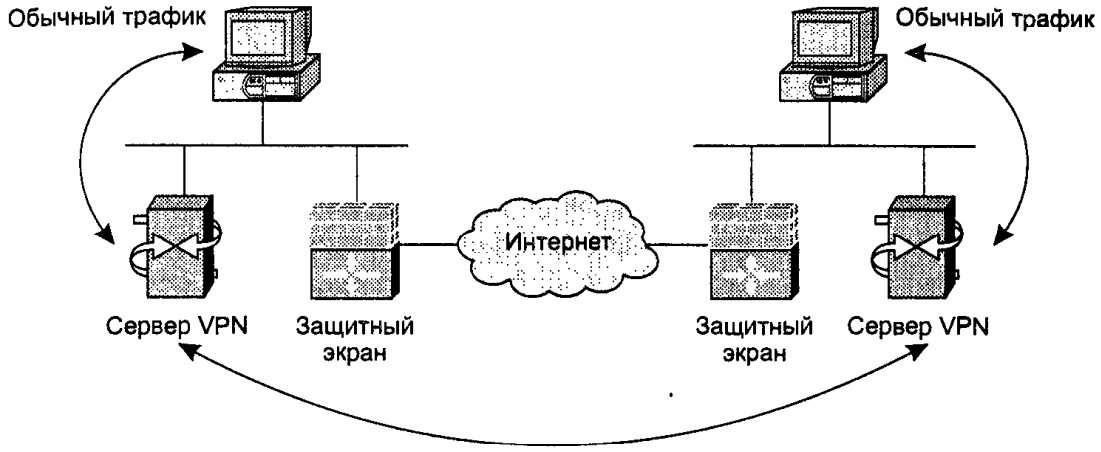


Рис. 10.3. Туннель между двумя объектами (трафик шифруется)

На рис. 10.3 показан пример связи двух объектов через туннель, «проложенный» по сети Интернет. Это пример типовой, и само собой, в конкретных реализациях ожидаемы изменения в приведенной схеме. Видно, что на каждом из объектов присутствует защитный экран (программное или аппаратное решение) и сервер VPN. Как уже было отмечено ранее, защитный экран должен поддерживать протокол PPTP — в рассматриваемом случае он просто обеспечивает прохождение IP-дейтаграмм, содержащих зашифрованные данные (или, иными словами, дейтаграмм, передаваемых по туннелю PPTP). Непосредственно процедуру шифрования и дешифрования данных выполняют серверы VPN, и для прочих пользователей в сетях это происходит абсолютно прозрачно. В идеале, после настройки туннеля эти две сети можно рассматривать как неразрывную сущность, части которой просто находятся на расстоянии двух переходов (hop) друг от друга (как будет показано далее, серверы VPN будут здесь играть роль маршрутизаторов с виртуальными интерфейсами PPTP). Для объектов на концах соединения вся промежуточная сеть Интернет будет как бы скрыта.

Основные принципы работы протокола PPTP

Виртуальная частная сеть, в основе которой лежит протокол PPTP, представляет собой простой канал связи между двумя серверами, которые просто выполняют шифрование данных при передаче их в туннель и дешифрование при получении данных из туннеля и передаче их в локальную сеть. Процедура настройки туннеля между двумя серверами не вызывает сложностей и требует выполнения

3–4 шагов. Проще может быть только настройка удаленного клиента на подключение к серверу через туннель. Для того чтобы два сервера могли установить между собой туннель, необходимо сначала настроить и запустить на них службу RRAS. При первом запуске этой службы предлагается воспользоваться услугами мастера, но как показала практика, настраивать лучше все самостоятельно, как будет показано ниже.

Сам по себе туннель PPTP является двусторонним — то есть каждый сервер самостоятельно устанавливает соединение с удаленным партнером, и только когда оба сервера выполнили эту процедуру, туннель считается рабочим. При настройке туннеля на каждом сервере необходимо указать имя и пароль пользователя, которые затем проверяются на другом конце соединения. Учитывая то, что все серверы могут входить в один домен, следует рекомендовать создать отдельную учетную запись в домене для каждого из них, которая затем и будет использоваться обоими серверами. Такой подход позволит легко управлять отдельными туннелями и получать статистику их работы — служба RRAS способна вести учет (журнал) работы указанного удаленным сервером пользователя при установлении туннеля. Пароль желательно выбрать сложно предсказуемым (например, генератором паролей), а помнить его не обязательно — единожды настроенные параметры соединения на интерфейсе DOD (Dial-on-Demand — «дозвон» по требованию) запоминаются сервером и используются автоматически в случае восстановления туннеля после проявления проблем со связью. Важным компонентом туннеля является виртуальный интерфейс DOD, который администратор должен настроить на двух серверах-партнерах по туннелю (учитывая то, что интерфейс DOD может обслуживать не только туннель виртуальной частной сети, а также, например, интерфейс с подключенным аналоговым модемом для автоматического установления соединения, далее в главе будет употребляться термин «DOD VPN», подчеркивающий то, что речь идет именно о туннеле).

В принципе, настройка интерфейса DOD VPN и подразумевает под собой настройку туннеля. Впоследствии DOD VPN будет фигурировать на сервере (в частности, в консоли службы RRAS) как обычный интерфейс, который может участвовать в передаче и получении данных, а также в маршрутизации. Естественно, таких интерфейсов на сервере VPN может быть более одного, в зависимости от того, сколько туннелей сервер поддерживает. Например, на рис. 10.4 показан пример консоли службы RRAS, на которой настроено 10 интерфейсов DOD VPN, каждый из которых обслуживает свой туннель. Процедура настройки будет в деталях показана далее в главе, а пока отметим, что для удобства администрирования есть возможность гибко назначать имена интерфейсам. В приведенном на рисунке примере каждое имя (L138, SH4 и т. п.) соответствует выбранной схеме именования удаленных объектов (например, по адресу в городе или по кругу задач).

После того как были созданы интерфейсы DOD VPN на серверах-партнерах по туннелю, а между последними установлено соединение, их можно рассматривать как обычные маршрутизаторы, несмотря на то, что физически каждый по-прежнему будет иметь один сетевой интерфейс (в принципе, сервер может быть оснащен и несколькими физическими сетевыми интерфейсами). Один

огромное удобство организации туннеля средствами Microsoft RRAS как раз в том, что он позволяет маршрутизировать трафик между физическими интерфейсами и интерфейсами DOD VPN, которые в нашем случае являются логическими (виртуальными) интерфейсами.

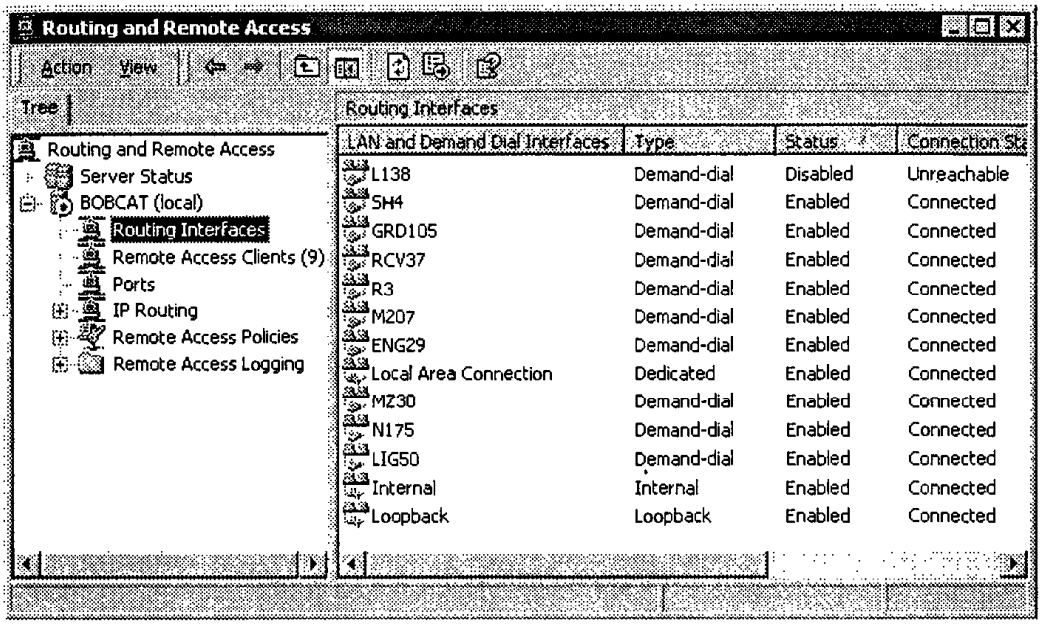


Рис. 10.4. Пример настроенных интерфейсов DOD VPN

Протокол PPTP для своей работы создает логическое соединение протокола TCP между двумя серверами-партнерами (или клиентом и сервером, если, например, в сеть пытается получить доступ мобильный клиент с переносным компьютером), участвующими в процессе создания туннеля. Первоначально в этом соединении номер порта получателя протокола TCP устанавливался в значение 5678, но позже он был изменен на 1723. Этот номер и используется в настоящее время (официально зарегистрирован в IANA). В терминологии протокола это соединение называется управляющим (Control Connection). Однако это только «рельсы» туннеля, отвечающие за служебные функции, в то время как для передачи самих данных используются протоколы PPP и GRE (Generic Routing Encapsulation — общая инкапсуляция при маршрутизации). Само по себе управляющее соединение поддерживается с помощью сообщений keepalive («пока живой»), посылаемых серверами, и такой подход позволяет быстро отслеживать перебои в связи. Туннель (то есть дейтаграммы, передающие предварительно зашифрованные данные) и управляющее соединение неотъемлемы друг от друга и совокупно формируют то, что называется туннелем PPTP между двумя объектами (или серверами на объектах).

Для лучшего понимания протокола GRE (документы RFC 1701 и RFC 1702) и его роли в создании и поддержании работы туннеля полезно рассмотреть структуру пакета этого протокола. (Тут может возникнуть некая путаница, так как название GRE не несет в себе термин «протокол», но указанные стандарты

определяют помимо всего прочего заголовок пакета и метод инкапсуляции данных в него. То есть можно с уверенностью говорить «протокол GRE».) Сам протокол был разработан как удобный способ решения общих задач инкапсуляции данных, посылаемых через сеть IP (то есть сфера его применения не ограничена только туннелем PPTP). После того как управляющий сеанс протокола PPTP был установлен (он, как будет показано далее, организуется в первую очередь при установлении туннеля), протокол GRE используется при инкапсуляции данных для последующего шифрования. Формат пакета протокола GRE показан на рис. 10.5.

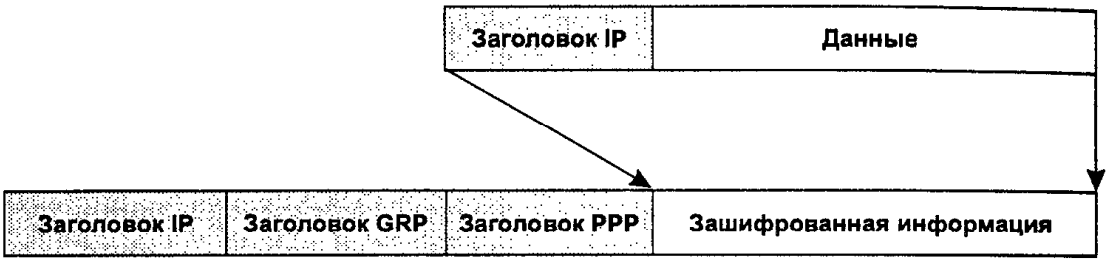


Рис. 10.5. Формат инкапсуляции дейтаграмм IP

Данные, которые должны пройти через туннель (исходная дейтаграмма, отправляемая через интерфейс DOD VPN) снабжаются заголовком протокола PPP (напомним, что протокол PPTP — это расширение протокола PPP, и поэтому заголовок этого протокола добавляется к результирующей дейтаграмме), а затем помещаются внутрь пакета GRE, который потом переносится между двумя точками виртуальной сети (или между двумя серверами) внутри новой дейтаграммы IP. После того как пакет GRE достиг противоположной точки виртуальной сети, его заголовок удаляется и упакованная дейтаграмма отправляется ее изначальному получателю. Фактически, сервер VPN, получая исходную дейтаграмму, создает еще одну — новую, в которую инкапсулируется исходная. Протокол GRE не имеет возможности устанавливать сеансы и обеспечивать защиту данных от злоумышленников. Для этого эксплуатируется способность PPTP создавать управляющее соединение. Применение GRE в качестве метода инкапсуляции ограничивает поле действия PPTP только сетями IP.

Заголовок протокола GRE содержит информацию о подтверждениях и номерах в последовательности, предназначенную для того, чтобы обеспечить некий уровень контроля перегрузок и обнаружения ошибок в туннеле. Кроме того, управляющее соединение используется для определения скорости отправки и параметров буферизации для контроля потока пакетов PPP определенного сеанса. Сам протокол PPTP не предлагает конкретных алгоритмов для контроля перегрузок и протокола, но в стандартах можно найти рекомендованные методы определения адаптивных тайм-аутов для восстановления потерянных данных.

Рисунок 10.5 показывает общую, теоретическую схему инкапсуляции данных, но для понимания деталей работы туннеля необходимо рассмотреть происходящие процессы на сетевом уровне с помощью иллюстрации перехваченных в сети пакетов. На сетевом уровне новая дейтаграмма, передаваемая по туннелю (и несущ

в себе предварительно зашифрованную оригинальную дейтаграмму), выглядит, как показано на рис. 10.6. Видно, что в нее инкапсулирован пакет протокола GRE, а в него вложен пакет протокола PPP, за которым следуют зашифрованные данные (в рассматриваемом примере — 165 байт, а размер всего кадра Ethernet составляет 216 байт).

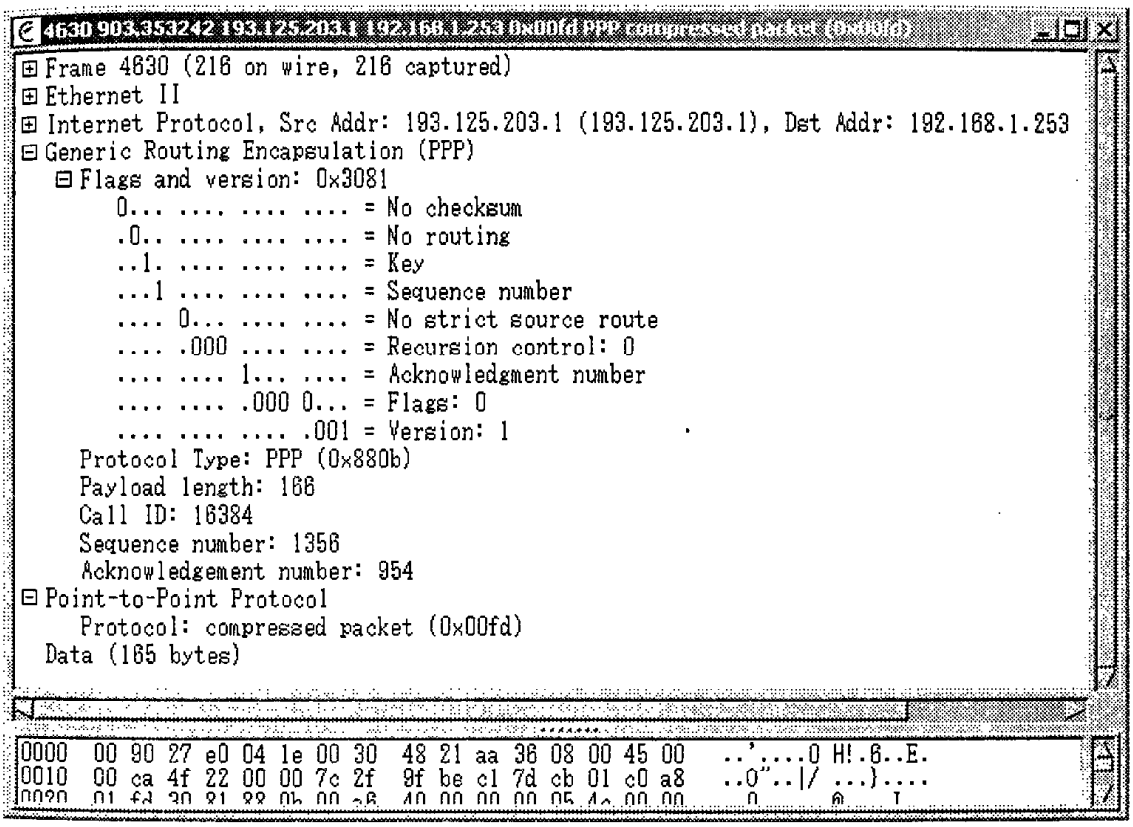


Рис. 10.6. Формат дейтаграммы с зашифрованными данными

В реализации протокола PPTP используется слегка измененная версия протокола GRE, которая была изначально описана в документах RFC 1701 и RFC 1702. Не останавливаясь на подробностях базовой реализации этого протокола, отмечу, что новшеством является поле Acknowledgement Number (номер подтверждения), которое используется для подтверждения, что определенный пакет (или группа пакетов) была успешно доставлена удаленному серверу по туннелю. Помимо этого поля, заголовок GRE включает в себя еще несколько полей, некоторые из которых не изменяются, а другие могут отличаться от пакета к пакету. Так, есть поле Call ID (идентификатор сеанса) которое позволяет отличать пакеты, принадлежащие определенному туннелю (или в терминологии стандарта протокола PPTP — сеанса) — ведь на одном сервере VPN могут поддерживаться множество туннелей с удаленными серверами. Поле Protocol Type (тип протокола) всегда установлено в значение 0x880b, что соответствует инкапсулированному протоколу PPP. Сам вложенный пакет протокола PPP содержит только два поля — Protocol (указывает на содержимое поля данных) и непосредственно

поле Data — данные. Последнее поле и объемлет предварительно зашифрова ную исходную дейтаграмму.

Для того чтобы показать в деталях происходящее во мраке туннеля, рассмот рим следующую схему связи двух объектов (рис. 10.7). На каждом из объекто есть защитный экран, реализованный на базе сервера, работающего под упра лением Microsoft Windows 2000 и программного продукта Kerio WinRoute Fi Серверы имеют по две сетевые карты, одна из которых включается во внутренню сеть, а другая в сегмент, подключенный напрямую к сети Интернет. В кажд сети также присутствует сервер VPN, который будут заниматься шифрование и дешифрованием передаваемых данных. Далее, в каждой сети используют адреса из диапазонов 192.168.3.0/24 (объект 1) и 192.168.2.0/24 (объект 2). А реса могут назначаться администратором или выделяться по протоколу DHC В последнем случае серверы VPN способны выполнять и эту задачу, впроче как и многие другие задачи (быть контроллером домена Active Directory, по товым или SQL-сервером и т. п.). Опыт показал, что никаких ограничений функциональность нет, за исключением повышенных требований к аппаратн му обеспечению. Сама процедура шифрования трафика занимает относитель низкий процент процессорного времени (4–5%).

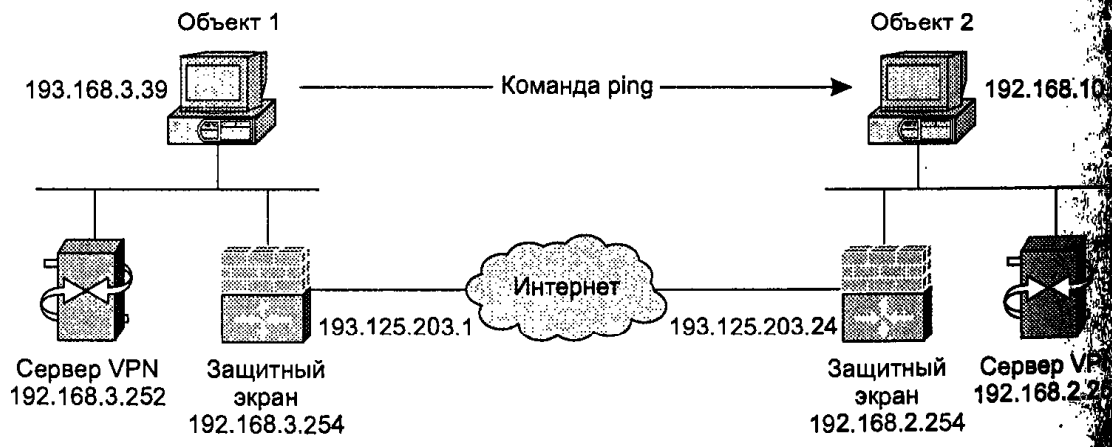


Рис. 10.7. Пример построения сети VPN

Предположим, что каждый из клиентов в сети настроен на использова шлюза по умолчанию (Default gateway), который соответствует внутренне интерфейсу защитного экрана (то есть интерфейсу в локальную сеть). То же сам относится и к серверам VPN — они, как правило, будут иметь некие статическ адреса протокола IP, но на них также должны быть настроены адреса шлюза умолчанию. На данном этапе будем считать, что туннель между двумя серве ми VPN установлен и работает. Для демонстрации происходящего на сетев уровне предположим, что некий хост на объекте 1 (адрес 192.168.3.39) выполня команду ping с адресом хоста на объекте 2 (команда ping 192.168.10.1, рис. 10. Учитывая то, что получатель дейтаграммы, содержащей запрос ICMP echo requ (протоколу ICMP посвящена глава 4), находится в сети, отличной от сети отп вителя, дейтаграмма отправляется по адресу шлюза по умолчанию — 192.168.3. (внутренний интерфейс защитного экрана).

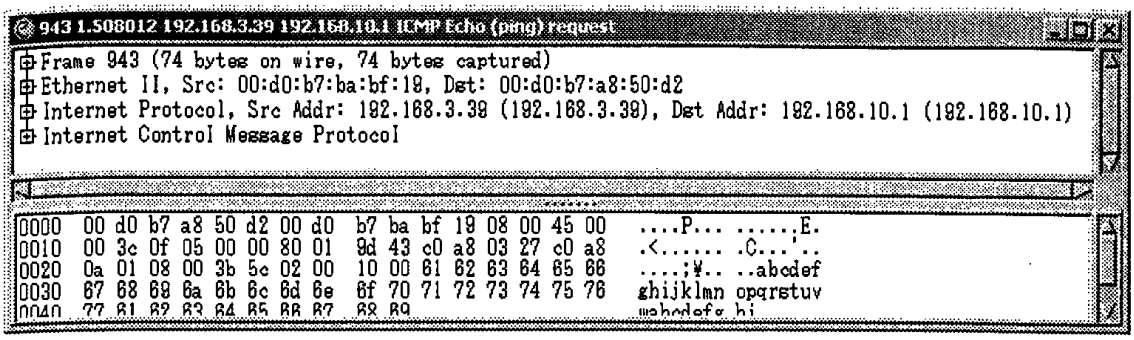


Рис. 10.8. Исходная дейтаграмма, содержащая запрос ICMP echo request

Защитный экран, получая дейтаграмму, принимает решение о том, как передавать ее дальше. Здесь основой для этого являются таблицы маршрутизации — ведь защитный экран в общем случае представляет собой маршрутизатор с расширенной поддержкой правил фильтрации и трансляции адресов. Так как в предлагаемой в качестве примера сетевой конфигурации существует отдельный сервер VPN, то защитный экран должен быть настроен на дальнейшую передачу поступившей дейтаграммы серверу VPN. Для этого можно воспользоваться статической маршрутизацией, или динамической (например с помощью протокола RIP). В рассматриваемом примере защитный экран должен иметь в своей таблице маршрутизации запись, говорящую о том, что «все дейтаграммы, направленные в сеть 192.168.10.0/24, нужно перенаправлять по адресу 192.168.3.252 (адрес сервера VPN для объекта 1)». Обратите внимание, что защитный экран, получив от клиента первую дейтаграмму, перешлет ее и одновременно отправит клиенту сообщение ICMP redirect (рис. 10.9), указывающее на то, что все последующие дейтаграммы для адреса 192.168.10.1 следует отправлять напрямую по адресу 192.168.3.252 (адрес сервера VPN), минуя защитный экран. Это позволит сократить задержки, которые вносятся лишним переходом (хоть он и находится в рамках одной локальной сети).

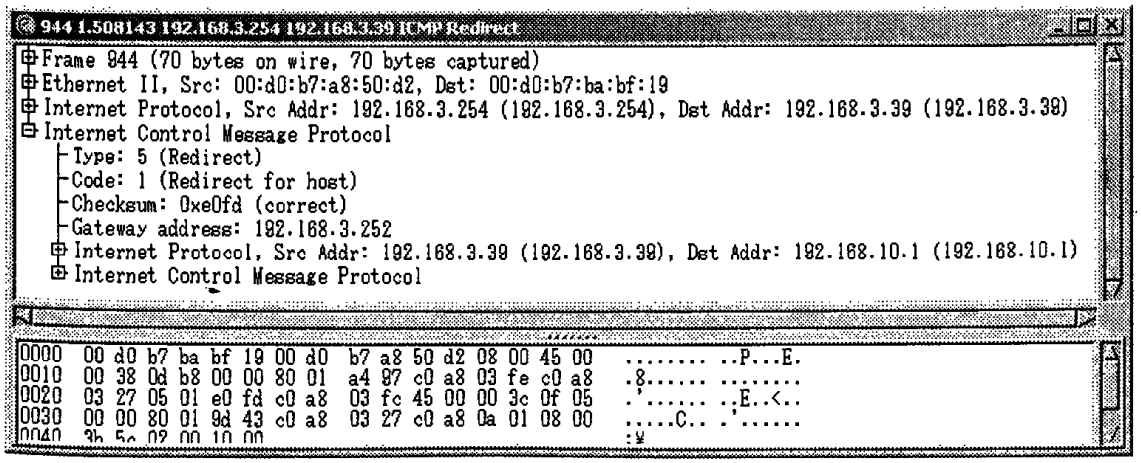


Рис. 10.9. Сообщение о перенаправлении

Далее, на сервере VPN указано, что дейтаграммы с адресом получателя из сети 192.168.10.0/24 должны отправляться через созданный интерфейс DOD VPN

(согласно правилу в таблице маршрутизации). Сервер получает дейтаграмму, анализирует адрес получателя, видит, что она направлена в известную ему сеть, и начинает выполнять следующие действия (обобщенно). Дейтаграмма целиком шифруется и единым блоком помещается в новую дейтаграмму с новым заголовком, в котором в качестве адреса получателя указывается внешний (зарегистрированный в сети Интернет) адрес сервера-партнера по сети VPN. После этого сервер проводит маршрутизацию полученной дейтаграммы заново. Так как на сервере VPN указан шлюз по умолчанию, на который нужно отправлять дейтаграммы для всех неизвестных адресов, то сервер отправляет ее шлюзу (защитному экрану), передающему ее дальше в сеть Интернет. Дейтаграмма «путешествует» по сети Интернет до тех пор, пока не поступит на внешний интерфейс защитного экрана на объекте 2, который по настроенным правилам ретранслирует дейтаграмму серверу VPN, который, в свою очередь, распакует и расшифрует исходную дейтаграмму.

Теперь рассмотрим процесс в деталях. Дейтаграмма, направленная на объект 2 (и содержащая запрос ICMP echo request) поступает на физический сетевой интерфейс сервера VPN с адресом 192.168.3.252. На сервере к ней добавляется заголовок протокола PPP. Затем она целиком шифруется (вместе с добавленным заголовком протокола PPP) и инкапсулируется в пакет GRE. Далее к получившемуся пакету GRE добавляется новый заголовок протокола IP и вновь сформированная дейтаграмма отправляется через сеть Интернет по адресу удаленного конца виртуального соединения. На обратном конце сервер выполняет обратную операцию — расшифровывает исходную дейтаграмму и отправляет ее к нужному получателю. Рисунок 10.10 показывает, как инкапсулированная дейтаграмма выглядит на сетевом уровне. На нем наглядно видны различные заголовки протоколов, которые добавляются к исходной дейтаграмме.

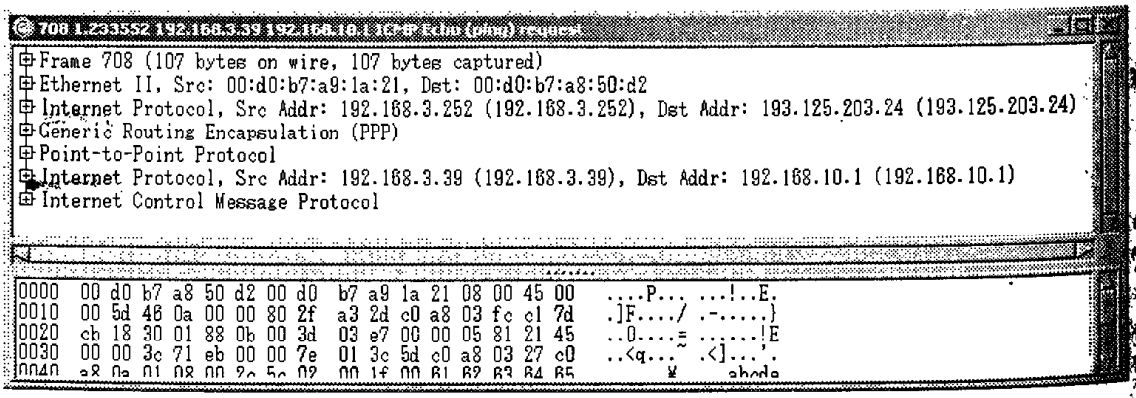


Рис. 10.10. Дейтаграмма в состоянии готовности к продвижению через Интернет

Учитывая то, что дейтаграмма была перехвачена в сети Ethernet, в ней присутствует заголовок Ethernet II, далее следует стандартный заголовок дейтаграммы IP (сейчас MAC-адресами можно пренебречь). То есть до этого момента на рисунке мы видим классическую дейтаграмму. Но вот далее, согласно теории, добавлены заголовки протоколов GRE и PPP, за которыми следует оригинальная дейтаграмма (исходный запрос ICMP echo request), которую нужно бы

инкапсулировать и зашифровать (в этом примере туннель PPTP был настроен на работу без шифрации и сжатия данных — в противном случае результирующая дейтаграмма была бы не информативна).

Обратите внимание на адреса в новой дейтаграмме. Адрес отправителя 192.169.3.252 (это сервер VPN на объекте 1), а адрес получателя — 193.125.203.24 — это адрес внешнего интерфейса защитного экрана на объекте 2. Рассматриваемая дейтаграмма была перехвачена на защитном экране объекта 1 — сервер VPN выполнил преобразование и вновь сформированную дейтаграмму отправил в сеть Интернет по адресу 193.125.203.24. А так как маршрут ему неизвестен, то дейтаграмма поступила по адресу шлюза по умолчанию или на внутренний интерфейс защитного экрана. Защитный экран отправляет дейтаграмму далее в сеть Интернет, но предварительно выполняет трансляцию адресов — заменяет в новой дейтаграмме адрес отправителя с 192.168.3.252 на 193.125.203.1 (подставляет адрес своего внешнего интерфейса — работает функция трансляции адресов (NAT) — особенность защитного экрана Kerio WinRoute Pro) и пересчитывает поле контрольной суммы. Кроме того, уменьшается на единицу поле Time to live после прохождения через защитный экран (согласно теории, маршрутизаторы декрементируют это поле на 1 при передаче дейтаграммы — см. главу 2). В таком виде дейтаграмма и поступает в сеть Интернет. Предложенный пример скорее исключение из правил, так как для наглядности были принудительно отключены опции шифрования и сжатия данных. В результате анализатор протоколов смог расшифровать все инкапсулированные данные в новой дейтаграмме. В том случае, если бы содержимое сжималось или (и) шифровалось, эта же самая дейтаграмма была бы теоретически нечитаемой анализатором. В своем дальнейшем пути дейтаграмма, минуя промежуточные маршрутизаторы сети Интернет, поступит на внешний интерфейс защитного экрана на объекте 2, который перенаправит ее локальному серверу VPN. Сервер выполнит обратную процедуру — удалит заголовки протоколов IP, GRE и PPP, восстановит исходную дейтаграмму и отправит ее хосту с адресом 192.168.10.1 (это будет обычный запрос ping). Ответ пройдет аналогичную цепочку преобразований, пока не достигнет отправителя запроса.

Рассмотрим другой, более сложный пример сети, в которой три объекта связаны между собой через сеть Интернет с использованием протокола PPTP (рис. 10.11).

Здесь три объекта подключены к сети Интернет. Но если рассматривать картину с точки зрения пользователей, все три объекта связаны между собой и находятся на расстоянии двух переходов друг от друга. Обратите внимание на то, что топология сети может отличаться от приведенной на рисунке, где туннели созданы в расчете на формирование логической схемы «звезда» с центром на объекте 1 (например, если он является центральным офисом компании). Здесь также можно было бы создать туннель между объектом 2 и объектом 3, а затем организовать туннель между последним и центральным офисом. В этом случае трафик между центральным офисом и объектом 3 имеет на своем пути один транзитный узел (то есть для конечных устройств количество переходов будет равно 3). Однако, учитывая, что, как правило, трафик чаще всего передается между центральным офисом и удаленными площадками, выбранная топология туннелей оправдана. Но также обратите внимание, что чем больше туннелей создается, тем жестче нагрузка на сервер в центральном офисе, который является одним из окончаний туннелей (в нашем случае сервер VPN на объекте 1) и, естественно,

этот сервер будет критичным узлом в сети. Опыт показывает, что самый обычный сервер на платформе Intel Pentium II может без затруднений обслуживать схему, в которой один центральный офис и десять удаленных объектов связаны по вышеприведенной схеме.

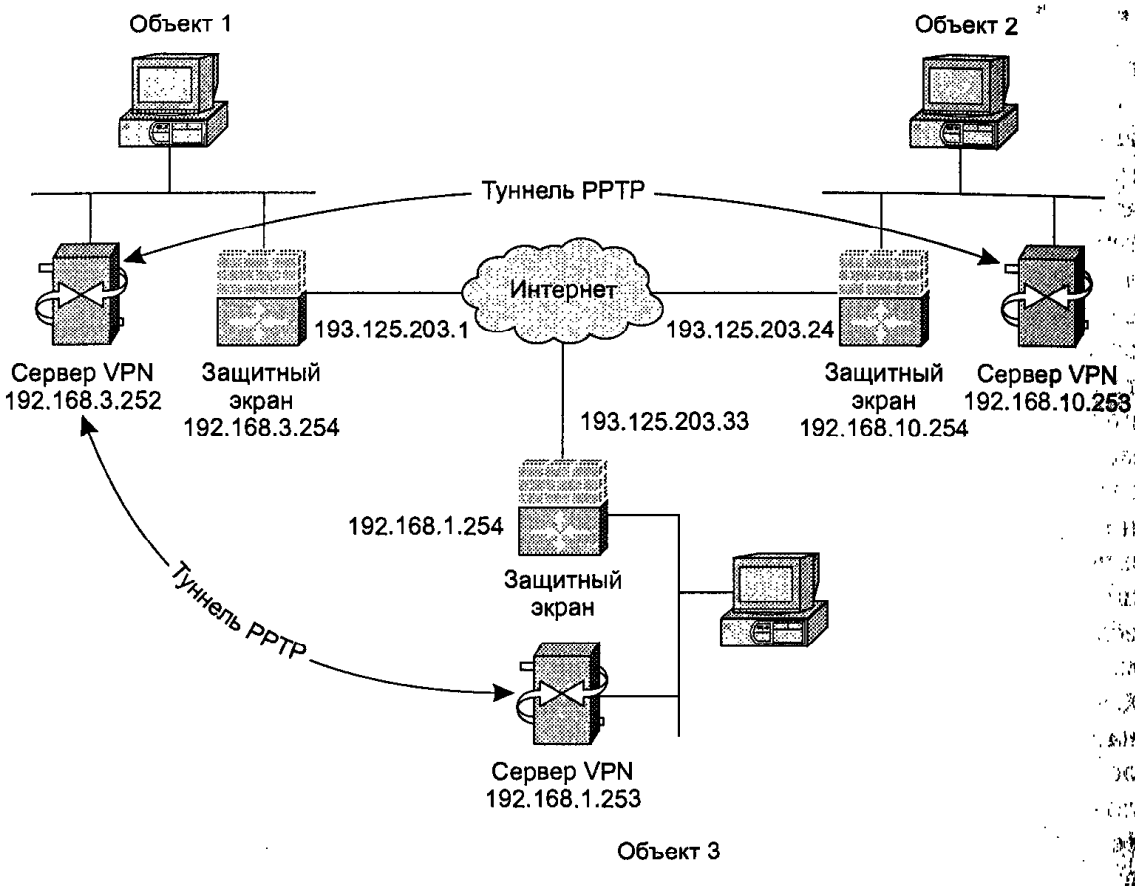


Рис. 10.11. Пример связи трех объектов

В предположении, что серверы помимо всего прочего выполняют роли маршрутизаторов, необходимо настроить и корректную маршрутизацию во всей сети. Перед этим рекомендуется детально проработать адресную схему, благо что администратор волен задействовать целые сети (например, из диапазона 192.168.X.X/24) даже для небольших (по количеству компьютеров) удаленных точек вместо того, чтобы заниматься выделением подсетей — помимо подсетей будет еще масса проблем, которые нужно будет решать, поэтому лучше облегчать себе жизнь на самых первых шагах.

Напомним, что после создания интерфейса DOD VPN, он может (и должен) участвовать в деле маршрутизации. Рассмотрим еще раз схему маршрутизации (только более детально) и прохождения информации из центрального офиса в объект 2 (один из внутренних клиентов вновь выполняет команду ping 192.168.10.1). До этого детализировался простой пример, когда туннель VPN связывает только два объекта — теперь сделаем то же самое для более жизненного примера.

Предположим, что для всех клиентов на объекте 1 с помощью сервера DHCP выдается адрес шлюза по умолчанию 192.168.3.254 (разумеется, помимо другой адресной информации). После проверки адреса получателя (а он принадлежит сети 192.168.10.0 и маршрут клиенту не известен) дейтаграмма отправляется шлюзу по умолчанию — серверу, который выполняет роль защитного экрана сети центрального офиса. Дейтаграмма поступает на внутренний интерфейс и после этого принимается решение о том, куда ее послать далее. На защитном экране работают служба RRAS и протокол маршрутизации RIP (версия 2). Тут следует отметить, что вместо использования динамической маршрутизации вполне можно решить все задачи с помощью статических маршрутов, задаваемых «вручную». Рекомендации относительно выбора того или иного метода приводить сложно, так как каждый имеет свои плюсы и минусы. Отмечу только, что протокол RIP в состоянии помочь избежать проблем с маршрутами при изменении адресной схемы — он просто начнет рассылать информацию о новых сетях.

Защитный экран проверяет свою адресную таблицу (сейчас не будем акцентировать внимание на то, каким образом она была построена — статически или с помощью протокола динамической маршрутизации) и обнаруживает, что маршрут в сеть с адресом 192.168.10.0/24 знает сервер VPN и, соответственно, «заворачивает» полученную дейтаграмму, направляя ее этому серверу. Последний получает ее и выполняет проверку таблицы маршрутизации. На этом сервере «заведен» статический маршрут, указывающий на то, что весь трафик, адресованный в искомую сеть, должен отправляться через интерфейс DOD VPN, связанный с сервером на объекте 2. На рис. 10.12 показан пример статического маршрута в сеть 192.168.10.0/24 — согласно этой записи все дейтаграммы, направляемые в данную сеть, должны отправляться через интерфейс DOD VPN с именем SH4 (интерфейс может именоваться по выбору администратора).

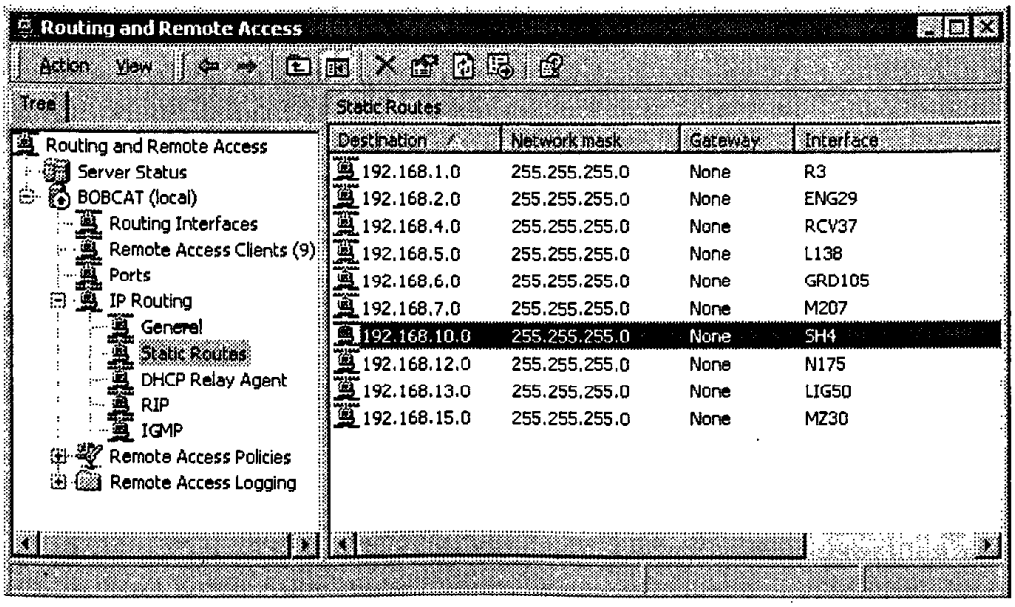


Рис. 10.12. Пример статической маршрутизации

Дейтаграмма проходит стадию шифрования и отправляется в сеть Интернет. При этом адрес получателя устанавливается в адрес, присвоенный внешнему интерфейсу защитного экрана на объекте 2 (193.125.203.24 — адрес сервера-партнера по туннелю указывается при его настройке). Роль защитного экрана в процессе маршрутизации минимальна — на нем настроены соответствующие правила, согласно которым трафик протокола PPTP (а точнее пакеты GRE и сегменты TCP с портом назначения 1723) должен передаваться во внутреннюю сеть серверу VPN. Этот сервер, получая дейтаграмму, расшифровывает ее и проверяет оригинальный заголовок для уточнения получателя. А получатель оригинальной дейтаграммы имеет адрес 192.168.10.1 — сервер просто посылает ее по данному адресу. Ответ на запрос ping проходит всю описанную цепочку, но в обратном порядке. Можно посчитать, через сколько маршрутизаторов прошла одна дейтаграмма — через четыре. То есть правильнее сказать, что до сети 192.169.10.0 из центрального офиса четыре перехода и первые два маршрутизатора, через которые прошла дейтаграмма, находятся в одной локальной сети (защитный экран и сервер VPN). К сожалению, помимо нагрузки на серверы VPN, нагрузка так же ложится и на защитные экраны — им нужно «заворачивать» дейтаграммы, направленные в удаленные точки, связанные туннелями (отчасти помогает сообщение ICMP redirect — в результате количество переходов сокращается).

Не очень простая схема? Но если рассмотреть всю цепочку передачи одной дейтаграммы, то картина будет еще более запутанная. Ведь сейчас не было учтено, как именно передается зашифрованная дейтаграмма от сервера VPN на объекте 1. Было сказано «отправляется по туннелю», но на практике добавляются еще задачи — дейтаграмма шифруется и вновь отправляется защитному экрану, так как после шифрации исходной дейтаграммы к ней добавляется новый заголовок, где в качестве адреса получателя устанавливается 193.125.203.24, а в качестве адреса отправителя адрес сервера VPN — 192.168.3.252. Вновь сформированная дейтаграмма отправляется шлюзу по умолчанию — а именно защитному экрану, поскольку сервер VPN абсолютно не владеет информацией о маршруте в сеть 192.168.10.0/24 — для него это где-то в сети Интернет. Можно, конечно, прописать на нем статический маршрут, да только смысла в этом не будет никакого — все равно дейтаграмма должна выйти за пределы внутренней сети через защитный экран. Защитный экран, получая дейтаграмму, проверяет свою таблицу маршрутизации, не находит маршрут в нужную сеть и передает ее далее в Интернет. Однако перед тем как отправить ее, он должен выполнить трансляцию адреса, так как адрес отправителя установлен в 192.168.3.252 (напомним, что это адрес из диапазона немаршрутизируемых в сети Интернет). Если трансляцию не выполнить и послать дейтаграмму без изменения адреса отправителя, то она должна дойти до адресата — ведь промежуточные маршрутизаторы проверяют только адрес получателя — да только вот ответа можно уже не дождаться — сеть 192.168.3.0/24 относится к разряду немаршрутизируемых.

При трансляции адреса дейтаграмма претерпевает изменения — меняется адрес отправителя и порт на его стороне (изменение порта выполняется, если защитный экран настроен на подстановку только одного адреса, и в таком случае номер порта позволяет идентифицировать сеансы от разных хостов). И только после этого дейтаграмма отправляется в путешествие через сеть Интернет до внешнего интерфейса защитного экрана на объекте 2, который транслирует ее без изменения локальному серверу VPN. Там дейтаграмма вновь претерпевает

изменения — удаляется новый заголовок, а расшифрованные данные формируют исходную дейтаграмму. Аналогично можно посчитать количество трансформаций — их тоже три. Учитывая то, что на все требуется процессорное время, начинаешь к рекомендациям производителей по поводу аппаратного обеспечения относиться с меньшим скепсисом (кстати, скептицизм этот сугубо российский, на Западе к рекомендациям относятся более внимательно). Можно задать вопрос — что будет, если все же защитный экран не выполнит трансляцию адреса и дейтаграмма уйдет в сеть Интернет с адресом отправителя 192.168.3.252 (если принять как версию, что до сервера VPN она дойдет)? Ведь новый заголовок (с этим адресом) будет там удален!

Методы шифрования информации

Для выполнения процедуры шифрования данных протокол PPTP использует процесс RAS с разделяемым закрытым ключом (RAS «shared-secret» encryption process). Такое название он получил из-за того, что оба участника туннеля используют один ключ шифрования. В реализации PPTP от компании Microsoft этот ключ получается с помощью пользовательской информации, вводимой на стадии настройки туннеля (то есть имени и пароля учетной записи, созданной для туннеля). Кроме того, протокол PPTP может дополнительно сжимать передаваемые по туннелю данные для уменьшения их размера.

Стандарт на протокол PPTP не уточняет, как именно должно выполняться шифрование данных, а вместо этого ссылается на технологию MPPE (Microsoft Point-to-Point Encryption, шифрование на каналах типа «точка-точка», см. документ RFC 3078). Как легко догадаться из названия, эта технология разработана компанией Microsoft для организации процедуры шифрования информации на каналах типа «точка-точка». Сама по себе технология MPPE является расширением технологии Microsoft Point-to-Point Compression (MPPC, RFC 2118), которая применяется для сжатия данных, передаваемых в пакетах протокола PPP, и была разработана для оптимизации разделения полосы пропускания между множеством одновременных соединений.

Технология MPPE использует алгоритм RC4 (лицензия компании RSA Data Security Inc., www.rsa.com) с различными длинами ключей шифрования (например 40 бит или 128 бит), которые, в свою очередь, получаются из информации пользователя. Алгоритм RC4 относится к потоковому (stream chiper) и, следовательно, размеры зашифрованной и расшифрованной дейтаграммы одинаковы (это очень важно, так как алгоритм не вносит накладных расходов). Отметим, что реализация технологии MPPE от компании Cisco Systems полностью совместима с возможностями, предлагаемыми Microsoft. В их перечень входит возможность работы «без истории» (режим historyless), которая позволяет повысить пропускную способность на ненадежных каналах связи (в том числе и в туннелях виртуальных частных сетей, проходящих через сеть Интернет), так как ни одна из сторон не должна синхронизировать контекст шифрования при потере дейтаграмм. Данные шифруются при использовании технологии MPPE только в том случае, когда была согласована аутентификация с помощью алгоритмов MS-CHAP, MS-CHAP v2 или EAP/TLS, которые позволяют генерировать свои

собственные начальные ключи для шифрования. При этом оба участника получают один ключ.

К СВЕДЕНИЮ

RC4 — это потоковый алгоритм для побайтового шифрования входных данных с использованием ключа, длину которого можно варьировать в зависимости от желаемой скорости обработки информации и стойкости шифра. Алгоритм работает очень быстро и поэтому часто применяется там, где требуется шифрование непрерывно передаваемых данных. По своей сути RC4 представляет собой генератор псевдослучайных чисел, которые затем комбинируются с шифруемыми данными при помощи операции побитового сложения по модулю 2.

Технология MPPE настраивается на вкладке Encryption в диалоговом окне свойства политики удаленного соединения (remote access policy) на использование ключей разного размера (рис. 10.13). Так, при установке флажка Basic (базовое шифрование) используется 40-битовый ключ, флажок Strong (строгое) подразумевает ключ длиной 56 бит, а флажок Strongest (самое строгое) задает ключ длиной 128 бит. Первоначально 40- и 56-битовые ключи предназначались для использования за пределами США, а ключ длиной в 128 бит не подлежал применению за пределами США и был доступен только в соответствующей версии операционной системы Windows 2000. Однако с выходом второго пакета обновления (Service Pack 2) интернациональные пользователи также получили возможность использования 128-битового ключа для создания зашифрованных туннелей. Такое стало возможным и по причине смягчения экспортных требований, осуществленного правительством США в начале 2000 года.

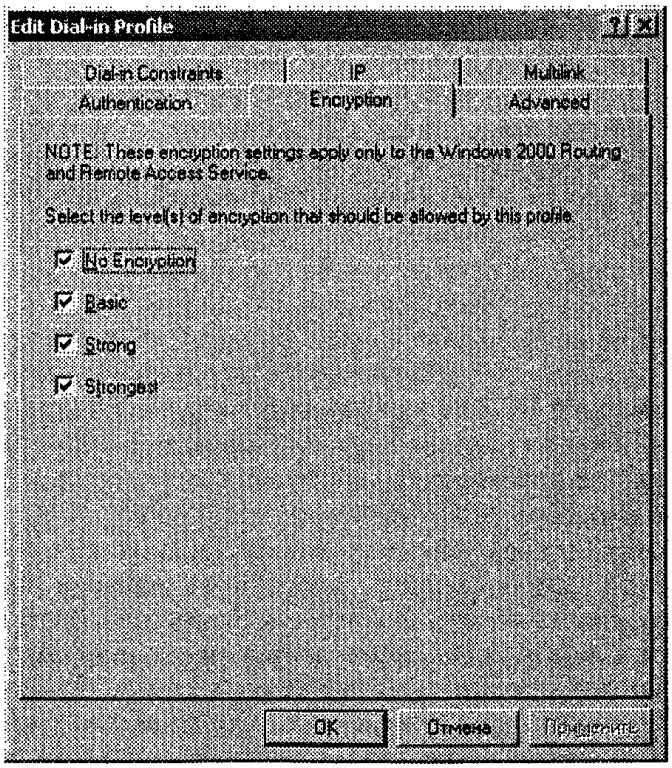


Рис. 10.13. Настройка длины ключа для шифрования

После того как вы установите второй пакет обновлений или иначе Service Pack 2, ваша операционная система будет обновлена, и любая система, поддерживающая максимально 56-битовый ключ, будет подготовлена до наивысшего уровня шифрования. Интересно, что после обновления вернуться обратно на предыдущий уровень шифрования будет невозможно, даже при деинсталляции пакета обновления. Кроме того, подобное обновление осуществляется автоматически и пользователь не имеет возможности его запретить.

Ключи для сеанса формируются на основе пользовательского пароля. В случае применения алгоритмов MS-CHAP (v1 и v2) необходимо предпринять дополнительные меры по хранению пароля в тайне и выбирать его следует таким образом, чтобы снизить вероятность подбора. Так, можно порекомендовать воспользоваться генераторами случайных паролей при настройке туннеля (точнее, при создании в домене пользователя, который затем будет указываться в свойствах интерфейса DOD VPN — если, конечно, администратор предпочел такую схему), а затем пароль можно «забыть». Ведь он будет нужен только один раз, и всегда есть возможность его поменять. Важно отметить, что ключ шифрования отправляемых дейтаграмм меняется для каждой из них и, следовательно, расшифровка каждой дейтаграммы не зависит от получения предыдущих. Для того, чтобы синхронизировать ключи, используется специальное поле с номером последовательности, и если дейтаграммы потерялись в пути или пришли не в порядке их отправления, то ключи адаптируются к изменению условий.

Для того чтобы показать, как именно работает протокол PPTP на сетевом уровне, и продемонстрировать, что он действительно обеспечивает должный уровень безопасности (точнее, выполняет шифрование данных, так как анализ уровня криптоустойчивости алгоритма RC4 выходит за рамки главы), выполним команду ping в отношении удаленного сервера, находящегося на другом конце виртуального соединения (адрес 192.168.10.254 — это адрес внутреннего интерфейса защитного экрана на объекте 2). Сама команда выполняется с защитного экрана, установленного на объекте 3 (рис. 10.11), имеющего адрес 192.168.1.253 (также адрес внутреннего интерфейса — напомним, что если хост имеет несколько сетевых интерфейсов, то при отправке дейтаграмм подставляется адрес того интерфейса, через который дейтаграмма поступает в сеть). Кстати, одна особенность команды ping в операционной системе Windows 2000 очень полезна — по умолчанию она заполняет повторяющимися символами алфавита поле данных. Ниже показан пример, где для наглядности размер поля данных был задан равным 512 байт и все эти 512 байт были упорядоченно заполнены английским алфавитом. Результирующий размер дейтаграммы составляет 540 байт (рис. 10.14).

Учитывая, что адрес получателя дейтаграммы находится в другой сети (192.168.10/24) и маршрут до этой сети предполагает прохождение сквозь туннели PPTP (сначала через туннель до объекта 1, а затем через туннель от объекта 1 до объекта 2), дейтаграмма посылается серверу VPN на объекте 1, который знает маршрут в искомую сеть (то есть этот сервер является партнером сразу двух других серверов VPN — на объектах 2 и 3). Сервер VPN (его адрес 192.168.1.253) на объекте 3 согласно заложенной логике выполняет шифрование дейтаграммы и отправляет ее в сеть Интернет. После выполнения процедуры шифрования исходная дейтаграмма выглядит, как это показано на рис. 10.15.

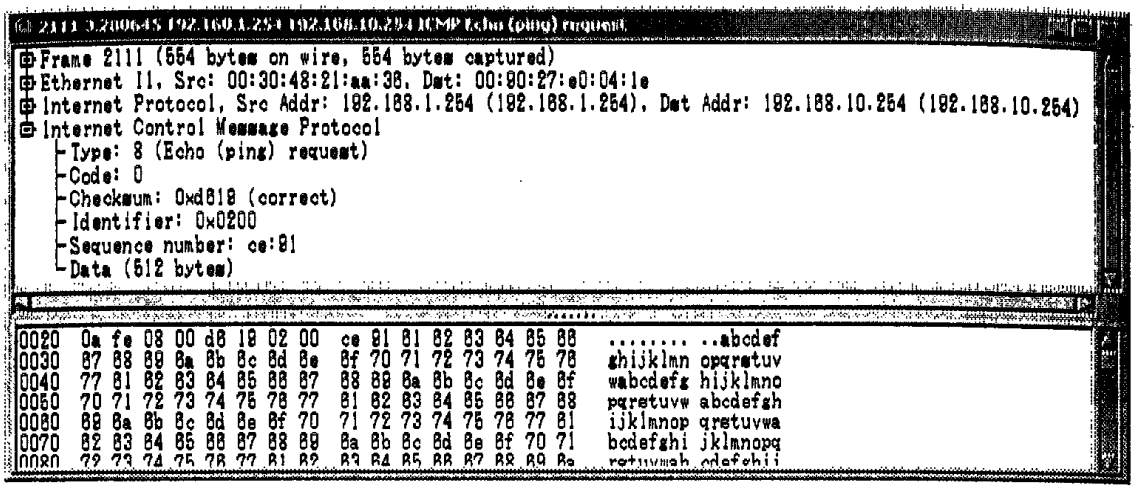


Рис. 10.14. Исходная дейтаграмма, содержащая запрос ICMP echo request

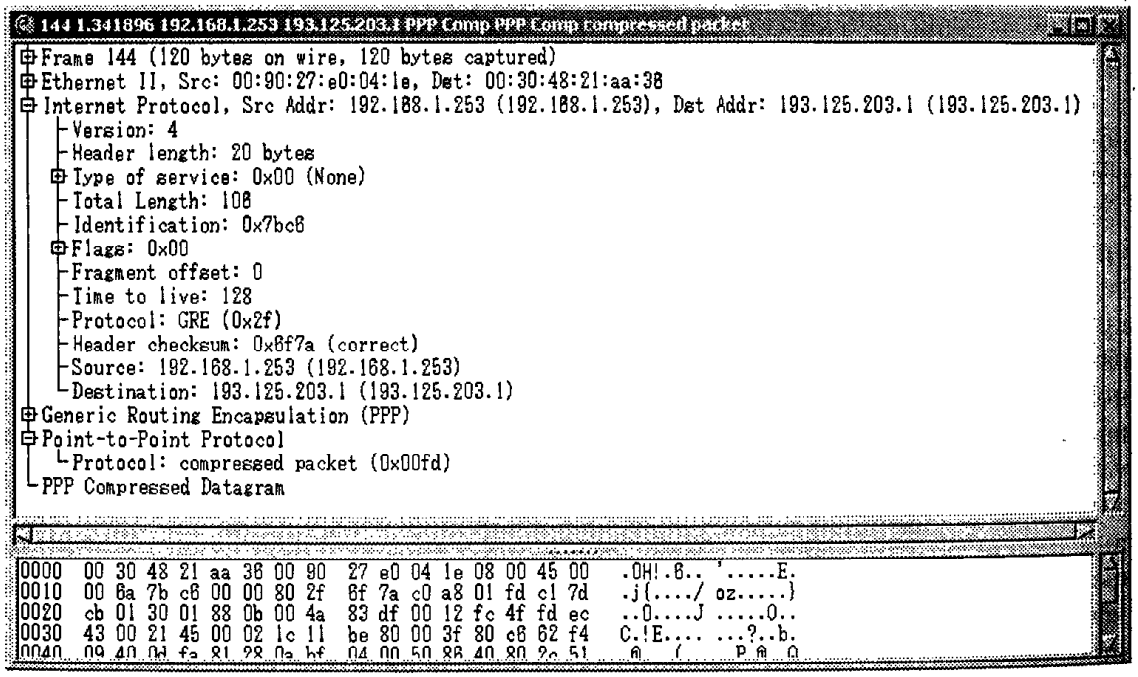


Рис. 10.15. Запрос ICMP echo request после процедуры шифрования

Видно, что формируется новая дейтаграмма. К ней добавляется заголовок протокола GRE с указанием того, что вложенные данные представляют собой пакет протокола PPP. В пакете протокола PPP доступно только одно поле заголовка, а именно Protocol (протокол), которое указывает на то, что содержимое пакета сжато (compressed packet). Хитрость заключается в том, что числовое значение 0x00fd свидетельствует, что пакет либо сжат (работает протокол MPPC), либо зашифрован (работает протокол MPPE), либо все вместе. Просто в данном случае анализатор протокола посчитал, что пакет сжат. Посмотрите, размер исходной дейтаграммы больше 512 байт (512 байт данных ICMP плюс заголовки — итого 540 байт), но после того как она была отправлена в сеть Интернет (или

иными словами, после ее обработки сервером VPN), ее размер сократился до 106 байт.

Согласно логике работы алгоритма шифрования данных (RC4), используемого в протоколе PPTP (а точнее в технологии MPPE), данные не увеличиваются и не уменьшаются в размере после процедуры шифрования — их размер остается прежним. То есть можно сделать вывод, что данные были предварительно сжаты с помощью протокола MPPC, и уже затем зашифрованы. Трудно утверждать, что это добавляет надежности передаваемым данным, но вот нагрузку на серверы-партнеры виртуальной частной сети это добавляет несомненно. С другой стороны, общий размер исходной дейтаграммы был 540 байт, а после процедур сжатия и шифрования получилась дейтаграмма длиной 106 байт. Пятикратное сжатие позволяет достичь более высоких скоростей (точнее говоря, за промежуток времени для передачи 540 байт можно будет передать пять дейтаграмм по 106 байт, выигрывая в скорости). Правда, следует отметить, что приведенный пример был не совсем корректен с точки зрения выигрыша в размере — заполнение поля данных сообщения ICMP имеет большую избыточность. Для сжатия данных используется популярный алгоритм LZ, далеко не во всех случаях обеспечивающий такой высокий уровень сжатия (см. Документ: Lempel, A. and Ziv, J., *A Universal Algorithm for Sequential, Data Compression*, IEEE Transactions On Information Theory, Vol. IT-23, No. 3, May 1977).

Несколько слов относительно уязвимости технологии PPTP. Есть интересная статья *The Crumbling Tunnel* (*security tunnel*) по адресу http://solaris.opennet.ru/base/sec/p53-12.txt.html, где рассматривается возможность компрометации туннеля с помощью различных методов. Учитывая сказанное, рекомендуется при настройке правил на защитных экранах ограничивать права отправителя и получателя трафика туннеля. Например при настройке соединения между двумя серверами, как правило, внешние граничные адреса серверов известны и неизменны. То есть следует, например, указать, что сегменты TCP с портом 1723 могут проходить через защитный экран только в том случае, если они направлены по адресу удаленного сервера и никак иначе. Такие меры позволят снизить риск от несанкционированного доступа. Однако, вне всяких сомнений, они не помогут от возможного перехвата информации и последующей ее расшифровки.

Анализ криптоустойчивости туннеля PPTP выходит за рамки книги, однако следует заметить, что в операционной системе Windows 2000 реализован ряд решений, которые затрудняют расшифровку перехваченных данных. Одним из них, например, является смена ключа шифрования для каждой дейтаграммы. В технической литературе такой режим иногда называют «безразличным» (stateless), и при его использовании (а он включен по умолчанию) расшифровка следующей дейтаграммы не зависит от предыдущей. Уникальный ключ для каждого пакета значительно повышает нагрузку на модуль шифрования на сервере, однако позволяет серверам не ожидать поступления предыдущего пакета, для того чтобы расшифровать текущий. В результате значительно повышается производительность на критичных каналах связи.

Самым очевидным методом взлома является перехват зашифрованных дейтаграмм с последующей попыткой их дешифрации. Здесь стоит полагаться на надежность и устойчивость криптографического алгоритма, применяемого для шифрования данных. В целом можно предположить, что такое действие будет

не под силу одному человеку или группе лиц, за которыми не стоит технически оснащенная организация. Дейтаграммы следуют от маршрутизатора к маршрутизатору, которые, как правило, установлены у авторитетных и крупных поставщиков услуг Интернета. При этом получить доступ для проведения анализа данных довольно затруднительно. Тем не менее, атакующий может попытаться сам установить туннель PPTP со внутренним сервером и, как следствие, получить доступ во внутреннюю сеть. Грубо говоря, все что нужно для «взлома» протокола PPTP и реализации туннелей на базе Microsoft Windows 2000, это просто настроить интерфейс DOD VPN, указать IP-адрес сервера и имя и пароль пользователя. Адрес сервера не так трудно выяснить с помощью сканирования портов — например, атакующий может провести сканирование и увидеть, что порт 1723 протокола TCP открыт на таком-то хосте.

В схеме, показанной на рис. 10.11, этот порт может быть открыт на внешнем интерфейсе защитного экрана, прикрывающего сервер VPN. При этом защитный экран выступает простым ретранслятором — получая сегменты TCP с портом назначения 1723 (как и дейтаграммы, содержащие данные протокола PPTP), он просто транслирует их внутреннему серверу. То есть задача атакующего сводится только к знанию имени и пароля пользователя, которому разрешено устанавливать туннель, или к попыткам эту информацию найти методом подбора. Самым простым и, на взгляд автора, надежным методом защиты будет ограничение на защитном экране адресов, с которых могут поступать данные, принадлежащие туннелю PPTP. Напомним, что эти данные слагаются из сегментов TCP с целевым портом 1723, принадлежащих управляющему соединению PPTP, и дейтаграммы IP, переносящей сами зашифрованные данные (пакеты GRP). Практически любой современный защитный экран позволит ввести такие ограничения, которые упрощаются тем, что IP-адреса серверов-партнеров всегда известны и статичны. Впрочем, в том случае, если есть мобильные клиенты, которые периодически подключаются через Интернет к корпоративной сети, ситуация усложняется, так как здесь адреса будут динамическими. На опыте эту проблему автор решал обнаружением целой сети провайдера, на телефоны модемного пула которого чаще всего звонят мобильные пользователи (например те, кто подключается через переносной компьютер и модем GSM), и разрешал доступ для всей сети. До сих пор такое решение работает, хотя, несомненно, оно несет в себе некий элемент риска — мало ли кто еще с адреса этой сети (обычно класса C для России) будет пытаться установить соединение. Дополнительно на защитном экране можно порекомендовать ограничить исходящий в сеть Интернет трафик от внутреннего сервера, принадлежащий протоколу PPTP. Причем ограничить по той же адресной группе, которая создавалась для фильтрации входящего трафика. Подобные настройки стоит выполнить на всех защитных экранах, «опекающих» серверы VPN, естественно, создавая каждый раз свои адресные группы в соответствии с выбранной топологией построения VPN. В результате атакующий просто не сумеет «прокопать» туннель PPTP к удаленной точке сети, так как защитный экран пакеты от его адреса просто не пропустит во внутреннюю сеть. Правда, тут открывается поле деятельности для выполнения атак, основанной на замене адреса отправителя на разрешенные адреса.

Например, предположим, что, как показано на рис. 10.11, два сервера установили соединение PPTP. Защитные экраны настроены на соответствующие правила прохождения и отсеивают неизвестные адреса. Но если нападающий знает пару адресов, не блокируемых экранами, то он может попытаться скомпрометировать соединение, посылая управляющие сегменты TCP или сами пакеты GRE с указанием правильных адресов отправителя. В этом случае защитные экраны пропустят означенные пакеты к внутренним серверам. Тем самым появляется вероятность организации атаки DoS, при которой установленное соединение может быть разорвано серверами по причине поступления большого количества пакетов. Отмечу, это только теоретические изыскания и на практике с подобным автор не сталкивался или просто не обращал внимания — ведь обычно в день туннели PPTP через Интернет разрываются по 2–3 раза (в большинстве случаев по причине резкого повышения задержки) и понять истинную причину перебоя довольно сложно.

Возвратимся к рассмотрению (рис. 10.15) прохождения получаемой дейтаграммы. Как видим, она отправляется по адресу 193.125.203.1 — на внешний интерфейс защитного экрана на объекте 1. Для того чтобы выполнить маршрутизацию, защитному экрану вовсе не обязательно анализировать содержимое дейтаграммы (да это после шифрования и невозможно), но он настроен таким образом, что дейтаграммы, поступающие на его внешний интерфейс и содержащие данные протокола PPTP (анализируется поле Protocol в дейтаграмме), должны транслироваться локальному серверу VPN. Необходимость подобного правила очевидна — подразумевается, что данные зашифрованы и особой ценности внутри сети без процедуры дешифрования не представляют, соответственно их нужно передать серверу, который сможет выполнить такую процедуру. Сервер VPN на объекте 1 получает эту дейтаграмму, расшифровывает ее и извлекает исходный запрос ICMP echo request. Затем он смотрит, по какому адресу полученную дейтаграмму нужно передать. Учитывая, что адрес получателя 192.168.10.253 и он принадлежит другой сети, транспортировка в которую подразумевает шифрование данных (отправка по туннелю DOD VPN), сервер выполняет обратную операцию — шифрует запрос ICMP echo, формирует новую дейтаграмму и отправляет ее далее по маршруту. Сервер VPN на объекте 3 получает дейтаграмму, расшифровывает ее и доставляет конечному получателю. Получатель (в данном случае — защитный экран объекта 3) принимает обычный запрос ping с адресом отправителя 192.168.1.254 и отвечает на него (кстати, ответ содержит тот же объем данных, что и запрос). Ответ проходит всю описанную цепочку преобразований вновь и поступает серверу VPN на объекте 3 (адрес на схеме 192.168.1.253).

То есть при построении сети, где все туннели сходятся на одном из объектов, выполнение одного запроса подразумевает под собой две процедуры шифрования и дешифрования данных. Как уже было отмечено, подобная схема построения будет оправдана, если в сети присутствует некий центр и если характер распределения трафика соответствует этой схеме. В противном случае при интенсивном обмене данными между объектами 2 и 3 рассмотренная модель будет нецелесообразной, так как сервер VPN на объекте 1 будет выполнять просто роль некого передаточного узла. Если характер трафика предполагает в большей сте-

пени обмен данными между объектами 2 и 3, то между ними можно настроить дополнительный туннель (потребуется только зарегистрировать маршруты на серверах VPN).

Примеры настройки защитного экрана

Если в своей сети читатель решит внедрить схему, рассмотренную выше, то следует не забыть о том, что защитные экраны на обоих концах соединения также нужно настраивать, иначе серверы не смогут связаться друг с другом. Методы настройки различных защитных экранов на поддержку протокола PPTP отличаются друг от друга, но принципы общие и опираются они на параметры соединения. Так, при использовании защитного экрана Kerio WinRoute Pro настройка проводится следующим образом. (Напомню, что этот программный продукт для своей работы использует технологию NAT/PAT, меняющей адрес отправителя и номер порта на стороне отправителя. Последняя функциональность нужна для того, чтобы выделять различные сеансы от различных хостов в защищаемой сети.) В качестве зарегистрированного IP-адреса используется адрес, назначенный внешнему интерфейсу защитного экрана. Для того чтобы дейтаграммы, принадлежащие протоколу PPTP, попадали к внутреннему серверу VPN, нужно настроить так называемое отображение портов (port mapping), где указать, что дейтаграммы, поступающие на определенный порт, должны отображаться (транслироваться) по указанному адресу до внутреннего сервера.

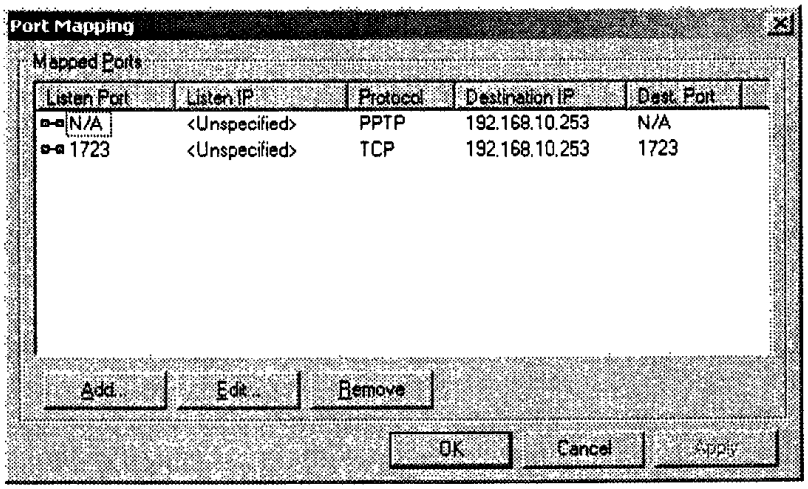


Рис. 10.16. Настройка параметров отображения портов

Из рис. 10.16 видно, что дейтаграммы, принадлежащие протоколу PPTP и направленные на адрес внешнего интерфейса защитного экрана на объекте 2, должны перенаправляться в защищаемую сеть по адресу 192.168.10.253 — это адрес внутреннего сервера VPN (сервера-партнера туннеля с объектом 1). Подобная настройка применяется для трансляции сегментов TCP, принадлежащих контрольному соединению протокола PPTP (номер порта назначения 1723). Подоб-

ные настройки нужно сделать на защитных экранах на обоих концах соединения (конечно, учитывая то, что адреса протокола IP будут другими).

Если на защитном экране не определен протокол PPTP, то это определение легко создать. Для чего следует указать, что все дейтаграммы с установленным в значение 0x2f полем Protocol в своих заголовках принадлежат означенному протоколу. Анализатор будет идентифицировать такие дейтаграммы как принадлежащие протоколу GRE, что не должно вводить вас в заблуждение, поскольку вся информация, проходящая по туннелю, построенному на базе протокола PPTP, передается с помощью протокола GRE. То есть можно сказать, что PPTP это название не протокола, а технологии. В этой технологии используются два протокола — протокол TCP с портом назначения 1723 и, как уже отмечено, дейтаграммы IP с инкапсулированным заголовком GRE (десятичное значение поля Protocol в заголовке дейтаграммы — 47).

Далее, если на защитном экране настроена фильтрация исходящего трафика в сеть Интернет, то нужно не забыть открыть доступ для серверов-партнеров виртуальной частной сети. Так, следует открыть доступ по протоколу TCP с портом получателя 1723 к удаленному защитному экрану. Например, на защитном экране, установленном на объекте 1, для защиты локальной сети включена фильтрации исходящего трафика. Для того чтобы сервер VPN на этом объекте мог установить и поддерживать управляющее соединение с сервером VPN на объекте 2, нужно настроить разрешающее правило (рис. 10.17). Согласно этому правилу, защитный экран разрешит прохождение сегментов протокола TCP с заданными адресами получателя и отправителя и портом на стороне получателя, равным 1723 (порт на стороне отправителя может быть любой — ключевое слово any). Этот пример применим к продукту Kerio WinRoute Pro и, естественно, процедура настройки для иных реализаций защитных экранов будет другой, сохраняется только общий принцип.

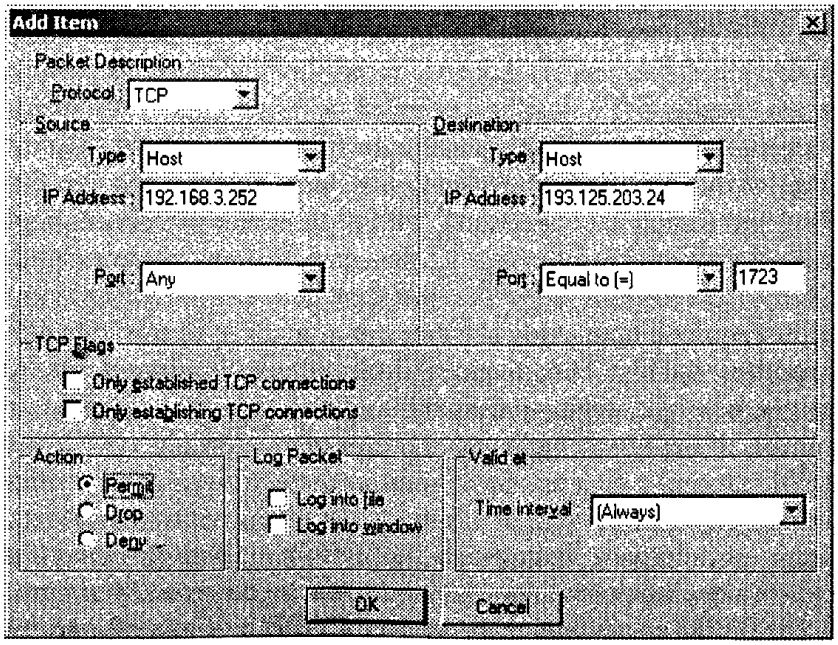


Рис. 10.17. Пример настройки правила фильтрации на защитном экране объекта 1

Учитывая то, что соединение PPTP двустороннее (это будет показано далее), то есть каждый сервер устанавливает соединение с удаленным сервером, нужно также разрешить проход через защитный экран и обратному трафику от удаленного сервера. То есть если сервер VPN на объекте 2 отправляет сегменты TCP с портом получателя 1723, то в ответ сервер VPN на объекте 1 будет посылать сегменты со случайным (или трудно предсказуемым) портом получателя, но с портом отправителя 1723 — это требуется учесть при настройке правил (рис. 10.18).

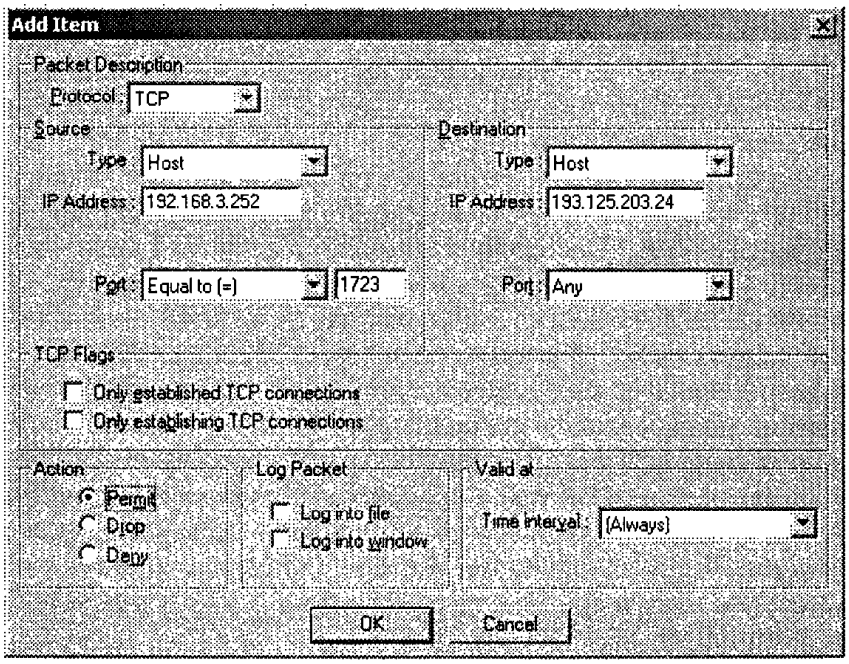


Рис. 10.18. Пример настройки правила фильтрации на защитном экране объекта 2

И наконец, следует разрешить прохождение через защитный экран дейтаграмм, содержащих заголовок GRE. В применении к защитному экрану Kerio WinRoute Pro, это не составляет труда, так как он изначально понимает, что означает протокол PPTP (у этого программного продукта есть встроенная поддержка протокола PPTP). Учитывая, что протокол PPTP работает на сетевом уровне (уровень IP), при его настройке не нужно указывать номера портов (рис. 10.19).

Безусловно, настраивать правила фильтрации нужно на каждом защитном экране, который может пропускать через себя трафик протокола PPTP. Приведенные настройки правил безопасности применимы также в том случае, если виртуальное соединение устанавливается не между серверами, а между сервером и клиентом. Например, вероятна следующая ситуация — компания имеет выход в сеть Интернет и для предоставления этой услуги своим сотрудникам из дома установила сервер доступа. Пользователи звонят на сервер и после авторизации (скажем, средствами протокола RADIUS) получают доступ в Интернет. Однако можно также предоставить им доступ и во внутреннюю сеть с помощью технологии PPTP. И хотя сама технология PPTP в большей степени ориентирована на использование между территориально удаленными точками, ничто не

мешает разрешить и описанный доступ. Если такая схема реализована, то компании не требуется создавать внутри сети еще один сервер доступа.

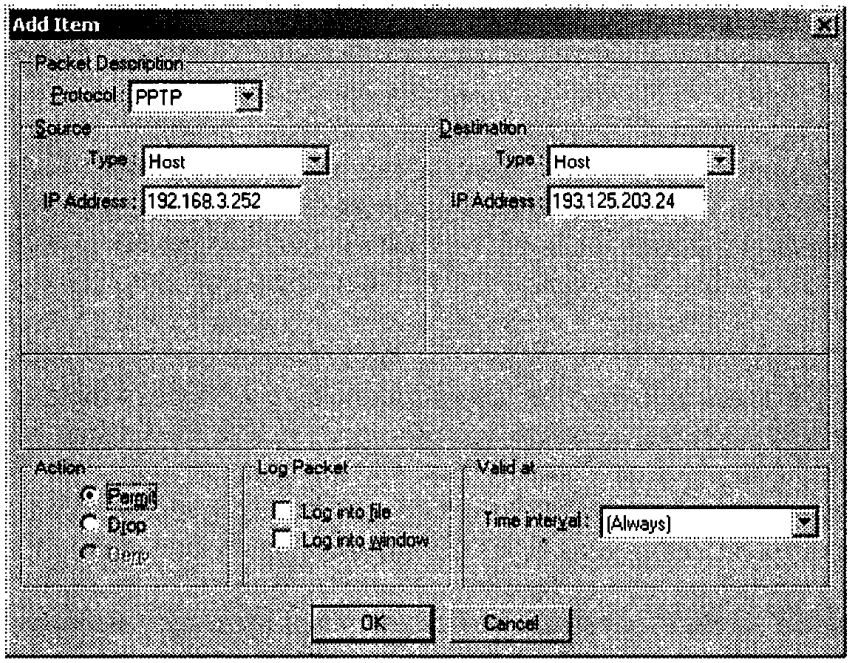


Рис. 10.19. Пример правила для протокола PPTP

Если «подняться» на уровень выше и посмотреть на связь трех рассматриваемых объектов с точки зрения структуры Active Directory (естественно при условии, что сеть построена на базе Microsoft Windows 2000), то можно предложить к рассмотрению схему, в которой на каждом объекте серверы VPN выполняют также роль контроллеров домена и участвуют в процедуре репликации информации домена. При этом схема репликации может либо настраиваться администратором собственноручно, либо создаваться автоматически самими серверами (процесс проверки целостности знаний, KCC — Knowledge Consistency Checker). В последнем случае, например, могут быть созданы репликационные связи между серверами на объектах 2 и 3. С точки зрения логики работы домена, это не создает никаких проблем — ведь в конечном итоге все контроллеры будут содержать одинаковую информацию. Но вот на сетевом уровне процесс репликации между этими серверами будет не оптимален с точки зрения трафика — поскольку пакеты будут проходить через сервер VPN на объекте 1 транзитом, создавая двойной трафик. Впрочем, подобные опасения становятся существенными только тогда, когда каждый из удаленных объектов содержал бы не десяток пользователей, а под тысячу. Здесь правильнее было бы настроить репликацию по схеме «звезда с центром в центральном офисе».

Вообще говоря, если для организации виртуальных сетей было выбрано программное решение от компании Microsoft, то администратор может выбирать схемы размещения серверов VPN из нескольких вариантов. Первый — когда сервер VPN располагается за защитным экраном, во внутренней сети. Такая

схема была выбрана для рассматриваемой сети (см. рис. 10.11). Преимущества данного решения в том, что сервер VPN скрыт защитным экраном и, в общем, не доступен для прямого воздействия. Под прямым воздействием понимается возможность посылать непосредственно пакеты серверу, а затем ожидать (или не ожидать) получения ответа. В последнем случае можно организовать атаку по типу DoS на сервер VPN с фальшивых адресов в надежде на то, что он перестанет нормально работать. Например, зная, что сервер использует протокол TCP с портом 1723 для управляющих соединений, можно попытаться «завалить» его потоком сегментов TCP (или более того, просто посылать различные команды по управляющему соединению в расчете на то, что серверы «упадут»). Если сервер находится за защитным экраном, то на последнем можно просто указать адреса, от которых разрешается прохождение во внутреннюю сеть сегментов TCP с портом 1723 и пакетов PPTP. Когда серверы VPN огорожены защитным периметром сети в виде экранов, их лучше включать в домен (например, домен Active Directory в силах охватить все объекты) и, как следствие, пользоваться единой политикой учетных записей. Так, при настройке туннеля PPTP на обоих серверах-партнерах указываются учетные записи, которые будут использоваться при проверке легитимности создания туннеля. Если серверы включены в домен, то эти учетные записи (а лучше одна учетная запись на один туннель) могут быть также членами домена. Можно порекомендовать выбирать сложные пароли и установить опции, указывающие на то, что пользователю не позволено менять пароль самостоятельно, а срок действия последнего не истечет никогда.

Другим вариантом может быть размещение сервера VPN за пределами внутренней сети, с внешней стороны защитного экрана. С точки зрения маршрутизации такая схема абсолютно рабочая. Напомню, что при маршрутизации поступающей дейтаграммы службой RRAS операционной системы Windows 2000 логический интерфейс DOD VPN рассматривается как обычный сетевой интерфейс, который, соответственно, способен передавать и получать дейтаграммы. А тот факт, что они при этом будут зашифрованы, не играет роли. То есть в такой конфигурации сервер VPN формально не отличается от некоего маршрутизатора с двумя и более портами, который знает маршруты в определенные сети. Недостаток такого размещения серверов очевиден — они становятся доступны из сети Интернет и нужно будет либо настраивать фильтры на граничном маршрутизаторе, либо настраивать фильтры в самой операционной системе Windows 2000 (кстати, возможности фильтрации в ней впечатляют). Ну и управление сервером усложняется — просто ли получить доступ для полного управления компьютером за пределами защитного экрана?

И последним решением может быть размещение сервера VPN прямо на защитном экране — такую конфигурацию, например, поддерживает продукт Microsoft ISA Server. Схема очень удобная и устраняет необходимость в дополнительных серверах — действительно, нужно будет приобрести только защитный экран, который параллельно станет выполнять и роль сервера VPN. В такой схеме вовсе не обязательно задействовать протокол PPTP, так как многие современные защитные экраны включают в себя поддержку создания подобных конфигураций на базе других протоколов — например протокола IPSec.

Остановимся отдельно на таком важном факторе, как надежность виртуального соединения. Естественно, она ниже, чем в случае использования выделенных цифровых каналов связи, созданных либо самостоятельно, либо полученных от провайдера услуг. Мониторинг туннеля между реально работающими объектами показал, что его надежность составляет 98% или, если приводить абсолютные цифры, то из 449 часов его работы общее время простоя составило порядка 6 часов (на момент написания этих строк). В общем, при рабочем доступе в сеть Интернет на обоих концах требуется порядка 2–3 минут для восстановления соединения, если оно было потеряно по каким либо причинам. Наиболее распространенной причиной для разрыва связи между серверами по протоколу PPTP является резкое увеличение задержки при передаче информации на промежуточных узлах в сети Интернет. В этом случае серверы теряют связь и сразу же начинают процедуру повторного соединения. Ниже приведен отчет системы мониторинга соединения, которое выполняет обычную команду ping с интервалом в две минуты:

```
Mon, Nov 19, 2001, 18:54:30
State Change: Device Down / Verifying (удаленный сервер не отвечает на ping)
Detail: Missing reply from Ping
Mon, Nov 19, 2001, 18:56:31
State Change: Device Down / Verifying (удаленный сервер не отвечает на ping)
Detail: Missing reply from Ping
Previously device was down / Verifying
Mon, Nov 19, 2001, 18:58:32
State Change: Device Up (удаленный сервер ответил)
Previously device was down / Verifying
```

Видно, что потребовалось немного времени для восстановления соединения (в данном конкретном случае 4 минуты). К сожалению, сбои в работе виртуального соединения зависят от множества факторов, которые, как правило, не поддаются контролю. В иной ситуации надежность соединения может быть очень высокой или, наоборот, очень низкой, если между точками в сети Интернет есть ненадежные или сильно загруженные маршрутизаторы. Качество связи между конечными точками можно проверить с помощью команды traceroute (оценить задержки на каждом переходе).

ИЗ ОПЫТА

Практика показала, что иногда при попытке передать очень большие объемы данных через туннель PPTP (скажем 1–2 Гбайт на высокой скорости — 2 Мбит/с), серверы VPN перестают отвечать на запросы и для возвращения их в рабочее состояние требуется перезагрузка. Кроме того, подобное поведение проявляется при продолжительной работе (2–3 месяца без перезагрузки). Не исключено, что проблема будет решена в очередном пакете исправлений от компании Microsoft, ну а пока самым простым решением является периодическая перезагрузка сервера, например в выходные дни ночью. Если сервер отвечает только за поддержание одного или нескольких туннелей, то перезагрузка будет абсолютно незаметна для пользователей, ведь туннели восстановятся автоматически. Следует оговориться, что описанная проблема типична для сервера, на котором «сходятся» много туннелей (более 10), в то время как серверы на удаленных площадках, поддерживающие единственный туннель, работают без нареканий (рекордное время работы без перезагрузки и вообще административного вмешательства 10 месяцев).

Анализ трафика PPTP на сетевом уровне

Для перехвата трафика, относящегося к туннелю PPTP, можно воспользоваться, например, следующей командой, которая запускает программу WinDump (глава 4) с ключами, указывающими на то, что размер перехватываемых данных равен 1514 байт (-s 1514). При этом пакеты перехватываются целиком и записываются в файл PPTP.log (параметр -w c:\temp\PPTP.log). И последний параметр предписывает перехватывать пакеты, адрес отправителя или получателя которых совпадает с x.x.x.x (ключ host, за которым должен следовать адрес).

```
windump -s1514 -w c:\temp\PPTP.log host x.x.x.x
```

К сожалению, подобный способ перехвата не исключает попадание в журнал «лишних» пакетов, но гарантирует то, что ни один пакет, принадлежащий туннелю PPTP, не будет не замечен. Другим, более тонким вариантом будет указание того, что нужно перехватывать пакеты управляющего соединения (TCP, 1723) и пакеты GRE.

Разберем в деталях, как происходит инициализация туннеля между двумя серверами, его работа и завершение. Кстати, это относится и к случаю взаимодействия клиента с сервером. Для данных целей был проделан простой эксперимент — один из серверов-партнеров принудительно перезагружался и затем автоматически устанавливал туннель. Параллельно происходил перехват трафика, отражающего взаимодействие между серверами-партнерами. Адрес сервера VPN, который оставался активным в процессе эксперимента — 192.168.3.252 (объект 1 в рассматриваемых примерах), а адрес перезагружаемого сервера — 192.168.1.253 (объект 3).

Первоначально, серверы-партнеры должны установить между собой управляющее соединение по протоколу TCP. Для этого они обмениваются друг с другом пакетами TCP с установленными флагами SYN и ACK (рис. 10.20). Первый пакет с опцией SYN (запрос на установление соединения) поступает от сервера VPN на объекте 3, имеющего адрес 192.168.1.253. Обратите внимание, что предлагается использовать технологию SACK (выборочное подтверждение — главу 5). Так как в рассматриваемой сети установлены защитные экраны, то пакет отправляется по адресу внешнего интерфейса защитного экрана (193.125.203.1) на объекте 1 и порт назначения 1723 (анализатор подставляет ключевое слово PPTP). Сервер-партнер по туннелю PPTP посылает в ответ пакет TCP с установленными флагами SYN и ACK и соглашается на использование опции SACK (выборочное подтверждение).

После того как соединение было установлено, сервер на объекте 3 отправляет по нему запрос START-CONTROL-REQUEST протокола PPTP (рис. 10.21). А в ответ на него соответственно посылается START-CONTROL-REPLY. Эти запросы служат для начала согласования управляющего соединения TCP (не путать с логическим соединением, которое к этому времени уже было установлено).

Обмен этими сообщениями, согласно стандарту на протокол PPTP, призван согласовать версию протокола используемого управляющего соединения между серверами (в нашем случае версия 1 — поле Protocol version). Само поле содер-

жит старший и младший номера версии протокола (версия 1, ревизия 0). Другие поля в этом сообщении практически не несут никакой особой смысловой нагрузки, за исключением, может быть, поля Cookie, которое всегда заполняется одним и тем же значение — 0x1a2b3c4d и служит для гарантии правильной синхронизации потока данных протокола TCP. В ответ на рассматриваемое сообщение посылается другое, которое содержит аналогичные поля и служит для подтверждения готовности начать установление туннеля (сообщение START-CONTROL-REPLY).

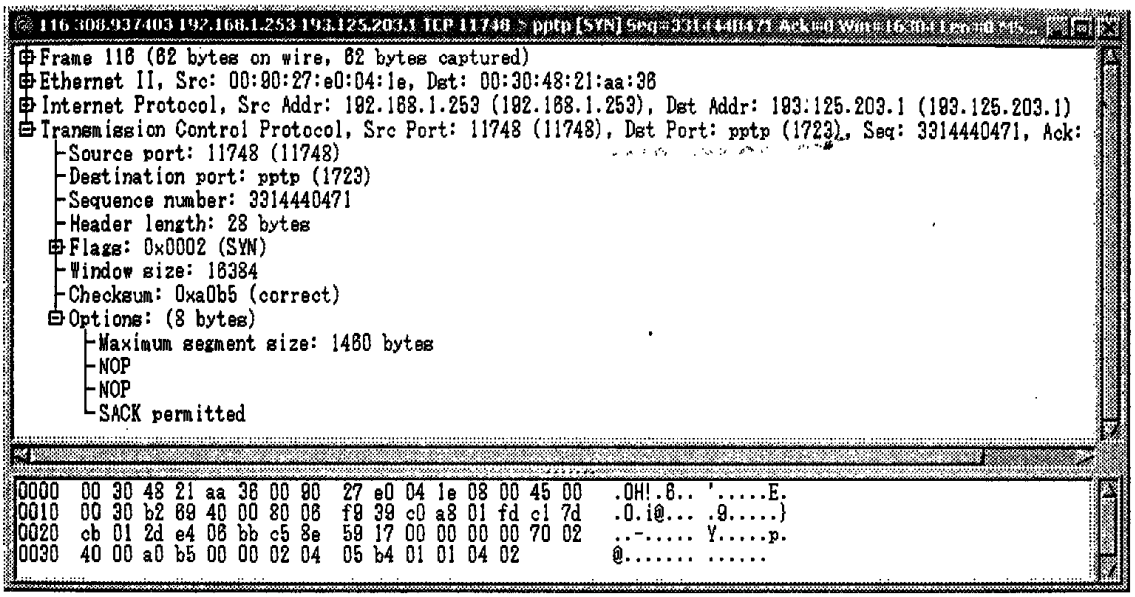


Рис. 10.20. Запрос на установление соединения TCP

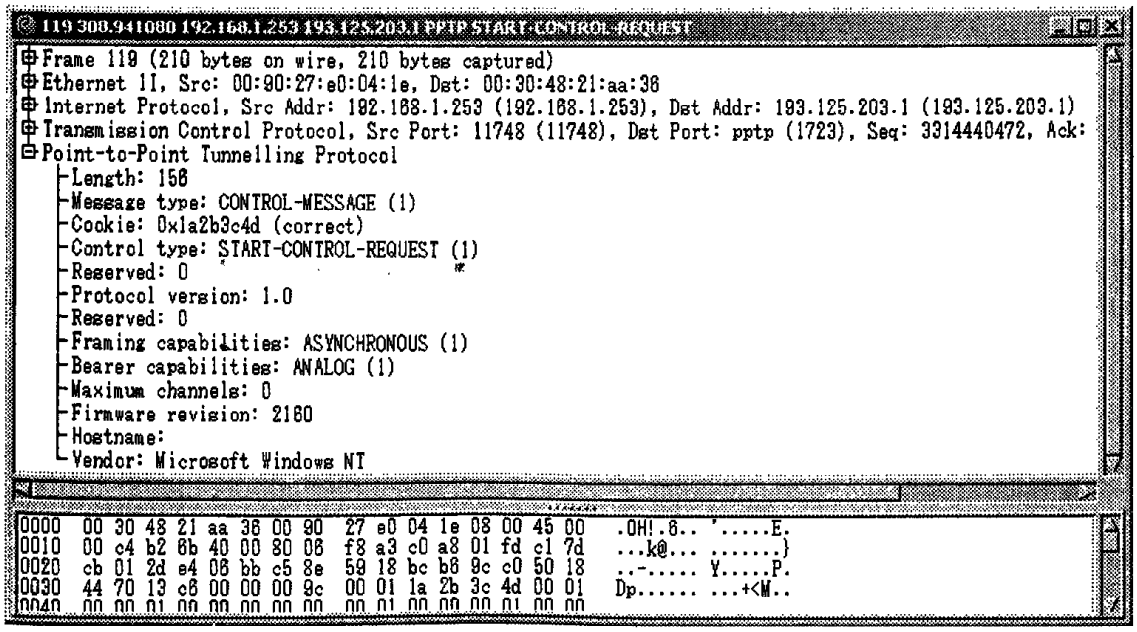


Рис. 10.21. Сообщение START-CONTROL-REQUEST

Следующее сообщение от удаленного сервера, после того как он убедился, что его партнер готов к продолжению, гораздо более информативно. Сервер VPN на объекте 3 отправляет сообщение OUTGOING-CALL-REQUEST, которое указывает на то, что удаленный сервер начинает процедуру установки самого туннеля PPTP (рис. 10.22).

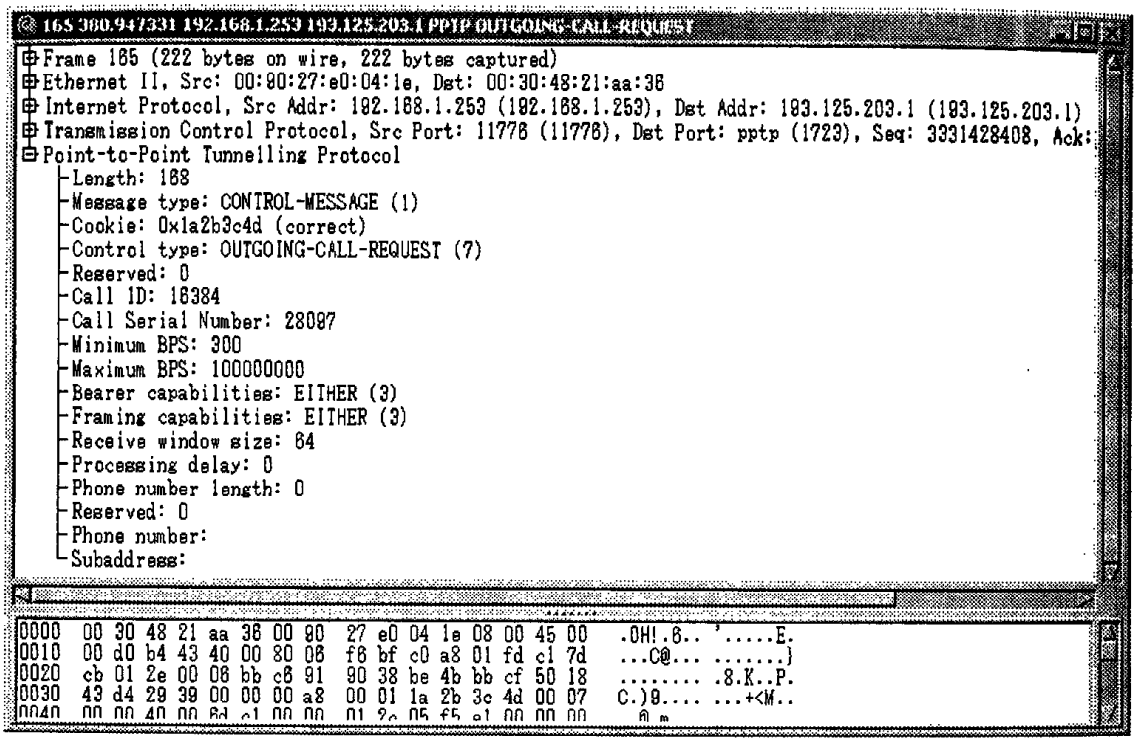


Рис. 10.22. Сообщение OUTGOING-CALL-REQUEST

В этом сообщении интересно такое поле, как Call ID (16384), содержащее уникальный идентификатор сеанса между серверами и использующееся для мультиплексирования и демультиплексирования данных, посылаемых по туннелю. Поле Call Serial Number (28097) содержит другой идентификатор сеанса, применяемый для различения разных сеансов при обмене. В отличие от предыдущего поля, это поле будет одинаково для обоих серверов. Удаленный сервер также предлагает размер окна (поле Receive window size) на принимающей стороне. В ответ на сообщение OUTGOING-CALL-REQUEST посылается OUTGOING-CALL-REPLY, которое обобщает результаты обработки предыдущего запроса (рис. 10.23).

Ответ дополнительно включает в себя сведения, позволяющие регулировать передачу информации для устанавливаемого сеанса. Учитывая, что поле Call ID уникально, ему присваивается уникальное число (в примере 33761). А вот поле Peer's call ID наследует значение поля Call ID (16384) из запроса OUTGOING-CALL-REQUEST, посланного первым сервером. Поле Result, как можно понять из названия, содержит результат попытки установить управляющего сеанса — в данном случае попытка была успешной, и в ответ посылается значение 1 — CONNECTED. В стандарте предусматривается 7 возможных сообщений (6 из них — сообщения об ошибках). Поле Connect speed содержит актуальное значение ско-

рости соединения в битах в секунду (порядка 14,1 Мбит/с). Поле Receive window size (размер приемного окна) указывает число пакетов, которые будут помещаться в буфер для этого сеанса.

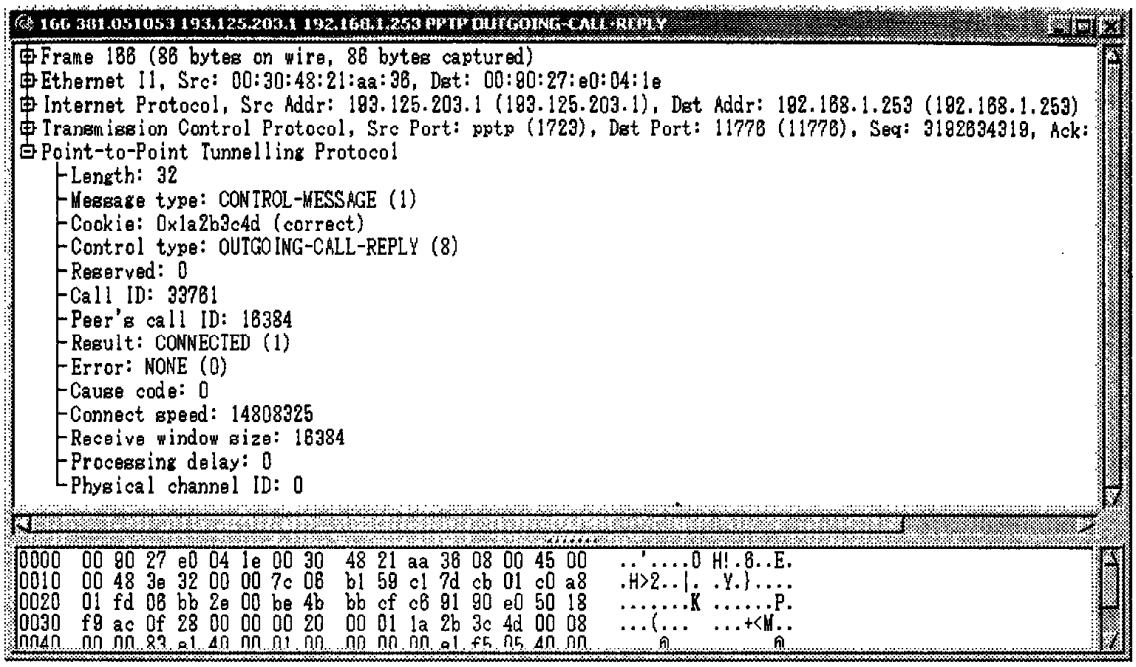


Рис. 10.23. Сообщение OUTGOING-CALL-REPLY

Следующим этапом создания соединения является проверка имени пользователя и пароля. Говоря о туннеле PPTP, следует помнить, что он состоит как бы из двух соединений, каждое из которых устанавливается сервером. Например, в рассматриваемом примере в создании туннеля участвуют два сервера, на каждом из них при настройке службы RRAS (а точнее при создании интерфейса DOD VPN) указываются имя и пароль пользователя. Впоследствии, когда сервер начинает процедуру формирования туннеля, он отправляет удаленному партнеру информацию об имени пользователя и его пароле. На сервере-партнере такой пользователь должен существовать, и он должен иметь право удаленного доступа. Эту процедуру выполняет каждый сервер-участник туннеля. В том случае, если оба сервера входят в домен, можно использовать одну доменную учетную запись.

Очередной шаг — проверка имени пользователя и пароля (только начинается). В нашем случае сервер VPN на объекте 3 первым начинает процедуру «постройки» своей части туннеля. Первенство ему отдано по той причине, что он перезагружался, а при запуске служба RRAS сразу пытается установить настроенные туннели, в то время как удаленный сервер на объекте 1 после того, как туннель оборвался, делает попытку восстановить его через небольшие периоды времени — 1–2 минуты (этот временной интервал может быть настроен).

При настройке туннеля администратору предлагается несколько схем аутентификации удаленного партнера. В сети Интернет есть публикация, которую я как автор настоятельно рекомендую прочитать (http://www.counterpane.com/pptpv2-paper.html, автор Брюс Шнайер (Bruce Schneier)) — она рассматривает

слабые стороны протокола PPTP, а если говорить точнее, то «узкие места» протокола MS-CHAP v1 (версии 1). Документ датирован 1998 годом, и в ответ на его появление компания Microsoft представила новую версию этого протокола MS-CHAP v2. Впрочем, следует отметить, что до этого момента версии как таковой и не было — был просто протокол MS-CHAP, считавшийся расширенным вариантом известного протокола CHAP, предложенного Microsoft. Новая редакция протокола похожа на первую версию, но не совместима с ней — назначение некоторых полей изменилось или от их использования было решено отказаться вовсе. Кроме того, сам протокол стал гораздо более устойчивым к различным методам его дискредитации и, очевидно, должен использоваться там, где это возможно. Учитывая вышесказанное, не будем останавливаться на обсуждении протокола MS-CHAP как не совсем удачного выбора в качестве средства аутентификации партнеров по туннелю, а сразу перейдем к рассмотрению второй версии этого протокола, отмечая только радикальные отличия между этими двумя версиями протоколов.

В процессе формирования туннеля, после того как было установлено управляющее соединение и серверы обменялись сообщениями OUTGOINT-CALL-REQUEST и OUTGOINT-CALL-REPLY, начинает работать процедура согласования параметров соединения протокола PPP. Для этих целей используется протокол PPP LCP (Link Control Protocol — протокол управления соединением). Основная задача данного протокола — согласовать различные параметры, которые будут затем использоваться в соединении PPP (напомню, что туннель PPTP в своей работе полностью опирается на протокол PPP, пакеты которого просто инкапсулированы в дейтаграммы IP). При согласовании параметров серверы обмениваются между собой несколькими пакетами PPP LCP, в одном из которых указывается и протокол аутентификации, который предполагается задействовать. Сервер VPN с адресом 192.168.1.253 посылает своему партнеру по туннелю запрос (PPP LCP Configuration Request) с предложением использовать алгоритм MS-CHAP v2 для аутентификации (рис. 10.24).

Поскольку спецификация PPTP опирается на возможности протокола PPP, разработчикам из Microsoft практически не пришлось «изобретать велосипед». Гораздо проще задействовать имеющиеся и апробированные механизмы, каковым является протокол PPP, и дополнительные возможности, такие как проверка имени и пароля (метод PAP или CHAP). Однако первоначально протокол PPP позиционировался для работы на канальном уровне модели OSI, в то время как для создания туннеля требуется взаимодействие на сетевом уровне и выше, или, иными словами, необходимо передавать пакеты PPP, инкапсулируя их в дейтаграммы IP. Для этих целей привлекается протокол GRE, который просто предоставляет каркас для любых видов инкапсуляции, в том числе и пакетов PPP, в дейтаграммы.

Протокол аутентификации MS-CHAP v2 активируется указанием числового значения 0x81 в поле Algorithm (алгоритм) опций заголовка протокола PPP LCP. В остальном конфигурационные параметры для протокола MS-CHAP v2 идентичны тем, которые используются при согласовании стандартного протокола CHAP (для него поле Algorithm будет иметь значение 0x05, то есть пароль будет проверяться по алгоритму MD5). В приведенном на рис. 10.24 перехваченном

пакете один из серверов посылает запрос PPP LCP с предложением использовать протокол MS-CHAP v2 для аутентификации. Если другая сторона соглашается с предложением, то в ответ посылается подтверждение (сообщение PPP LCP Configuration Ack). В рассматриваемом случая оба сервера настроены на поддержку этого протокола и запрос был успешен.

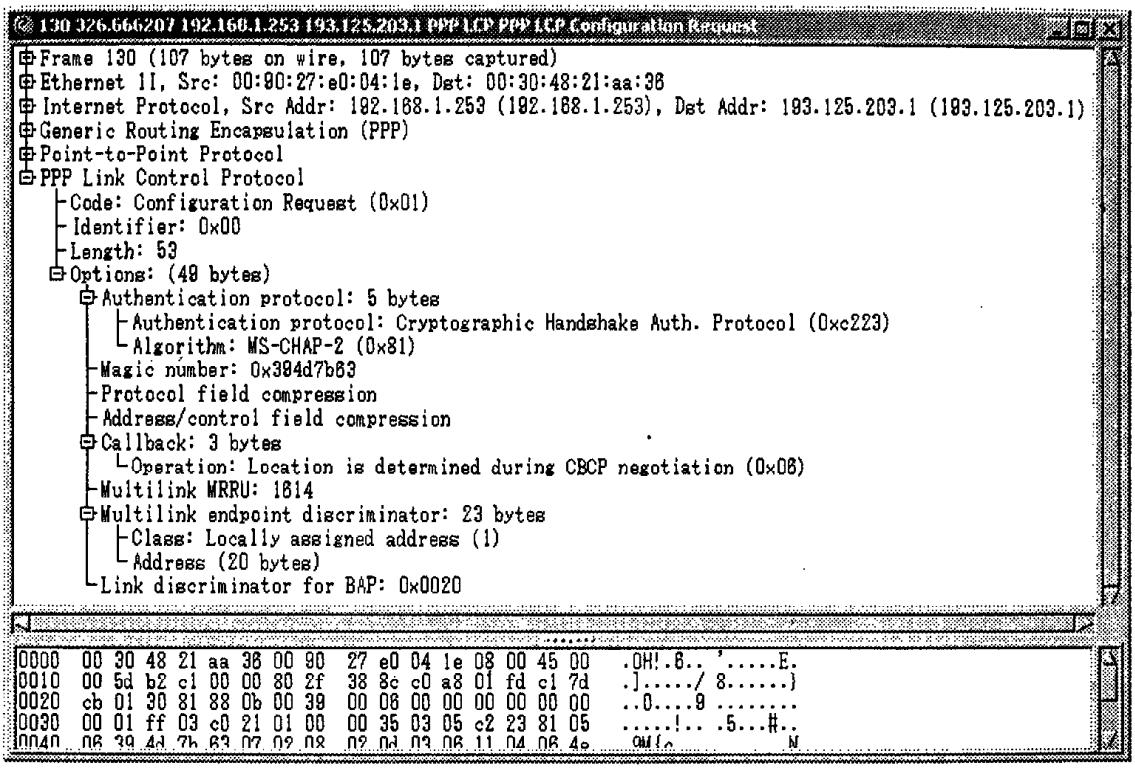


Рис. 10.24. Запрос на согласование схемы аутентификации

Далее наступает очень важный момент в процедуре формирования туннеля — проверка правильности имени и пароля удаленного клиента. Сейчас рассматривается конфигурация, когда туннель устанавливается между двумя серверами, но при этом важно помнить, что туннель является двунаправленным — то есть каждый сервер по очереди выступает как клиент и как сервер в процессе инициализации туннеля. По этой причине все сказанное будет верно и для ситуации, когда удаленный пользователь пытается установить туннель с сервером в своей сети для работы с внутренними ресурсами.

После того как стороны согласовали протокол аутентификации (в нашем случае MS-CHAP v2), происходит обмен пакетами PPP CHAP, который помогает удостовериться, что пробующий установить туннель сервер (или в терминологии алгоритма MS-CHAP — клиент; в нашем случае туннель пытается организовать сервер VPN на объекте 3 — он и будет клиентом) имеет необходимые для этого полномочия. В случае туннеля PPTP сервер должен проверить имя и пароль удаленного клиента, убедиться, что таковой зарегистрирован в домене (Active Directory — если сеть построена на базе Windows 2000) и имеет полномочия устанавливать туннель. Так как пароль нельзя передавать по сети в открытом виде

(хотя при установления туннеля он задается именно так) используются так называемые «знаки внимания» или «вызовы» (challenges). На первом шаге клиент запрашивает «вызов» от сервера и сервер посылает его в виде 16-байтовой числовой последовательности (рис. 10.25).

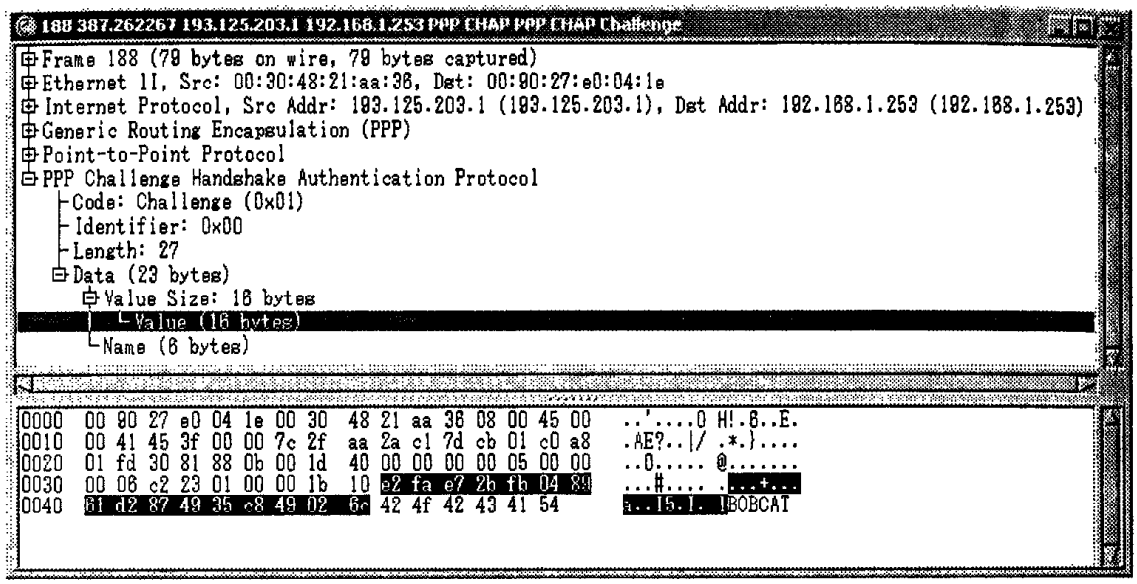


Рис. 10.25. Пример пакета, содержащего «вызов»

В перехваченном в сети пакете этот «вызов» представляет собой набор 0xe2fae72bfb048961d2874935c849026c. Эта последовательность должна быть случайной. Обратите внимание, что в пакете есть поле Name (8 байт) — оно содержит имя сервера VPN на объекте 1 (BOBCAT). Клиент получает «вызов» и, в свою очередь, формирует новую 16-байтовую последовательность, которая в стандарте называется Peer Authenticator Challenge (будем далее ссылаться на нее как на PAP). Далее он формирует другую цепочку символов длиной 8 байт с помощью двух перечисленных «вызовов» и имени пользователя, под которым выполняется попытка установить туннель. Указанные 8 байт формируются с помощью процедуры хэширования (hashing). Конкретно происходит следующее — объединение двух 16-байтовых «вызовов» и имени пользователя (в шестнадцатеричном виде), после чего результат «проходит» через функцию SHA-1 и первые 8 байт результата становятся 8-байтовой последовательностью, которая затем применяется для шифрования.

НА ЗАМЕТКУ

В документе RFC 2759 от января 2000 года отмечено, что в настоящее (в то) время продукты компании Microsoft не заполняют поле Name — и это может измениться в будущем — «Microsoft authenticators do not currently provide information in the Name field. This may change in the future». Видимо, это будущее уже настало.

Затем клиент (сервер VPN на объекте 3) выполняет процедуру шифрования пароля пользователя с помощью этой 8-байтовой последовательности (для чего

вызывается стандартная функция хэширования, принятая в Microsoft Windows NT). В результате получается 24 байт данных отклика (response), которые отправляются удаленному серверу (рис. 10.26). Точнее, посылается такая строка: 16 байт случайной последовательности, сгенерированной клиентом (PAP); 8 байт, заполненных нулями (на будущее расширение); 24 байт — сформированный «ответ» и 1 байт — флаг (зарезервировано на будущее — пока равен нулю).

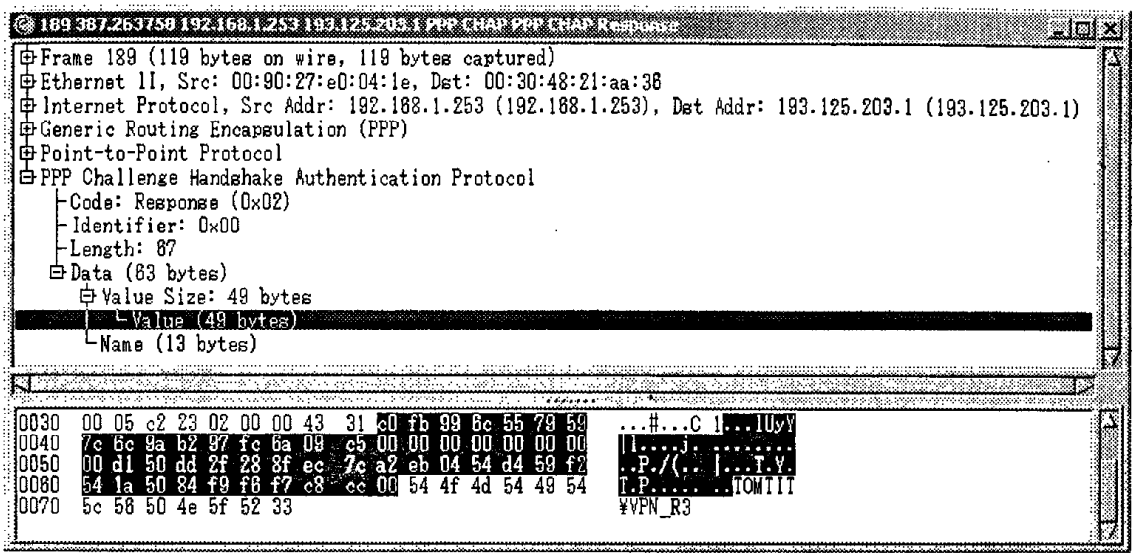


Рис. 10.26. Пример пакета, содержащего «ответ»

Наибольший интерес вызывает поле Name (имя), которое в соответствии со стандартом на протокол MS-CHAP v2 является строкой длиной до 255 символов, идентифицирующей пользователя (VPN_R3). Если учетная запись входит в домен Microsoft Windows, то его имя указывается как префикс (TOMTIT\), а в противном случае имя домена может быть опущено. Видно, что на данном этапе уже известно имя удаленного сервера (BOBCAT), что нельзя признать очень полезным для атаки, но что более важно — известно имя пользователя и домена, где он зарегистрирован. Теперь осталось только узнать пароль и можно сказать, что туннель надежным являться не будет.

Сложно объяснить, почему во второй версии протокола MS-CHAP реализована столь сложная схема формирования «ответа». Видно, что здесь не востребован (по крайней мере явно) 16-байтовый «вызов» сервера, а вместо этого генерируется новая случайная последовательность (тоже 16 байт — в нашем случае 0xc0fb996c5579597c6c9ab297fc6a09c5), которая, тем не менее, также посылается по сети. Из перехваченного пакета нам уже известен 24-байтовый «ответ» — 0xd150dd2f288fec7ca2eb0454d459f2541a5084f9f6f7c8cc. Как это значение получается? Процедура практически одинакова для двух версий протокола MS-CHAP — сервер берет хэшированное значение пароля пользователя (16 байт) и заканчивает эту последовательность нулями до получения 21 байт (добавляется пять нулевых байтов). Далее, новая 21-байтовая последовательность (обозначим ее как *Z*) разбивается на 3 блока по 7 байт, каждый из которых затем используется как ключ для шифрации 8 байт, полученных ранее с помощью алгоритма SHA-1.

Хэшированная версия пароля пользователя производится следующим образом — пароль перекодируется в Unicode с сохранением регистра, а затем к нему применяется функция MD4.

Подведем итог, что может быть известно злоумышленнику на данном этапе. Можно легко рассчитать 8-байтовую последовательность, которая затем участвует в шифровании — для этого подойдет любой калькулятор хэш-значений (например с сайта http://www.slavasoft.com). Вся необходимая информация уже есть (обозначим эту последовательность *X*) — в том числе имя пользователя. Далее, известно, что последовательность *X* используется для шифрования по алгоритму DES пароля пользователя (точнее, хэш-значения пароля) и нам известен результат — 24 байт (назовем это *Y*). *Y* формируется посредством применения функции DES к последовательности *X*, где ключом для шифрования служат 7-байтовые блоки из последовательности *Z*. В последнем блоке, как было отмечено, только первые два байта неизвестны — остальные равны нулю. То есть, если говорить о взломе, то, предполагая, что пароль представляет собой некую угадываемую комбинацию (значит, возможна атака по словарю), нужно перебрать 2^{16} возможных вариантов битов) комбинаций для получения последних двух байтов хэшированного значения пароля пользователя. Далее можно воспользоваться большим словарем, вычислить для каждого слова хэш-последовательность, а затем получить список тех, у которых последние два байта совпадают с вычисленными. Такой подход позволит значительно сократить размер словаря и повысить скорость вэлома (и надеюсь, что осознание его легкости подтолкнет вас принять дополнительные меры безопасности, по крайней мере, в отношении сложности пароля).

Вернемся к происходящему на сетевом уровне. После получения от клиента ответа (собственно, сервер интересует только последовательность *Y*), он, опираясь на хранящееся у него хэш-значение пароля пользователя, выполняет процедуру дешифрации *Y*. Если дешифрованные блоки совпадают с «вызовом», то считается, что клиент успешно прошел аутентификацию. Основное отличие двух версий алгоритмов MS-CHAP заключается в том, что версия 2 использует хэш-значения паролей Windows NT, в то время как первая — LAN Manager и Windows NT. Эти две разновидности отличаются друг от друга и первая гораздо менее устойчива к взлому, чем вторая.

В том случае, если клиент прошел процедуру аутентификации, сервер отправляет сообщение PPP CHAP Success, которое содержит так называемый «ответ аутентификатора» (Authenticator Response), размером 42 байт. Формат этого сообщения следующий — S=<auth_string> (см. выделенную часть рис. 10.27). Ответ представляет собой 20-байтовую последовательность (число, обозначим его как *AR*), кодированное в соответствии со стандартом ASCII как 40 шестнадцатеричных цифр.

Клиент получает последовательность *AR*, вычисляет ее самостоятельно и затем производит сравнение для того, чтобы завершить процедуру аутентификации. При расчете клиент выполняет повторное хэширование хэш-значения пароля (назовем это PHH — password hash-hash). Далее, клиент конкатенирует значение PHH с последовательностью *Y* (24 байт) и со строкой «Magic server to client constant» и затем подвергает результат обработке алгоритмом SHA-1. Полученный результат (20 байт — на выходе этого алгоритма хэширования всегда

последовательность такой длины) объединяется с последовательностью *X* (8 байт) и со строкой «Pad to make it do more than one iteration» и затем вновь поступает на вход алгоритму SHA-1, формируя последовательность для сравнения с *AR*. Трудно объяснить, зачем разработчики настолько усложнили алгоритм — дважды задействуется SHA-1.

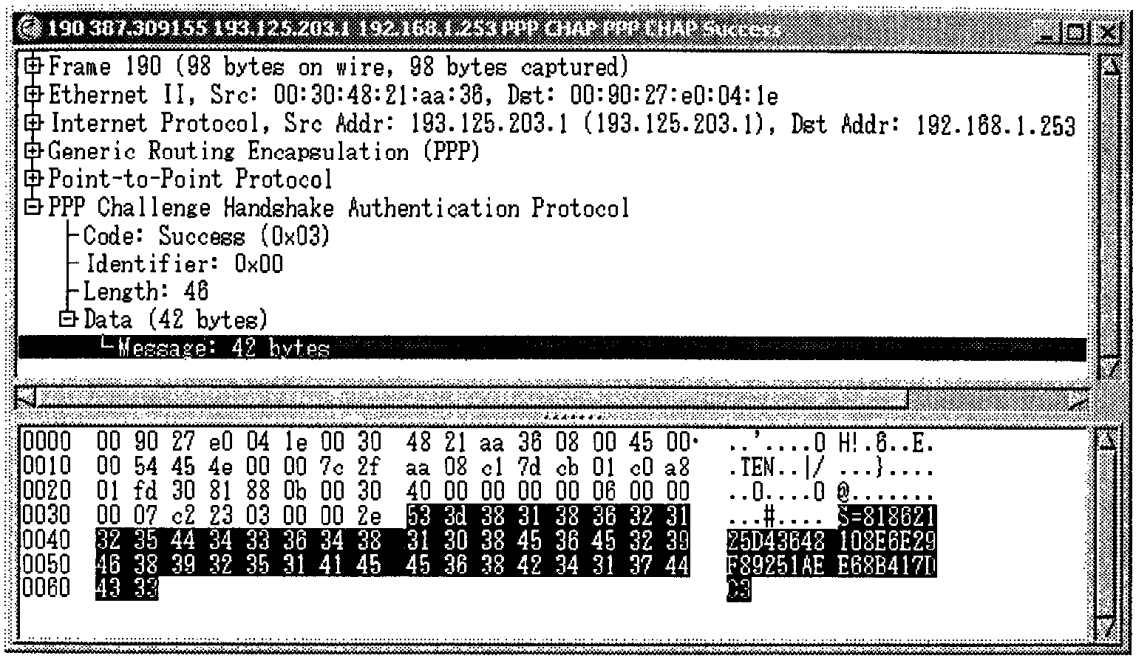


Рис. 10.27. Пакет, подтверждающий успешность аутентификации

Ранее уже упоминалось, что в шифровании данных, передаваемых по туннелю, участвуют технология MPPE и алгоритм RC4. Особенностью является то, что ключи для шифрования данных получаются из пользовательских данных, согласованных алгоритмом MS-CHAP v2, и в каждом направлении работает свой, уникальный ключ, хотя и получаются они из одних и тех же данных (собственно хэшированное значение пароля пользователя). При использовании алгоритма MS-CHAP v1 ключ формировался следующим образом — к хэшированному паролю применялась функция SHA-1, а затем результат обрезался в зависимости от длины ключа. При согласовании с помощью ключа длиной 40 бит результат функции укорачивался до 64 бит, а после первые 24 бит устанавливались в значение 0xd1269e, а для ключа длиной 128 бит результат функции усекался до 128 бит. Полученный ключ и использовался для шифрования трафика в обоих направлениях. Отметим, что ранние версии протокола MPPE для хэширования опирались на алгоритм MD4 (как и указано в стандарте RFC 3079), но потом было предложено задействовать алгоритм SHA-1.

В случае согласования алгоритма MS-CHAP v2 процедура получения ключей усложняется. Выполняется хэширование последовательности, состоящей из хэш- значения пароля (16 байт), последовательности *Y* (24 байт) и константы (строки «This is the MPPE Master Key» — 27 байт) с помощью алгоритма SHA-1. Результат укорачивается до 16 байт. Это так называемый master-master-ключ

(мастер-ключ). Далее, если согласован ключ длиной 40 бит, то на вход алгоритма хэширования поступает следующая последовательность: мастер-ключ; 40 байт, заполненных нулями; некая «магическая» константа длиной 84 байт и 40 байт, заполненных значением 0xF2. Из результата (20 байт) берется 8 первых байтов (64 бит), начальные 24 бит которых устанавливаются в значение 0xd1269e — это и будет ключом длиной 40 бит. Значение «магической» константы зависит от того, используется ли ключ для шифрования трафика от клиента к серверу, или наоборот. Ключ длиной 128 бит получается аналогично, за исключением того, что берется не 8, а 16 байт результата функции хэширования. Видно, что хотя процедура формирования ключей усложнилась, по сравнению с алгоритмом MS-CHAP v1, но все равно в основе лежит хэшированное значение пароля пользователя.

На практике, скорее всего, при формировании туннелей PPTP будет применяться ключ длиной 128 бит и алгоритм MS-CHAP v2, как обеспечивающие наивысший уровень безопасности передаваемых данных. Использование разных ключей для двух направлений передачи данных также помогает значительно повысить устойчивость к перехвату данных и последующим попыткам взлома. В любом случае, надежность туннеля будет определяться тем, насколько сложным и трудно предсказуемым (случайным) был выбран пароль.

Вернемся к рассмотрению процедуры установления туннеля, где мы остановились на согласовании параметров протокола IP. Так как туннель является двунаправленным, то каждый из серверов-партнеров выступает в роли сервера и клиента одновременно. Клиенту, как минимум, нужно предоставить адрес протокола IP, который выделяется ему из удаленной сети. Для этого используется протокол PPP IP Control Protocol (контрольный протокол — IPCP). Клиент (в нашем случае сервер на объекте 3) посылает запрос PPP IPCP Request с опциями в виде IP-адресов, которые изначально установлены в значение 0.0.0.0 (рис. 10.28).

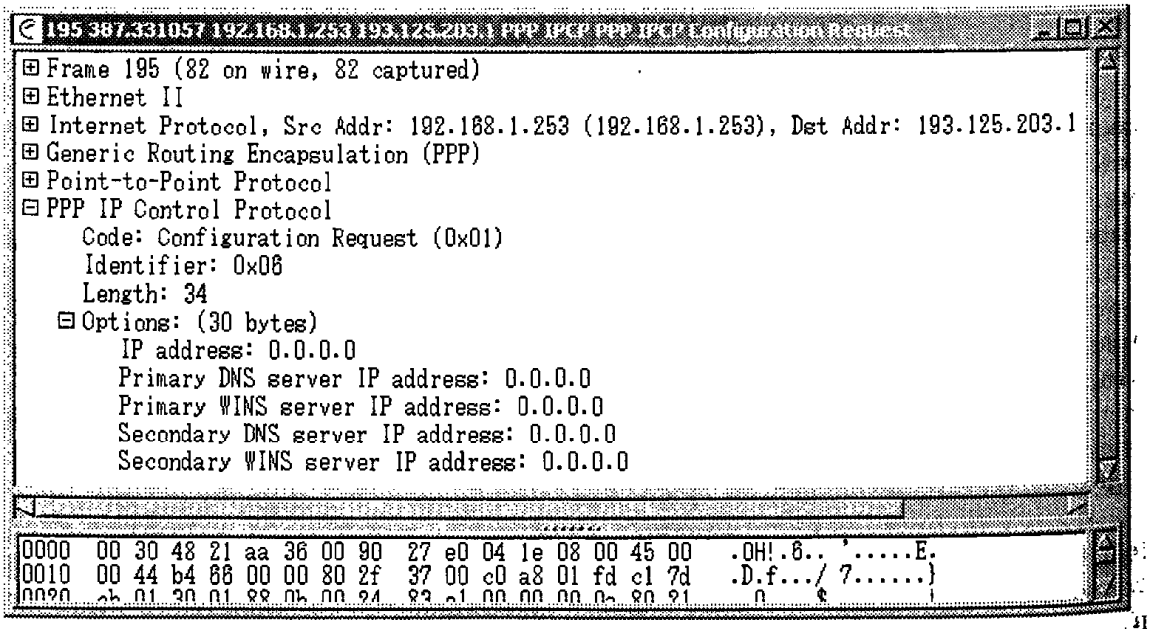


Рис. 10.28. Запрос на получение параметров протокола IP

Этим запросом клиент предлагает серверу установить соответствующие адреса, которые затем будут востребованы. Как видно из рисунка, клиента интересует собственно его адрес в удаленной сети (как правило, он выдается службой RRAS с помощью сервера DHCP), а также адреса серверов DNS и WINS. Для чего это нужно? Самым простым примером может быть ситуация, когда некий пользователь через сеть Интернет устанавливает туннель в корпоративную сеть. Хосту, за которым работает пользователь, присвоен некий адрес IP, выдаваемый обычно провайдером, который, естественно, не имеет ничего общего с адресной схемой, принятой внутри корпоративной сети (например, 192.168.3.0/24). Если хосту не будет назначен адрес внутри сети и другие параметры, то работать пользователь просто не сможет. Поэтому удаленный сервер VPN в ответ на такой запрос посылает аналогичный (IPCP Configuration Ack), но с уже заполненными полями (рис. 10.29).

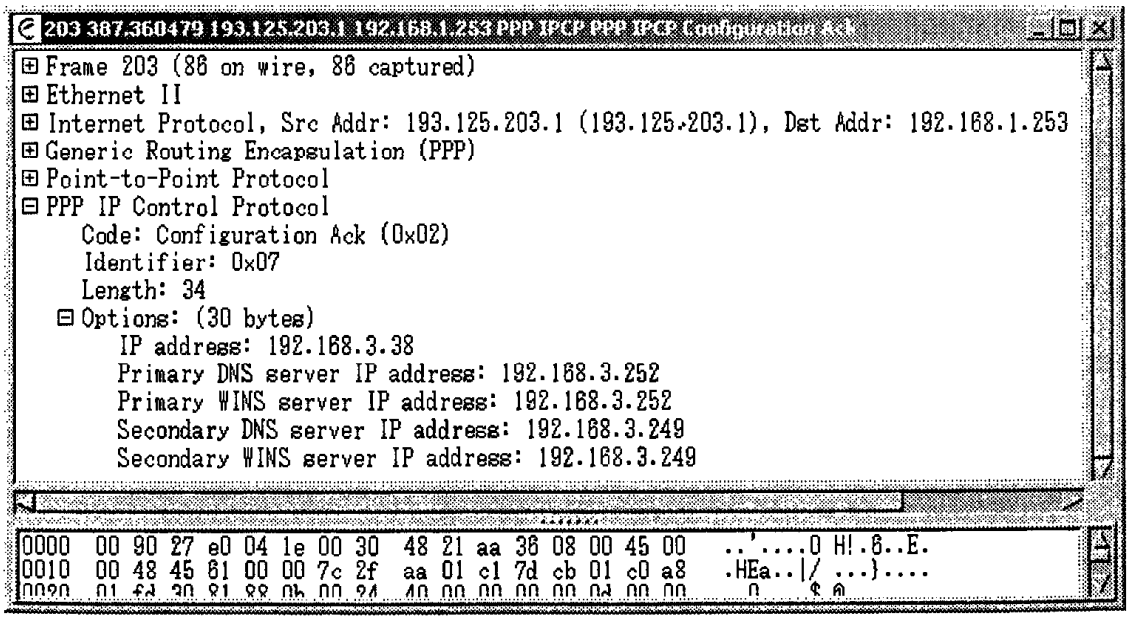


Рис. 10.29. Ответ сервера VPN с предлагаемыми параметрами

Теперь процедуру формирования туннеля можно считать законченной. Отметим, что мы только рассмотрели установление одного направления туннеля: от сервера с адресом 192.168.1.253 к серверу с адресом 192.168.3.252 (в перехватах фигурирует адрес получателя 193.125.203.1 — адрес внешнего интерфейса защитного экрана на объекте 1, который просто транслирует дейтаграммы серверу VPN на этом объекте). На практике серверы чаще всего начинают устанавливать оба направления туннеля одновременно, и, как правило, этот процесс занимает не более 40 секунд (в зависимости от скорости каналов связи). Очень редко наблюдается ситуация, когда один из серверов VPN потерял «свою» часть туннеля, в то время как другая продолжает успешно наблюдать «свет на противоположном конце». Автор сталкивался с такой ситуацией, и для нее характерно

то, что связь с удаленным объектом прерывается (например, хосты перестают отвечать на запросы ping), но в то же время можно получить доступ к удаленной сети непосредственно с локального сервера VPN (то есть с того, который еще поддерживает свое направление туннеля). Самым простым способом исправить ситуацию будет подключение с локального сервера VPN через терминальный доступ к удаленному серверу и принудительная перезагрузка последнего. Такая ситуация в нормальной работе туннеля встречаться не должна и скорее всего характеризует некий внутренний сбой в программном обеспечении.

Пора подвести итог тому, что удалось узнать из перехваченных пакетов (для обоих направлений туннеля) — имена двух серверов-партнеров, имя домена, имя пользователя в домене и адреса серверов DNS и WINS внутри двух сетей. И хотя этой информации будет явно недостаточно для компрометации туннеля, но все же она не может относиться к тому типу информации, которую стоит свободно распространять в сети Интернет. Обратите внимание, что контрольные соединения туннеля не защищаются никоим образом и, как следствие, появляется возможность организовать атаку на контрольное соединение TCP. Также злоумышленник может сгенерировать ложные управляющие сообщения, направленные одному из серверов-партнеров с целью разрушить воздвигнутый туннель.

Уже после того как туннель был установлен, серверы обмениваются между собой специальными пакетами ECHO (некий вариант сообщения keepalive — рис. 10.30). Обмен «эхо» происходит по управляющему соединению и, как можно догадаться из названия, служит для проверки того, что связь работает и туннель функционирует. Так, один из серверов посылает запрос ECHO-REQUEST в надежде, что удаленный сервер отзовется. Если ответа нет (в течение 60 секунд), то туннель считается разрушенным и серверы могут начинать процедуру его установки с самого начала.

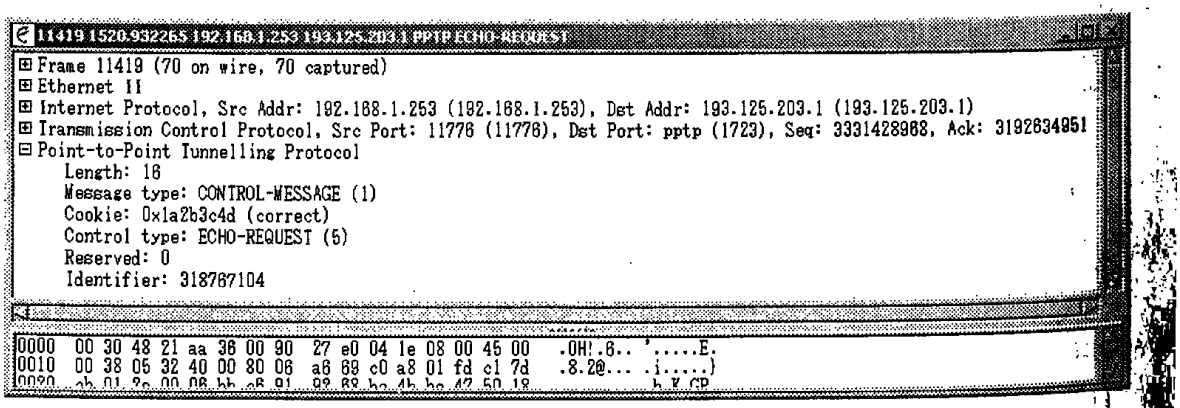


Рис. 10.30. Пример пакета ECHO

В запросе ECHO-REQUEST есть поле идентификатора (Identifier), предназначенное для того, чтобы дифференцировать поступающие ответы на разные запросы — значение данного поля в ответе должно совпадать со значением в запросе.

Процедура настройки туннеля

До этого момента рассматривались детали работы туннеля PPTP, а сейчас остановимся на процедуре настройки туннеля между двумя серверами, участвующими в его поддержании. Для примера покажем как настроить туннель на сервере VPN объекта 1 (адрес 192.168.3.252 на рисунках в этой главе) к одному из удаленных объектов (а точнее к объекту 2 — серверу с адресом 192.168.10.253). Первым этапом является запуск консоли службы RRAS на сервере VPN на объекте 1. Если администратор запускает консоль первый раз, то ему будет предложено воспользоваться мастерами по настройке различных вариантов подключений, в том числе и подключения по виртуальной частной сети. Как показывает практика, лучше выбрать опцию «Ручная настройка» и выполнить все описанные далее шаги самостоятельно.

На первом этапе нужно, перейдя в дерево Routing Interfaces, нажать правую кнопку мыши и выбрать создание нового интерфейса DOD (New Demand-Dial Interface). На экране появится приветствие Мастера создания интерфейса и после нажатия кнопки Далее будет предложено ввести имя интерфейса (рис. 10.31).

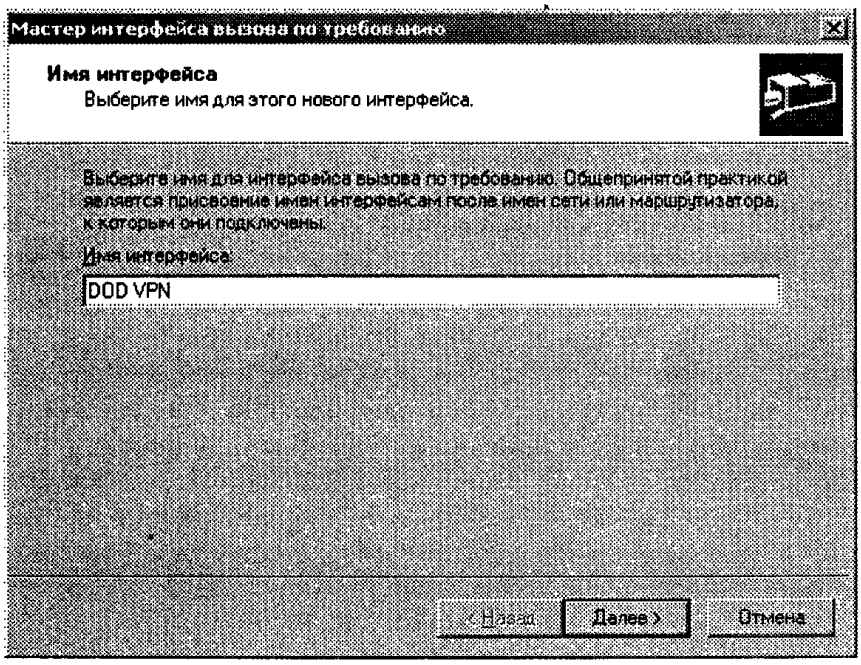


Рис. 10.31. Ввод имени интерфейса DOD

Вообще говоря, имя служит просто для удобства дальнейшего администрирования интерфейса и администратор волен выбрать его любым. Можно только посоветовать использовать такое имя, которое в дальнейшем позволит легко понять, на каком из объектов данный интерфейс будет поддерживать туннель. Далее предлагается выбрать протокол, который будет отвечать за туннелирование трафика. Сейчас мы остановимся на протоколе PPTP (рис. 10.32).

После этого будет предложено ввести адрес сервера (или его имя), с которым будет устанавливаться туннель. Как уже не раз отмечалось в главе, адреса

серверов-партнеров участников туннеля, как правило, являются фиксированными и определяются на этапе планирования сети (рис. 10.33)

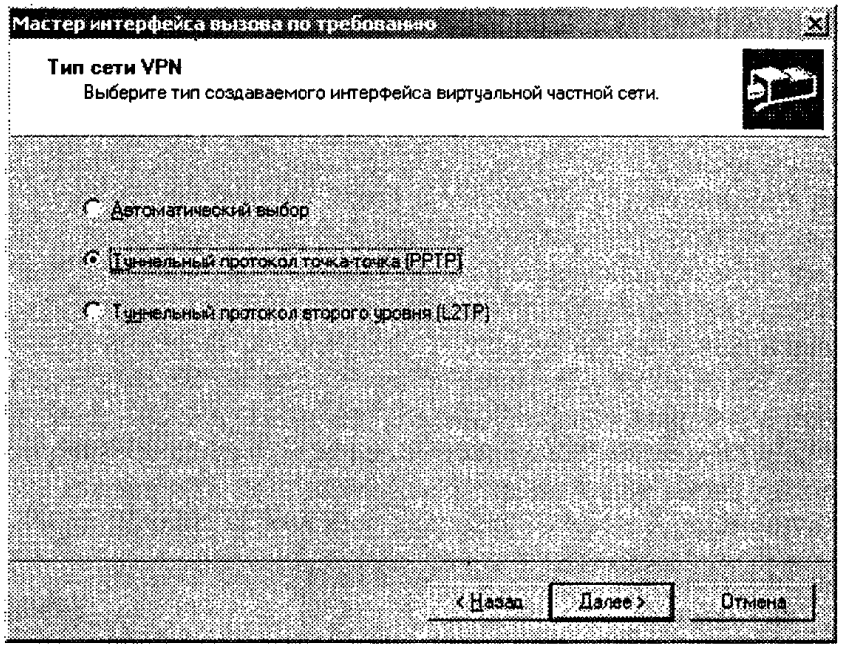


Рис. 10.32. Выбор типа технологии для использования в туннеле

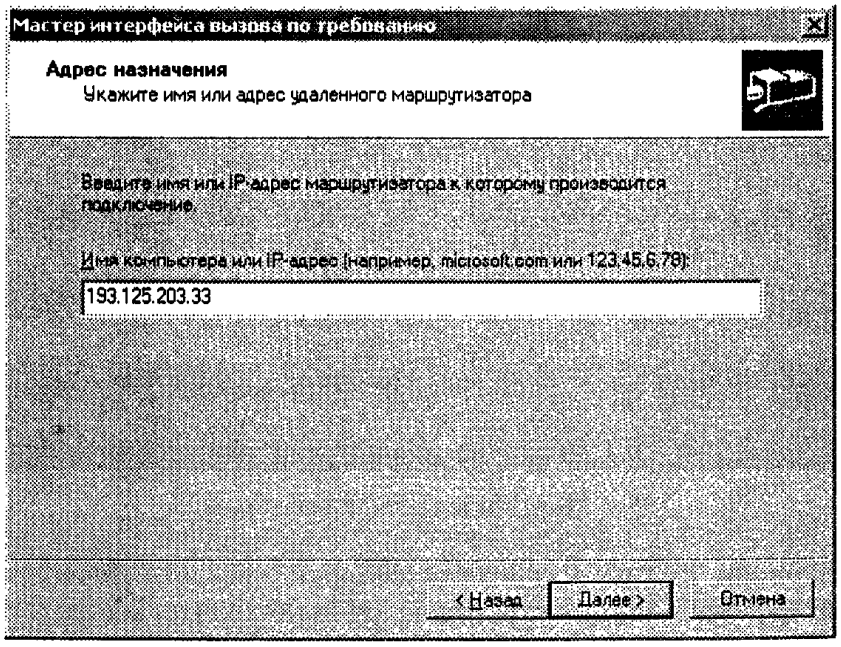


Рис. 10.33. Ввод адреса сервера-партнера по соединению

Далее предлагается выбрать опции маршрутизации (транспортных протоколов). Для настройки туннеля можно принять настройки по умолчанию («Перенаправлять пакеты IP на этот интерфейс») (рис. 10.34).

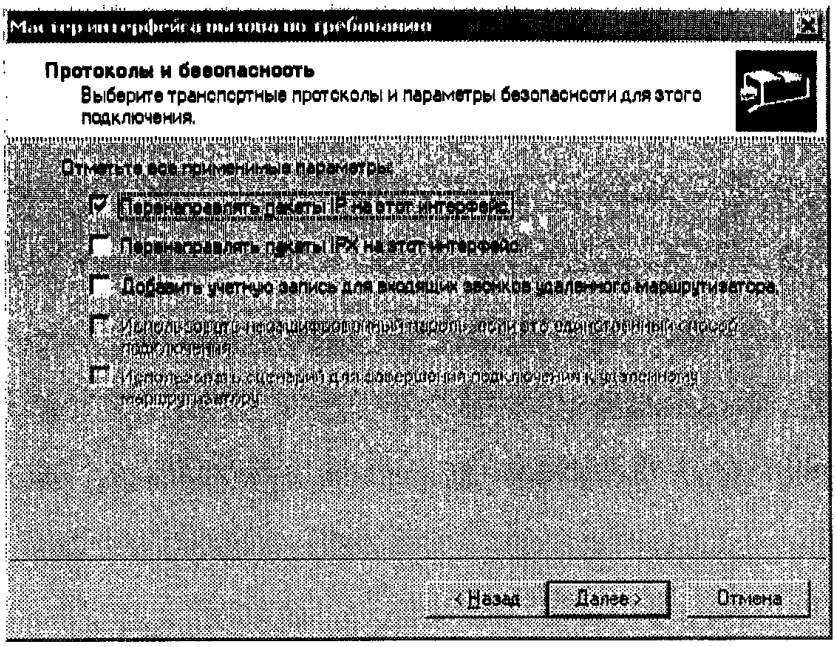


Рис. 10.34. Выбор транспортных протоколов для туннеля

Следующее окно служит для указания параметров учетной записи, которая будет проверяться удаленным сервером при установке туннеля. Напомним, что туннель является двусторонним и каждый из серверов-партнеров выполняет свою процедуру его установки, и только после того как оба успешно выполнили ее, туннель считается установленным. Учетная запись используется при установке одного направления туннеля с текущего сервера (на котором сейчас выполняются настройки). Это может быть доменная учетная запись или локальная. В последнем случае она должна быть явно создана на удаленном сервере VPN и ей должны быть назначены соответствующие права. Если на обоих объектах, между которыми устанавливается туннель, установлены контроллеры домена (когда сеть построена на базе Microsoft Network), то на обоих серверах можно задать одну учетную запись, которая предварительно должна быть зарегистрирована в домене. Из опыта автор рекомендует именно такое решение, так как оно позволяет гибко и централизованно управлять учетными записями, особенно если сеть включает более одного туннеля. Излишне говорить, что пароль для учетной записи должен быть сложно предсказуемым и что администратору следует включить флажок, указывающий на то, что пароль никогда не «устареет».

На рис. 10.35 используется доменная учетная запись (из домена TOMTIT). Применительно к домену, построенному на базе службы каталогов Microsoft Active Directory, серверы-партнеры могут одновременно выполнять роли контроллеров этого домена, и на каждом при настройке туннеля допускается указывать одну и ту же учетную запись. Следует только не забыть о том, что создавая учетную запись на одном из контроллеров, потребуется некоторое время, прежде чем эта информация будет реплицирована на другой контроллер, особенно если они располагаются в разных сегментах (sites). На этом этапе процедура настройки туннеля завершается. Вновь созданный интерфейс DOD VPN появится в консоли службы RRAS в дереве Routing Interfaces (интерфейсы маршрутизации) (рис. 10.36).

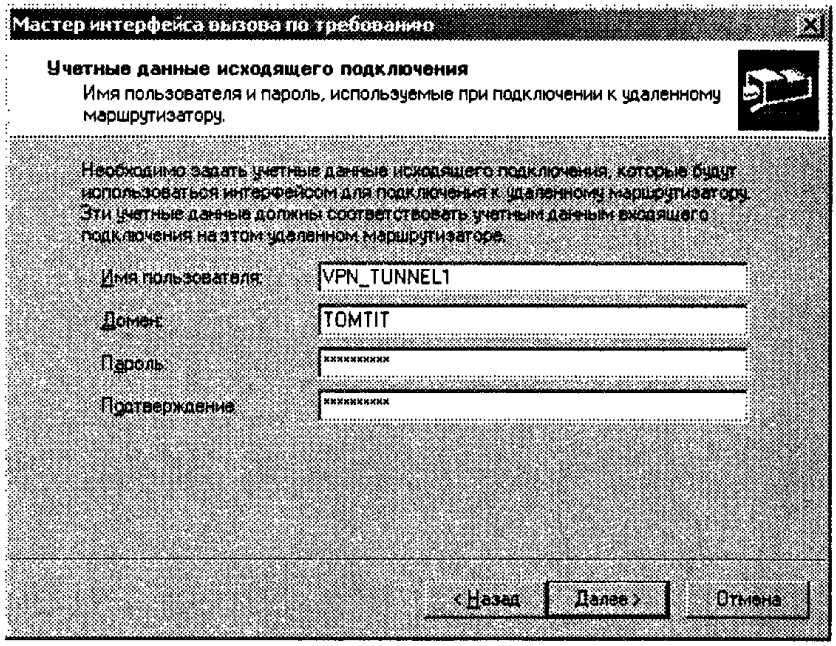


Рис. 10.35. Настройка учетной записи для туннеля

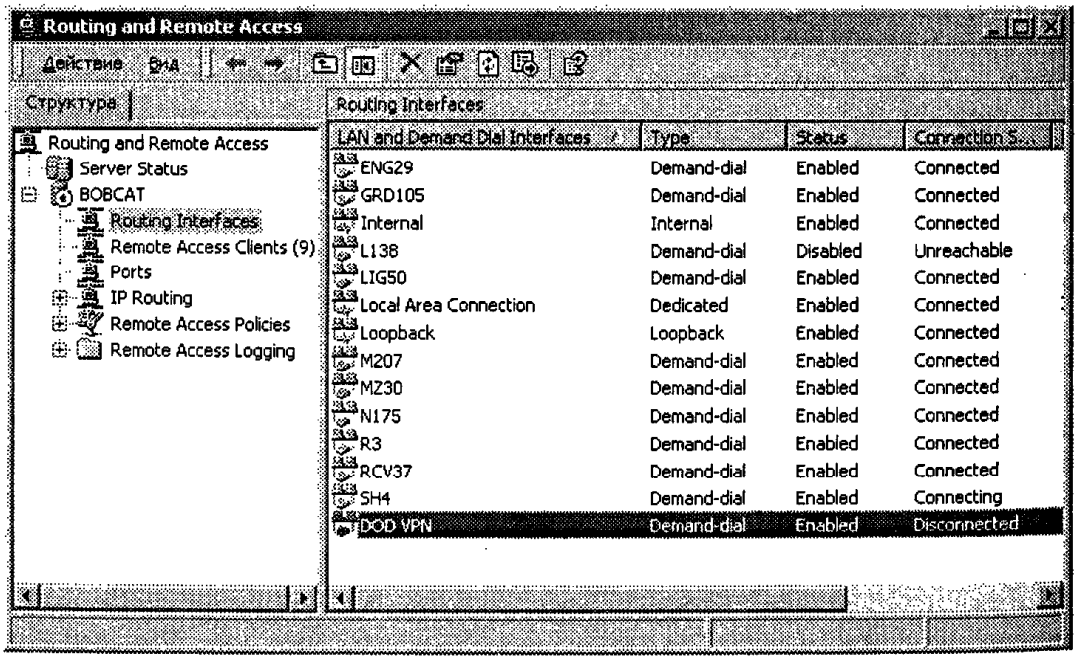


Рис. 10.36. Состояние нового интерфейса

У нового интерфейса есть статус (Status) который может быть Enabled (разрешен) или Disabled (запрещен — администратор волен менять статус в любое время). Как несложно догадаться, статус Disabled запрещает использование этого интерфейса на сервере. Также каждый интерфейс имеет статус соединения (Connection Status), который может быть Connected (интерфейс подключен, или, иными словами, сервер успешно установил свою часть туннеля), Connecting —

происходит попытка установить туннель, Unreachable (через этот интерфейс дейтаграммы посылаться не будут, как это показано на рисунке 10.36 для интерфейса с названием L138) и Disconnected — отключен. Вновь созданный интерфейс типа DOD VPN поначалу имеет такой статус, до тех пор пока администратор не выберет опцию Connect, щелкнув правой кнопкой мышки на интерфейсе. Там же есть пункт выпадающего меню Свойства (Property), выбор которого приводит к появлению диалогового окна с несколькими вкладками, на которых можно настроить параметры вновь созданного интерфейса (рис. 10.37).

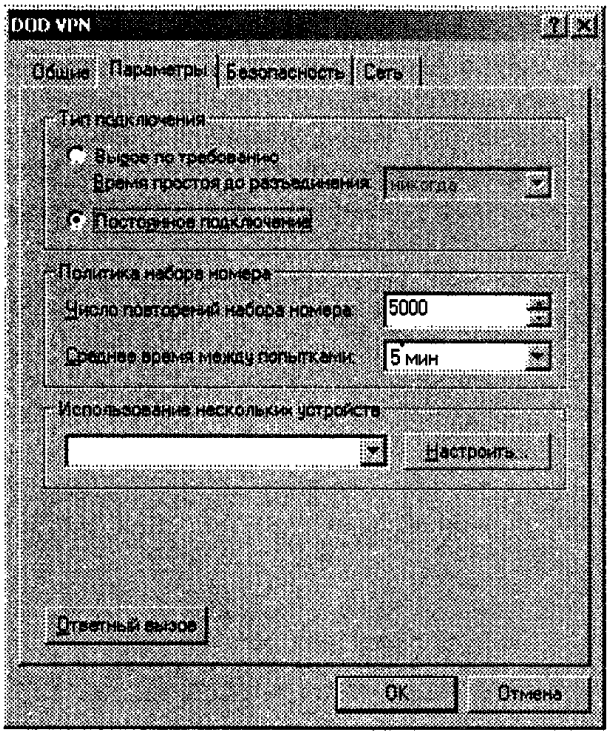


Рис. 10.37. Свойства подключения интерфейса

На вкладке Параметры диалогового окна настройки параметров интерфейса администратор может указать, должен ли интерфейс DOD VPN устанавливаться по требованию или поддерживаться постоянно. Первый вариант подразумевает, что сервер начнет процедуру инициации туннеля, когда он получает дейтаграмму, направленную в сеть, маршрут до которой подразумевает прохождение по туннелю (или, иными словами, отправку через настроенный интерфейс DOD VPN). Другой альтернативой является опция Постоянное подключение, при выборе которой туннель будет устанавливаться, а затем работать постоянно (то есть не разрываться по истечении времени неактивности). По опыту эксплуатации сети более чем с десятью туннелями рекомендуется выбрать именно последний вариант. Служебный трафик туннеля хоть и вносит дополнительные накладные расходы (как с точки зрения пропускной способности, так и в отношении финансовых затрат), они крайне несущественны — в сети, построенной на базе туннелей PPTP и с достаточно интенсивным трафиком между объектами (общим количеством порядка 10), служебный трафик (к которому относятся сегменты

протокола TCP с портом получателя 1723) составляет порядка 1% от суммарного объема трафика, проходящего по всем туннелям. Можно предположить, что такими накладными расходами стоит пренебречь и выбрать вариант, когда туннели поддерживаются серверами постоянно, а не устанавливаются по первому требованию и закрываются по истечении периода неактивности. Дополнительно (рис. 10.37) администратор может настроить еще два параметра, которые способны повлиять на работу туннеля — это **Число повторений набора номера** и **Среднее время между попытками**. Применительно к туннелю эти параметры начинают работать в случае, когда туннель оборвался и сервер пытается его восстановить (к сожалению, более подробной информации по этому вопросу не удалось найти даже на сайте компании Microsoft). Параметры, приведенные на рисунке, были выработаны экспериментальным путем при управлении сетью с туннелями. Фактически они означают, что при обрыве туннеля (если, например, пропала связь через сеть Интернет) сервер будет пытаться восстановить его 5000 раз с интервалом между попытками в 5 минут. После этого в случае неудачи интерфейс перейдет в состояние **Unreachable**.

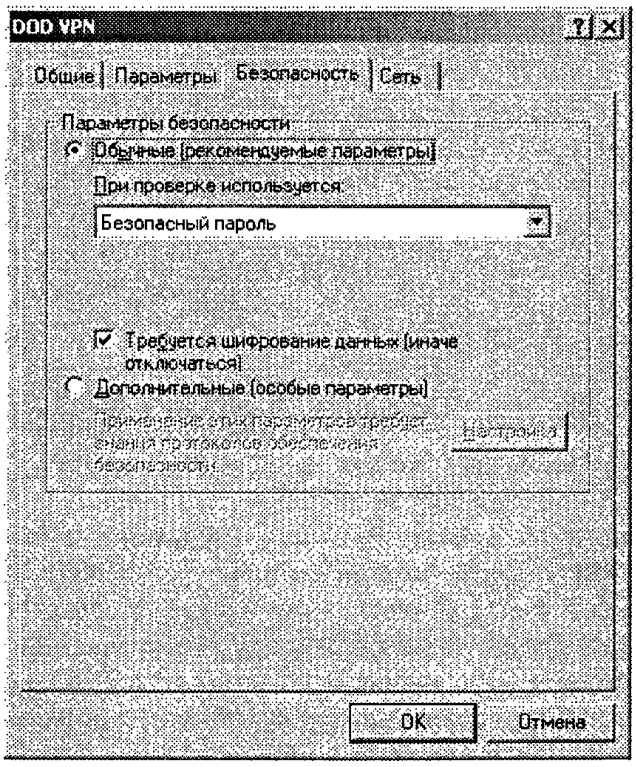


Рис. 10.38. Параметры безопасности туннеля

Другая вкладка (рис. 10.38) позволяет администратору настроить параметры безопасности применительно к туннелю. По умолчанию предлагается задействовать обычные параметры, которые являются рекомендованными для большинства ситуаций и обеспечивают высокую степень безопасности туннеля — используется безопасный метод проверки пароля (алгоритм MS-CHAP v2) и обязательным условием является шифрование передаваемых данных. Следует отметить, что

администратор имеет возможность достаточно гибко управлять параметрами туннеля, в том числе и настроить туннель таким образом, что будет выполняться, например, только сжатие данных без последующего шифрования; он волен отключить сжатие и оставить только шифрование; задействовать эти две возможности вместе или отключить их вообще (кстати, некоторые примеры перехваченных пакетов были получены в таких условиях). Если при настройке туннеля принимаются установки по умолчанию (рекомендованные), то это означает, что сервер будет пытаться при установке туннеля задействовать алгоритм MS-CHAP (начиная с усовершенствованной второй версии), данные будут сжиматься, а затем шифроваться. В том случае, если администратор хочет провести более гибкую настройку параметров туннеля, он может включить режим **Дополнительные (особые параметры)** и нажать кнопку **Настройка**. В результате появится диалоговое окно, показанное на рис. 10.39, где администратор волен выбрать один из четырех возможных вариантов шифрования данных, а также указать разрешенные к использованию алгоритмы аутентификации.

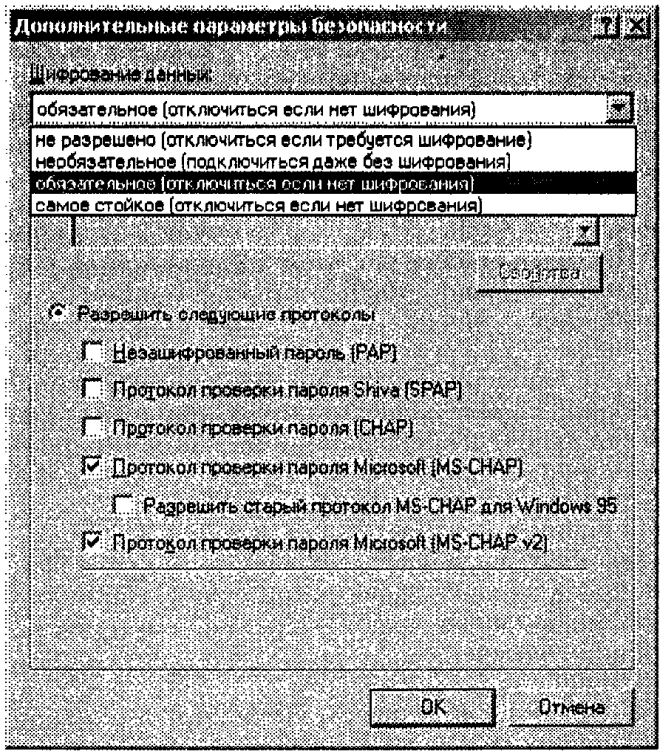


Рис. 10.39. Дополнительные параметры туннеля

Обратите внимание, что эти опции влияют на то, как именно будет работать созданный ранее интерфейс DOD VPN при установке одной части туннеля с удаленным сервером. Так, можно уточнить, что требуется обязательное шифрование, и если удаленный сервер настроен на то, чтобы передавать данные без этой процедуры, то интерфейс DOD VPN не сумеет перейти в состояние **Connected** (подключен). Как видим, есть несколько вариантов поведения при подключении в зависимости от требований к шифрованию данных. Обычно оставляют

настройки по умолчанию, подразумевая обязательное шифрование, и если удаленный сервер его не поддерживает, туннель установлен не будет (точнее, одна его часть — от настраиваемого сервера к удаленному). Далее, администратор может выбрать алгоритмы аутентификации — сервер будет проводить аутентификацию пользователя с удаленным сервером именно по выбранному алгоритму. Учитывая сказанное ранее в этой главе, настоятельно рекомендуется воспользоваться второй версией протокола MS-CHAP как наиболее безопасной. Если оба сервера работают на базе операционной системы Microsoft Windows 2000, то предпочтительно выбрать только указанный алгоритм (MS-CHAP v2), а остальные запретить к использованию.

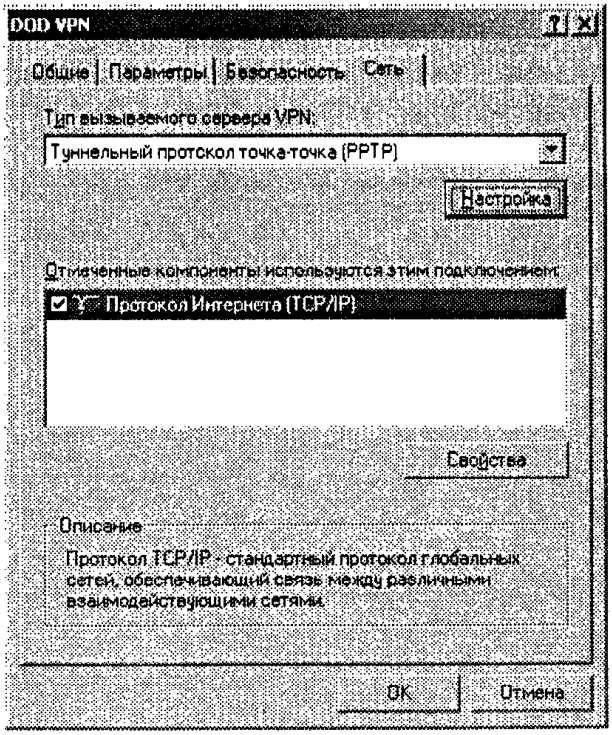


Рис. 10.40. Сетевые параметры туннеля

Следующая вкладка диалогового окна настройки параметров интерфейса DOD VPN включает в себя две группы параметров — тип вызываемого сервера VPN и параметры протокола TCP/IP для этого интерфейса при подключении (рис. 10.40). Первая группа позволяет администратору настроить параметры самого туннеля PPTP (а точнее протокола PPP, который является базовым для туннеля), как то задействование протокола LCP, контроль процедуры сжатия и согласование многоканальных подключений. По умолчанию все эти параметры активны, как показано на рис. 10.41. Из них наибольший интерес представляет флажок, включающий или выключающий предварительное сжатие информации перед выполнением процедуры шифрования (технология MPPC). В свою очередь, протокол управления каналом связи (LCP) обеспечивает метод организации, выбор конфигурации, поддержание и завершение работы протокола PPP.

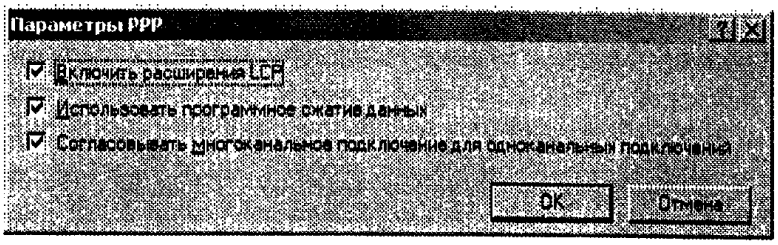


Рис. 10.41. Параметры протокола PPP

Администратор может также настроить параметры протокола TCP/IP для созданного интерфейса DOD VPN. Напомню, что каждый из серверов-партнеров поочередно будет выполнять роль клиента и сервера при установлении туннеля. В роли клиента он отвечает за подключение к удаленному серверу, как это, например, происходит при звонках пользователей в корпоративную сеть по обычному модему. По ходу клиенты получают адресную информацию протокола TCP/IP (включая такие параметры, как шлюз по умолчанию, адреса серверов WINS и DNS и т. п.). Аналогичная ситуация имеет место и при подключении клиента к серверу VPN — он автоматически получает эту информацию от удаленного сервера (как правило, последний задействует протокол DHCP — он опрашивает серверы DHCP в подключенной сети, получает адресную информацию для клиентов, а затем при установлении соединения предоставляет ее). То есть при подключении интерфейс DOD VPN приобретает свой собственный адрес IP и уточняет другие параметры, которые могут быть настроены администратором на серверах DHCP на удаленном объекте, с которым устанавливается туннель. На удаленной сети может работать обычный сервер DHCP, который помимо обслуживания клиентов в этой сети также предоставляет информацию и локальному интерфейсу DOD VPN. Кстати, сервер, отвечающий за установления туннеля, способен выполнять и роль сервера DHCP. По умолчанию параметры протокола TCP/IP интерфейса DOD VPN настроены на автоматическое получение адресной информации и в большинстве случаев эти настройки можно оставить без изменений.

Перейдем к другому, не менее важному аспекту настройки сервера для поддержания туннеля — формированию политики удаленного доступа (Remote Access Polices). Если до этого мы рассматривали параметры интерфейса DOD VPN как бы со стороны клиента (то есть настраиваемый сервер, начиная инициализацию своей части туннеля, выступает клиентом), то теперь перейдем к настройке параметров, определяющих его поведение со стороны сервера, принимающего запросы от удаленных клиентов. Для определения этих параметров как раз и служит политика доступа. Применительно к туннелю PPTP она настраивает параметры, которые будут обязательны для удаленного клиента (или клиентов, если на сервере «сходятся» несколько туннелей) и без учета которых сервер не сможет принять запрос на установку второго направления туннеля (от удаленного сервера к локальному). Даже если локальный сервер VPN установил свою часть туннеля (ведь политика настраивается и на удаленном сервере-партнере, но на обоих серверах параметры политики могут отличаться друг от друга), сам туннель не будет работоспособным, так как требуется обязательная функциональность обоих направлений (применительно к ситуации связи двух локальных сетей

«однобокий» туннель позволит передавать трафик только в одном направлении — здесь начинаются проблемы с маршрутизацией ответного трафика).

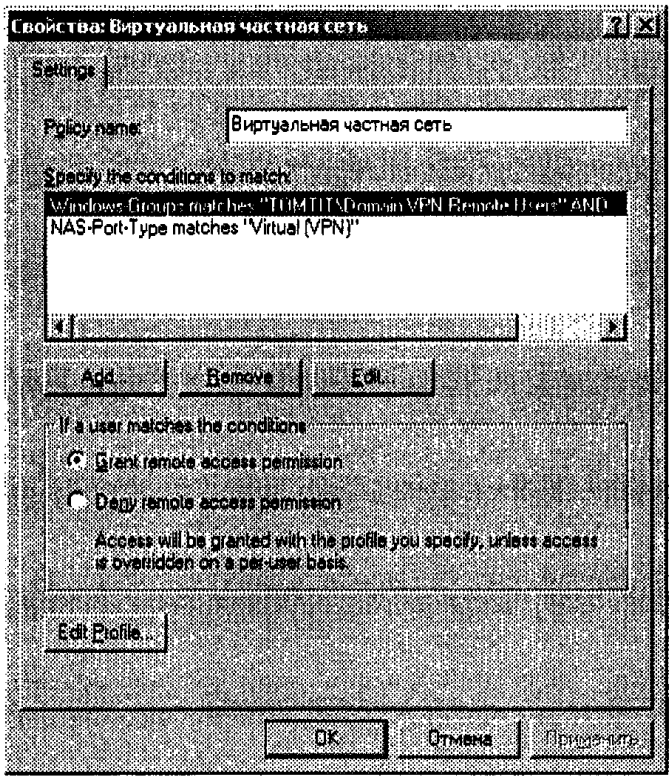


Рис. 10.42. Настройка политики доступа

На рис. 10.42 показано диалоговое окно настройки параметров политики доступа (ее имя «Виртуальная частная сеть»), определяющей те критерии, которые удаленный сервер должен поддерживать при установлении туннеля с локальным сервером. В политике доступа выделяются условия (conditions) и действие, которое должно выполняться при соблюдении этих условий. В нашем примере оба сервера-участника туннеля входят в единый домен (один из ранее отмеченных вариантов организации сети) и они оба используют для туннеля одну и ту же учетную запись домена при проверке имени и пароля пользователя. В том случае, если в сети существуют несколько туннелей и для каждого из них было принято решение задействовать свою доменную учетную запись, то можно посоветовать создать группу в домене, включающую в себя эти специальные учетные записи. Если вернуться к рисунку 10.42, то, как видим, одним из условий является то, что учетная запись, используемая при установлении части туннеля на этот сервер, должна входить в группу Domain VPN remote Users домена TOMTIT. Следующее условие специфичное для туннеля — тип порта NAS (NAS-Port-Type; тип порта сервера удаленного доступа) должен соответствовать Virtual (VPN). В принципе, двух таких условий достаточно для проверки того, что удаленный сервер, устанавливающий туннель с локальным, работает с корректной учетной записью, и того, что устанавливается именно туннель (условие — NAS-Port-Type

matches Virtual (VPN)). Далее следует указать, что именно нужно сделать, если условия соблюдены — разрешить или запретить доступ. Понятно, что следует выбрать опцию, разрешающую доступ, иначе туннель установлен не будет. Помимо условий можно настроить профиль политики доступа (**Edit Profile...**). При нажатии на соответствующую кнопку откроется диалоговое окно, в котором настройке поддается довольно большое количество параметров, но применительно к туннелю наибольший интерес вызывают две вкладки — аутентификация (**Authentication**) и шифрование (**Encryption**).

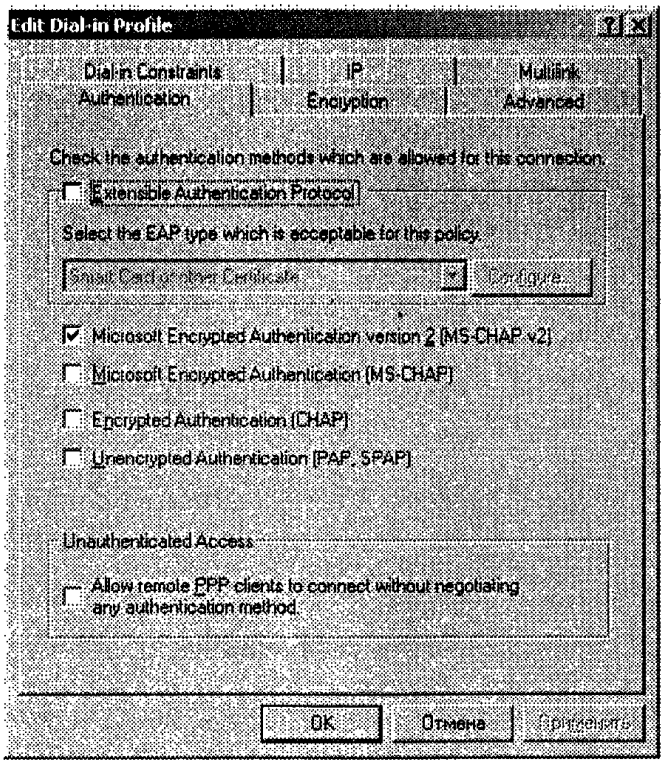


Рис. 10.43. Настройка параметров аутентификации в политики доступа

Рисунок 10.43 показывает содержимое диалогового окна на вкладке **Authentication**. Здесь администратор может выбрать протоколы аутентификации, которые будут поддерживаться для этой политики доступа (а политика доступа обслуживает как один, так и несколько туннелей). Видно, что предлагаются несколько протоколов на выбор, включая опцию «вообще не проводить аутентификацию удаленных клиентов». В данном случае установлен флажок **MS-CHAP v2**, и если удаленный сервер попытается задействовать другой алгоритм аутентификации, туннель не состоится. Следующая вкладка (рис. 10.44), которая может предоставлять интерес — **Encryption**, где администратору позволено задать уровни шифрования данных, поддерживаемые сервером. Сам уровень (за исключением параметра, вообще отключающего шифрование — **No Encryption**) зависит от длины ключа, который используется при шифровании. Зависимость простая — чем длиннее ключ, тем более устойчивее будут зашифрованные данные к попыткам взлома. В рассматриваемом примере сервер примет любой уровень шифрования

данных, даже в том случае, если эта процедура вообще не будет выполняться. Скорее, последний случай более подходит к тестовым целям, в то время как в рабочей сети рекомендуется выставить только последний флажок (Strongest), подразумевающий задействование ключа длиной 128 бит.

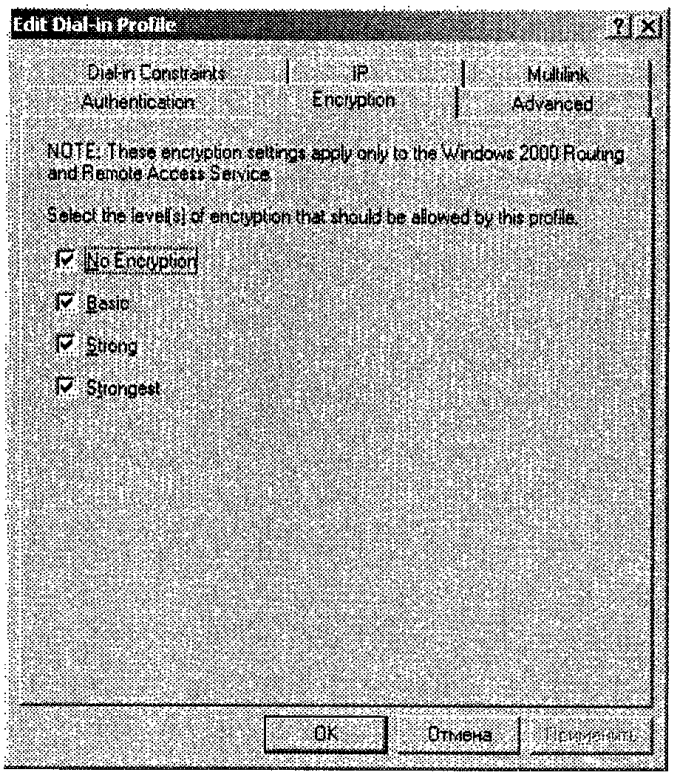


Рис. 10.44. Настройка уровня шифрования данных в профиле политики доступа

На этом можно считать что локальный сервер готов к обслуживанию туннеля через интерфейс DOD VPN. Точнее, он готов выполнять функции клиента (интерфейс будет пытаться установить одну составляющую туннеля с удаленным сервером VPN) и сервера (будет готов принять запрос на установление второй части туннеля). Как уже было отмечено, если первая часть процедуры установки туннеля была успешно выполнена, то интерфейс перейдет в состояние Connected (подключен). А когда удаленный сервер VPN успешно установит соединение с локальным (настраиваемым) сервером, на последнем это можно наблюдать в списке удаленных клиентов (Remote Access Clients). Как показано на рис. 10.45, в итоге к локальному серверу было подключено 8 клиентов. Причем это могут быть клиенты туннеля PPTP (удаленные серверы или конечные пользователи, подключающиеся к корпоративной сети через Интернет) или, например, клиенты, входящие в сеть через аналоговые модемы (для службы RRAS оба типа являются, в принципе, равноправными, нужно только учесть это при настройке политик доступа, их может потребоваться создать больше чем одну — одну для клиентов туннеля PPTP, другую для тех, кто работает через модемы).

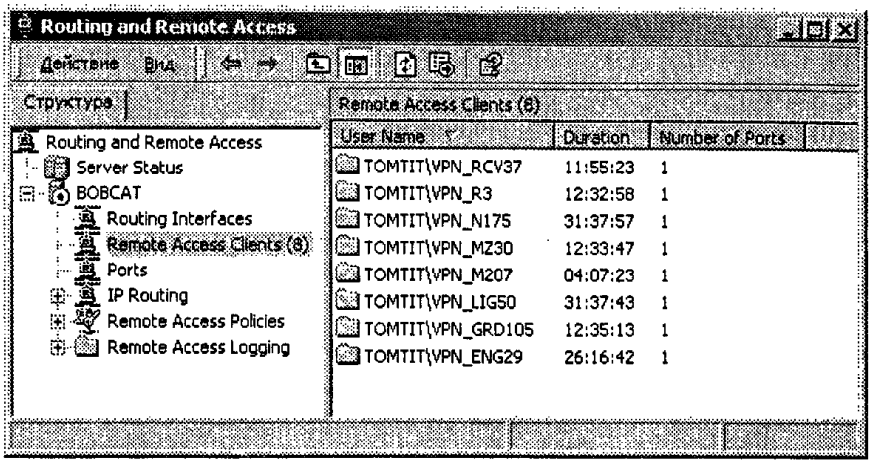


Рис. 10.45. Список подключенных клиентов

Рисунок 10.45 — «снимок» с работающего сервера, который обслуживает порядка десяти туннелей (используется схема типа «звезда») и в настоящий момент к нему подключено 8 туннелей, каждый из которых представляет некий сервер, работающий на удаленном объекте. Для любого из клиентов можно вызвать диалоговое окно статуса (Status), если щелкнуть на имени пользователя два раза мышкой (рис. 10.46).

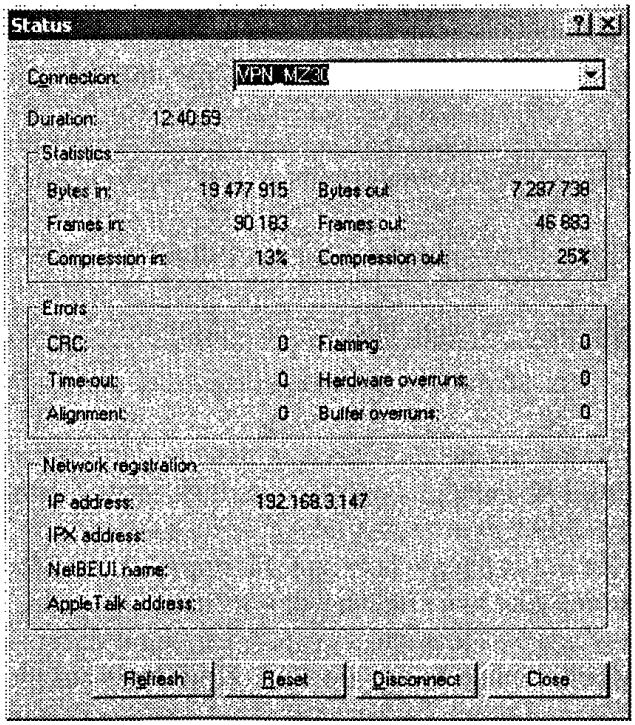


Рис. 10.46. Статус подключенного клиента

Появляющееся диалоговое окно предоставит такую информацию, как количество переданных и полученных по туннелю байтов (включая процент сжатия

в каждом направлении, если, конечно, согласовано использование алгоритма MPPC), время, в течение которого клиент был подключен к серверу (в примере — 12 часов), и адрес IP, который был назначен этому клиенту. Если нужно, администратор может принудительно отключить клиента, нажав кнопку Disconnect, хотя это не закроет весь туннель, а только отключит удаленный сервер VPN от локального. В дальнейшем, если условия связи позволят, удаленный сервер попытается вновь выполнить свою часть подключения. В консоли службы RRAS также есть такое свойство как Ports (порты), которое имеет непосредственное отношение к туннелям PPTP. Применительно к туннелю, порт — это некая точка подключения на локальном сервере. Администратор может настроить определенное количество портов, что аналогично настройке максимального количества туннелей, которые могут «сходиться» на этом сервере.

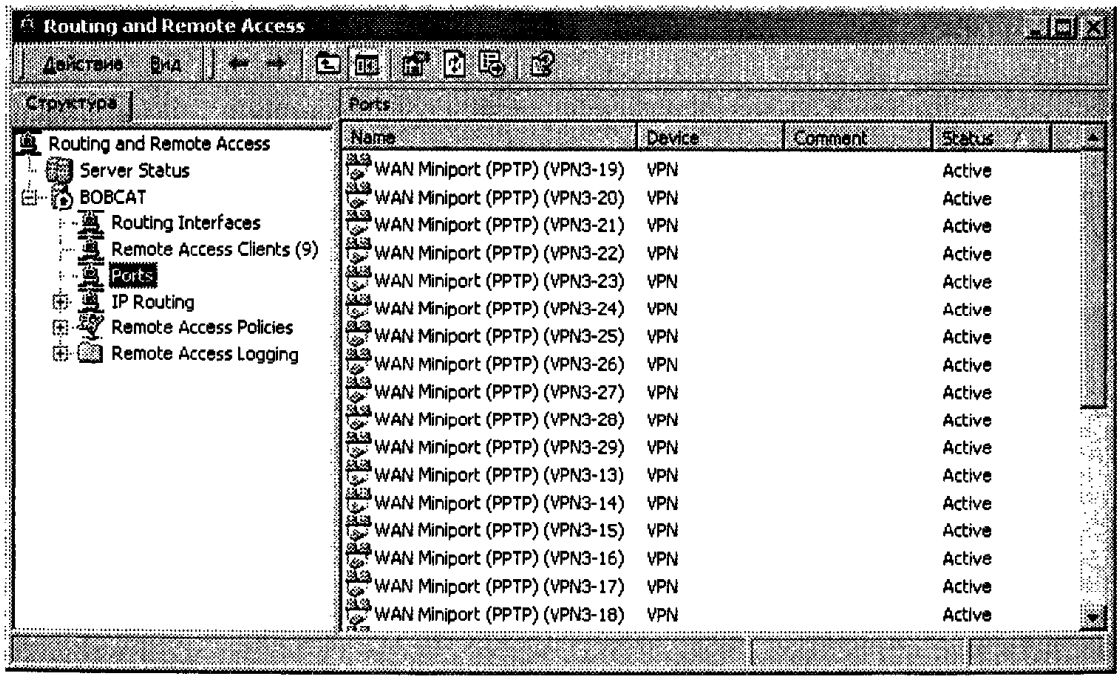


Рис. 10.47. Список активных портов на сервере

На рис. 10.47 показан перечень активных портов. Интересно отметить, что для каждого туннеля (а на момент получения рисунка сервер поддерживал одновременно 9 туннелей) задействуется два порта — по одному для каждого направления туннеля (сейчас на сервере активно 18 портов). Особой важности порты при создании туннелей для администратора не имеют, нужно только определить максимальное количество туннелей, которые сервер в состоянии поддерживать, и настроить соответствующее число портов. При этом различаются порты протоколов PPTP, L2TP и Direct Parallel (прямая связь через параллельный порт компьютера). Из рис. 10.48 видно, что для протокола PPTP настроено 30 портов и, учитывая, что каждый туннель задействует два порта, сервер в такой конфигурации сможет обслуживать до 15 туннелей одновременно. Администратор волен в любой момент изменить количество портов в большую или меньшую сторону.

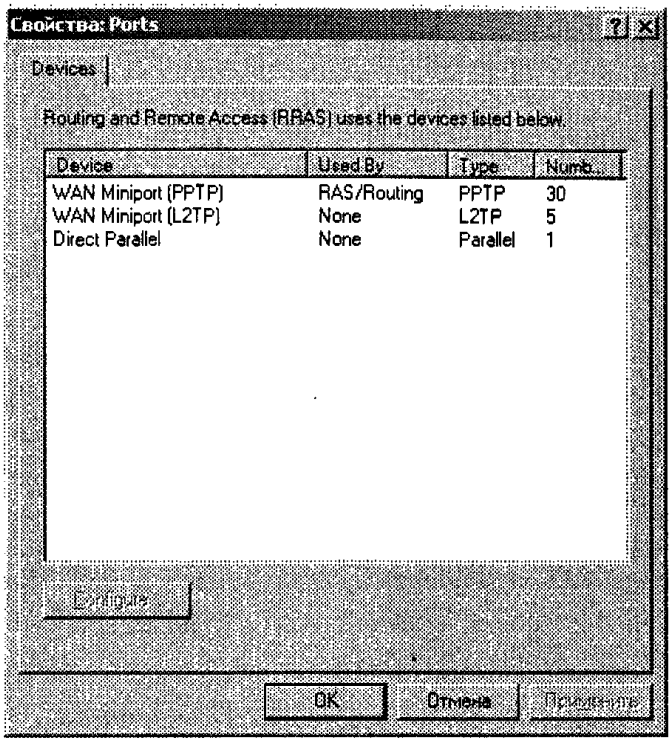


Рис. 10.48. Настройка количества портов на сервере

Мы рассмотрели все основные параметры, относящиеся к туннелям на базе протокола PPTP. Как видно, процедура настройки крайне простая и занимает совсем немного времени администратора. За рамками главы остались такие вопросы, как маршрутизация и ведение журнала, но, по мнению автора, они будут понятны и без дополнительного «разжевывания». Ведь после создания интерфейса DOD VPN он может рассматриваться как простой физический интерфейс, и, следовательно, к нему применимы все операции, связанные с маршрутизаций трафика, как к обычному интерфейсу.

А Протоколы

В этом приложении перечислены наиболее часто используемые протоколы, а также порты, которые они задействуют. Эта информация будет полезна, если администратору сети требуется настроить на маршрутизаторе защитный экран или списки доступа с целью блокировать или разрешить работу пользователей с тем или иным протоколом. Следует отметить, что некоторые протоколы фигурируют далее в нескольких таблицах.

Таблица 1. Перечень базовых протоколов, работающих в обход UDP/TCP

Протокол	Базовый протокол	Номера используемых портов	Полное название
EGP	IP	8	Exterior Gateway Protocol
GRE	IP	47	Generic Routing Encapsulation
ICMP	IP	1	Internet Control Message Protocol
EIGRP	IP	88	Enhanced Interior Gateway Routing Protocol

Таблица 2. Перечень базовых протоколов, использующих в своей работе UDP/TCP

Протоколы, опирающиеся на TCP/UDP	Базовый протокол	Краткое описание
FTP	TCP	File Transfer Protocol
Exchange	TCP	MS-RPC for Exchange
HTTP	TCP	Hypertext Transfer Protocol
Citrix	TCP	Citrix Published Application — native support 12.1(2)E
Netshow	TCP/UDP	Microsoft Netshow
RealAudio	TCP/UDP	RealAudio Streaming Protocol
r-commands	TCP	rsh, rlogin, rexec
SQL*NET	TCP/UDP	SQL*NET for Oracle
SunRPC	TCP/UDP	Sun Remote Procedure Call
TFTP	UDP	Trivial File Transfer Protocol
VDOLive	TCP/UDP	VDOLive Streaming Video

Таблица 3. Базовые протоколы с номерами используемых портов

Протокол	Базовый протокол	Номера портов	Полное название
BGP	TCP/UDP	179	Border Gateway Protocol
CU-SeeMe	TCP/UDP	7648, 7649	Desktop Videoconferencing
CU-SeeMe	UDP	24032	Desktop Videoconferencing
DHCP/Bootp	UDP	67, 68	Dynamic Host Configuration Protocol/ Bootstrap Protocol
DNS	TCP/UDP	53	Domain Name System
Finger	TCP	79	Finger User Information Protocol
Gopher	TCP/UDP	70	Internet Gopher Protocol
HTTP	TCP	80	Hypertext Transfer Protocol
HTTPS	TCP	443	Secure HTTP
IMAP	TCP/UDP	143, 220	Internet Message Access Protocol
IRC	TCP/UDP	194	Internet Relay Chat
Kerberos	TCP/UDP	88, 749	The Kerberos Network Authentication Service
L2TP	UDP	1701	L2F/L2TP Tunnel
LDAP	TCP/UDP	389	Lightweight Directory Access Protocol
MS-SQLServer	TCP	1433	Microsoft SQL Server
NetBIOS	TCP	137, 139	NetBIOS over IP (Microsoft Windows)
NetBIOS	UDP	137, 138	NetBIOS over IP (Microsoft Windows)
NFS	TCP/UDP	2049	Network File System
NNTP	TCP/UDP	119	Network News Transfer Protocol
Notes	TCP/UDP	1352	Lotus Notes
NTP	TCP/UDP	123	Network Time Protocol
PCAnywhere	TCP	5631, 65301	Symantec PCAnywhere
PCAnywhere	UDP	22, 5632	Symantec PCAnywhere
POP3	TCP/UDP	110	Post Office Protocol
PPTP	TCP	1723	Point-to-Point Tunneling Protocol
RIP	UDP	520	Routing Information Protocol
RSVP	UDP	1698,1699	Resource Reservation Protocol
SFTP	TCP	990	Secure FTP
SHTTP	TCP	443	Secure HTTP
SIMAP	TCP/UDP	585, 993	Secure IMAP
SIRC	TCP/UDP	994	Secure IRC
SLDAP	TCP/UDP	636	Secure LDAP
SNNTP	TCP/UDP	563	Secure NNTP

продолжение

Таблица 3 (продолжение)

Протокол	Базовый протокол	Номера портов	Полное название
SMTP	TCP	25	Simple Mail Transfer Protocol
SNMP	TCP/UDP	161, 162	Simple Network Management Protocol
SOCKS	TCP	1080	Firewall security protocol
SPOP3	TCP/UDP	995	Secure POP3
SSH	TCP	22	Secured Shell
STELNET	TCP	992	Secure TELNET
Syslog	TCP	514	System Logging Utility
Telnet	TCP	23	Telnet Protocol
X Windows	TCP	6000-6003	X11, X Windows

Следующая информация адресована администраторам сетей, построенных на базе сетей Microsoft Windows. Она способна помочь в ситуации, когда необходимо блокировать трафик, относящийся к сетям Microsoft Windows. Можно предположить, что наиболее востребованной она будет, когда администратору нужно разграничить части внутренней сети, оставляя, однако, связь между серверами и службами Microsoft. Важно отметить, что представленные в табл. 4 данные требуют периодического обновления, так как при появлении новых версий продуктов номера используемых портов могут меняться. Источник информации: www.microsoft.com.

Таблица 4. Протоколы, задействуемые службами в сетях Microsoft Network

Сервис	Протокол	Запрашиваемые порты для взаимодействия клиент-сервис	Входящий порт сервиса	Исходящий порт сервиса
Функция browsing (NetBIOS поверх TCP/IP)	UDP (запрос)	137	137	137
	UDP (ответ)	138	138	138
Служба DHCP lease	TCP (запрос)			67
	TCP (ответ)			68
Служба DNS (запросы от клиента к серверу)	TCP/UDP (зависит от реализации)	1024–5000	53	53
Служба DNS (запросы между серверами)	TCP/UDP (зависит от реализации)	53	53	53
Служба DNS (передача зоны primary a secondary)	TCP	53	53	1024–5000
Служба DNS (передача primary a secondary soa record)	UDP	53	53	53

Сервис	Протокол	Запрашиваемые порты для взаимодействия клиент-сервис	Входящий порт сервиса	Исходящий порт сервиса
Общие ресурсы в сети	UDP(разрешение имени)			137
	TCP (сеанс)			139
Протокол FTP — данные	TCP			20
Протокол FTP	TCP			21
HTTP	TCP			80
HTTP (SSL)	TCP			443
IMAP4	TCP			143
IRC	TCP			531
LDAP	TCP			389
LDAP (SSL)	TCP			636
Membership MSN	TCP			569
Microsoft Chat	TCP (сервер-сервер)			6665
	TCP (клиент-сервер)			6667
NetMeeting	TCP (обнаружение пользователя)			522
	TCP (T.120)			1503
	TCP (звуковой вызов)			1731
	UDP (протокол RTP)			Динамический
NetShow	UDP	Tcp/1755	Tcp/1755	Udp/1755
	TCP		1755	1755
	HTTP	80	80	80
	Групповая доставка 224.0.0.1–239.255.255.255	1–65000	1–65000	1–65000
	DCOM	135	135	1024–5000
NNTP	TCP			119
POP3	TCP			110
PPTP	PPTP (protocol 47)			1723
Общий принтер в сети	UPD (разрешение имени)			137
	TCP (сеанс)			139
RADIUS	UDP (аутентификация)			1645 или 1812

продолжение

Таблица 4 (продолжение)

Сервис	Протокол	Запрашиваемые порты для взаимодействия клиент-сервис	Входящий порт сервиса	Исходящий порт сервиса
	UDP (учет)			1646 или 1813
RPC (например, User Manager Server Manager и т. д.)	TCP			135
	TCP (сеанс)			Динамический
SMTP	TCP			25
SNMP	UDP			161
SNMP Trap	UDP			162
SQL Server (TCP)	UDP/TCP (разрешение имени)			53
	TCP (сеанс)			1433
SQL Server (RPC client, supports encryption over NetBEUI, IPX/SPX, TCP/IP)	UDP (name lookup)			137
Telnet	TCP			23
Регистрация WINS	UDP (NetBIOS поверх TCP/IP)			137
Репликация серверов WINS	TCP			42
Аутентификация Windows Challenge/ Response	TCP (NetBIOS поверх TCP/IP)			139
X400	TCP			102

Алфавитный указатель

A

AAA, 365
ACK, 155, 190
ACL, 344
Active Directory, 387, 411
ADSL, 308
any, ключевое слово, 351
AR, 324
ARP
  запрос, 171
ARP spoofing, 123
ARP-таблица, 120
  очистка, 121
Asymmetric Digital Subscriber Line, 308
ATM, 322
Automatic Broadcast Control, 377
AUX, 50

B

backlog, 194, 195
banners, 66
Bc, 326, 335
Be, 326
BECN, 325

C

CAR, 334
CBQ, 267
CDP, 135
CHAP, 368, 369
chap, ключевое слово, 369
Checkpoint Firewall-1, 239, 358
CIDR, 109
CIR, 324, 326, 335
Cisco ConfigMaker, 22, 67, 117, 119
  v2.4, 87
Cisco IOS
  буфер истории, 34
  встроенный отладчик, 137
  интерфейс командной строки, 28
  команды проверки аппаратных ресурсов, 52
  образ, 39
  подавление источника, 164
Cisco IOS (*продолжение*)
  приглашение, 29
    в режиме подконфигурации, 32
  режимы конфигурации, 29
  списки доступа, 344
Cisco IOS Firewall Feature Set, 358
Cisco Systems, 17
CiscoWorks 2000, 58
CLI, 27
Communication Administratively Prohibited,
config, ключевое слово, 32
CTY/CON, 50
CWND, 230

D

DE, 324
decoy-сканирование, 171
default, ключевое слово, 370
description, 131
Destination Host Unknown, 165
Destination Network Unknown, 165
DF, 46, 235
DLCI, 322
DNS, 126
  сервер, настройка адресов, 127
DoS, 194, 241
DWFQ, 271
Dynamic Host Configuration Protocol, 178
DynamicBacklogGrowthDelta, 199

E

eEye Iris Network Traffic Analizer, 141
enable password, 30, 69
enable secret password, 30, 69, 70
EnableDynamicBacklog, 198
Ethereal, 143
  и WinDump, 143
Exec, заголовок, 66
exec, ключевое слово, 372

F

Fast Retransmit, 226
FCS, 322

FECN, 325
FIFO, 246
FQ, 267
Fragmentation needed and DF flag Set, 165
Frame Relay, 320
FRTS, 334, 336, 337
FTP, 225

G

G.shdsl, 310
Gigabit Ethernet, 211
GTS, 334, 336
GuardianPro, 243
  v5, 243
  v5.02, 364
  v5.0x, 201

H

Host Precedence Violation, 165
Host Unreachable, 164
Host Unreachable for Type of Service, 165
host, ключевое слово, 351

I

IAS, 368
ICMP, 139, 235
  address mask reply, 179
  address mask request, 178
  destination unreachable, 164, 181
  echo reply, 156
  echo request, 151
  host unreachable, 168
  information request, 178
  parameter problem, 180
  port unreachable, 170
  protocol unreachable, 168
  redirect, 171
  source quench, 160
    маршрутизаторы Cisco Systems, 164
    нестандартные сообщения, 163
  sweep, 154
  time exceeded, 171, 181
  timestamp request, 176, 177
  авторизованный агент, 179
  временная синхронизация, 176
  данные
    шаблон заполнения, 159
  и UDP, 170
  и UNIX, 153
  и защитный экран, 162
  и модель OSI, 140
  идентификатор и номер последовательности, 152
  классификация сообщений, 150
  опрос работающих хостов, 154
  перенаправление, 172
  сообщение
    поле данных, 157
  сообщения
    обязательные поля, 150
    запросов, 151
  таймеры, 169
  тип и код сообщения, 147
  типы сообщений, 148
  условия формирования сообщений, 149
  фильтрация сообщений, 188
  фрагментация, 158
  широковещательный трафик, 155
IDS, 198, 361, 363
IETF, 219
IIS, 200
INTERNETWORK CONTROL, 163
InterNIC, 82
IP
  дейтаграмма
    смещение фрагмента, 186
  идентификатор дейтаграммы, 184
  флаг More Fragments (MF), 183
  фрагментация, 183
IP identification number, 183
IP Precedence, 249, 251
IP Subnet Calculator, 87
ip unnumbered, 118
IP-адрес
  зарезервированные, 79
  классы, 77, 80
    диапазон значений, 79
  формат, 77
IP-заголовок
  бит DF, 46, 235
  поле TOS, 249, 250, 251
IP-сеть
  качество обслуживания, 249
    поле IP Precedence, 249, 251
    поле TOS, 250
    рекомендации RFC 1812, 251
  контроль перегрузок, 228
    протокол RSVP, 228
  планирование, 86
  предотвращение перегрузок
    алгоритмы, 229
    медленный старт, 229
IP-телефония, 378, 379
ISN, 190
ISS, 363

K

keepalive, сообщения, 43
Kerio WinRoute Pro, 385

L

Layer-2 Forwarding, 382
Layer-2 Tunneling Protocol, 382, 383

Layer-3 Tunneling, 384
Link Control Protocol, 418
LMI, 327
local, ключевое слово, 369
log, ключевое слово, 357
Login, заголовок, 66
loopback, 129

M

MAC-адрес
  префикс производителя, 146
MaximumDynamicBacklog, 199
Microsoft Active Directory, 154
Microsoft Network Monitor, 141
Microsoft Point-to-Point Compression, 401
Microsoft Point-to-Point Encryption, 383, 401
Microsoft SMS Server, 141
Microsoft Windows 2000
  и технология VPN, 386
Microsoft Windows 2000 Professional, 152
MINCIR, 339
MinimumDynamicBacklog, 198, 200
MOTD, заголовок, 66, 67
MPPE, 383
MS-CHAP, 368
MSS, 212, 235
MTU, 45, 74, 183, 187, 235
  Path MTU, 46
  Path MTU Discovery, 234
  правило выбора, 45

N

NAR, 245
NAT, 238
  таймер неактивности, 240
NBMA, 329
NETWORK CONTROL, 163
Network Unreachable, 164
Network Unreachable for Type of Service, 165
nmap, 154, 187
no, ключевое слово, 37
none, ключевое слово, 372
NVRAM, 60

O

one-way, алгоритм шифрования паролей, 70
overload, ключевое слово, 241

P

PAP, 368, 369
password, ключевое слово, 371
Path MTU, 46
Path MTU Discovery, 234
  параметр реестра
    EnablePMTUBHDetect, 236
    EnablePMTUDiscovery, 236
ping, 151, 157
ping sweep, 154
PMTU, 46
Point-to-Point Tunneling Protocol, 374
Port Unreachable, 164
PPP, 369
PPP IP Control Protocol, 424
PPTP
  управляющее соединение, 391
Precedence cutoff in effect, 165
Protocol Unreachable, 164
proxy ARP, 245
proxy server, 387
PSH, 217
PVC, 130, 321

R

RADIUS, 366
radius-server, ключевое слово, 373
RAM, 18
rcp, сервер, 61
RCV.NXT, 204
receive timestamp, 177
RED, 252, 255
RedHat Linux 6.2, 157
RFC 1009, 96
RFC 1027, 121
RFC 1035, 66
RFC 1058, 292, 293, 295
RFC 1122, 151, 157, 177, 229
RFC 1191, 46, 139
RFC 1256, 139
RFC 1323, 212, 214, 224
RFC 1349, 250
RFC 1388, 100
RFC 1517, 109
RFC 1518, 109
RFC 1519, 109
RFC 1520, 109
RFC 1701, 393
RFC 1702, 393
RFC 1812, 81, 160, 178, 181, 250, 251, 360
RFC 1918, 238
RFC 2118, 401
RFC 2205, 229
RFC 2759, 420
RFC 3078, 401
RFC 3079, 423
RFC 791, 72, 181
RFC 792, 149, 152
RFC 793, 189, 224
RFC 826, 119
RFC 896, 139
RFC 950, 82, 87, 139
RFC 972, 139
RFC 990, 76

RFC 997, 76
RIP, 276, 290, 291, 299
RIP-1, 96, 97, 295
ROM Monitoring (ROMMON), 32
round-robin fashion, алгоритм, 262
Round-Trip Time, 159
RRAS, 239
RST, 155, 362
RSVP, 228
RTO, 223, 225, 226
RTT, 225
running-config, ключевое слово, 65
RWIN, 235

S

SACK, 219
SDH, 211
Simple Service Discovery Protocol, 170
SND.NXT, 204
SND.UNA, 204
Snort Intrusion Detection System, 144
Source Host Isolated, 165
Source Route Failed, 165
SRTT, 223, 225
SSTHREST, 232
start-stop, ключевое слово, 372
startup-config, ключевое слово, 65
station address, 41
SuperScoute for ISA Server, 387
SurfControl, 387
SYN, 190
SynAttackProtect, 196
Syslog, 58
  сообщения, 58

T

TACACS+, 366
tacacs+, ключевое слово, 371
tacacs-server, ключевое слово, 373
TCB, 204
  переменные, 204
    RCV.NXT, 204
    SND.NXT, 204
    SND.UNA, 204
TCP
  сегмент, 147
TCP Intercept, 200
TCP Path MTU Discovery, 234
TCP Timestamps, 224
TCPdump, 142
TcpMaxConnectResponseRetransmissions, 197
TcpMaxHalfOpen, 196
TcpMaxHalfOpenRetried, 196, 197
TCP-сканирование, 155
Telnet, 376
Telnet, сеанс
  завершение, 135
  подавление, 133
  просмотр отладочной информации, 137
  управление маршрутизатором, 133
Timestamp Echo Reply, 224
Timestamp Value, 224
TOS, 250
transmit timestamp, 177
TTY, 50
tunnel, 129
two-way, алгоритм шифрования паролей, 70

U

UDP, 170

V

Virtual Private Netowork, 374
VLSM, 96
VPN, 173, 374
  и атаки DoS, 412
  инкапсуляция данных, 392
  интерфейс DOD, 429
  топология туннелей, 397
  управляющее соединение, 414
VPN, port mapping, 408
VTY, 50

W

WFQ, 267
WIC-2T, 309
WIC-2T, 310
Windows 2000 Service Pack 2, 402
Windows Sockets, 195
WinDump, 142
  параметры командной строки, 142
WRED, 255

X

xDSL, 157, 286

А

автоматический контроль широковещательных пакетов, 377
авторизация удаленного доступа, 371
административно заданное расстояние, 300
  по умолчанию, 301
административное расстояние, 300
адрес
  внутренний, 238
    глобальный, 241
    частный, 241
  интерфейса маршрутизатора
    вторичный, 117
    первичный, 116
  открытый, 238

адрес (*продолжение*)
разрешение, 120
обратное, 120
сервера DNS, 127
трансляция, 238
динамическая, 242
приоритеты, 244
статическая, 242
таймер неактивности, 240
адресное пространство организации
рекурсивное разбиение, 99
акронимы команд, 36
алгоритм
DES, 422
MD4, 423
MS-CHAP, 433
RC4, 401
SHA-1, 423
round-robin fashion, 262
быстрой повторной передачи, 226
параметр реестра TcpMaxDupAcks, 228
временных отметок, 224
масштабирования окна, 212
медленного старта, 229
алгоритмы маршрутизации, 273
внутридоменные, 275
динамические, 274
междоменные, 275
многоадресные, 275
одноадресные, 275
одномаршрутные, 274
статические, 274
стоимость доставки, 273
требования, 273
анализатор трафика
Ethereal, 143
аппаратный адрес, 146
асинхронный порт (TTY), 50
атака
address spoofing, 358
ARP spoofing, 123
Cisco ident, 362
fraggle, 360
RIP Entry Added, 306
RIP Entry Timeout, 306
Smurf, 156, 158
основанная на фрагментации IP-дейтаграмм, 362
атаки типа DoS, 194, 241
Land, 201
Ping Of Death, 363
smurf, 359
stream.c, 201
TCP SYN flooding, 194, 195, 364
DynamicBacklogGrowthDelta, 199
EnableDynamicBacklog, 198
MaximumDynamicBacklog, 199
атаки типа DoS (*продолжение*)
MinimumDynamicBacklog, 198, 200
SynAttackProtect, 196, 197
TcpMaxConnectResponseRetransmissions, 197
TcpMaxHalfOpen, 196
TcpMaxHalfOpenRetried, 196, 197
TearDropFragmentation, 75
шторм запросов TCP SYN, 194, 195, 196, 364
аутентификация, 387
MS-CHAP v2, 418
по протоколу RADIUS, 368

Б

баннеры (заголовки), 66
безадресные интерфейсы, 317
безопасность
доступа к маршрутизатору, 68
сети, составляющие, 362
система обнаружения атак, 363
удаленного доступа
авторизация, 366
аутентификация, 365
настройка технологии AAA, 369
технология AAA, 365
учет, 366
бесклассовая маршрутизация, 109
бит
DE, 324
DF, 46, 235
синхронизации, SYN, 190
буфер истории Cisco IOS, 34
буферы сетевых пакетов, 53, 54
проверка состояния, 53
счетчик попаданий, 55
быстрая повторная передача, 226
параметр реестра TcpMaxDupAcks, 228

В

вес пакета, 271
виртуальное соединение Frame Relay, 321
постоянное, 321
виртуальный порт (VTY), 50
внутренние адреса, 238
глобальные, 241
частные, 241
внутренние часы, 58
настройка, 58
синхронизация на всех маршрутизаторах, 59
внутридоменные алгоритмы маршрутизации, 275
временная синхронизация, 176
временные отметки, 176
временных отметок, алгоритм, 224
RTO, 225
RTT, 225
параметр реестра
TcpMaxConnectRetransmissions, 225
TcpMaxDataRetransmissions, 225

временных отметок, алгоритм (*продолжение*)
поле
Timestamp Echo Reply, 224
Timestamp Value, 224
встроенный отладчик Cisco IOS, 137
выбор
маршрута, 275
стратегии очередей, 271

Г

глубина очереди
настраиваемой, 264
с приоритетами, 261
группы пользователей, 388

Д

дейтаграмма, 147
IP, 72
формат заголовка, 73
динамическая трансляция адреса, 242
динамические алгоритмы маршрутизации, 274
доступ к маршрутизатору
технология AAA, 370
драйвер WinPcap, 142

Ж

журнал
на маршрутизаторе, 57
уровни сообщений, 57
отбрасывания трафика по правилам списков доступа, 357

З

завершение
сеанса, принудительное, 132
сеанса Telnet, 135
заголовки (баннеры), 66
Exec, 66
Login, 66
MOTD, 66, 67
настройка, 67
заголовок IP
бит DF, 46, 235
поле IP Precedence, 249, 251
поле TOS, 250
задержка выпуска в очереди FIFO, 247
запись таблицы маршрутизации, 293, 300
защитные экраны
опыт эксплуатации, 386
защитный экран, 201
Checkpoint Firewall-1, 358
Cisco IOS Firewall Feature Set, 358
Cisco PIX Firewall, 246
GuardianPro, 243
v5, 243
v5.02, 364
v5.0x, 201
технология NAR, 245
защитный экран (*продолжение*)
Kerio WinRoute Pro, 385, 408, 410
Microsoft ISA Server, 386, 412
журнал работы, 387
защита от атаки TCP SYN flooding, 201
контроль действий пользователя, 387
доступа к сайтам Интернета, 387
кэширование запросов, 386
настройка, 388
для VPN, 409
правила фильтрации, 387

И

имя
маршрутизатора
ограничения длины, 66
по умолчанию, 66
рекомендации RFC 1035, 23
инкапсуляция, 74
ARPA, 121, 130
интерфейс
10BaseT, 376
недостатки, 378
DOD (Dial-on-Demand), 390
RS-232, 377
Serial
подключение, 42
типы кабелей, 42
V.35, 16, 311, 375, 378
длина кабеля, 15
командной строки Cisco IOS, 27
настройка описания, 131
пропускная способность, 45
просмотр информации о настроенном протоколе, 56
статистика работы, 43
статус, 43, 44
канального уровня, 43
физического уровня, 43
интерфейсы, 18
аппаратные, 40
именование и нумерация, 18
логические, 129
loopback, 129
tunnel, 129
подинтерфейсы, 130
физические, 128

К

кадр, 147
кадр Frame Relay
бит
BECN, 325
FECN, 325
структура, 322

калькулятор хэш-значений, 422
каналы связи, избыточные, 285
качество канала связи, 312, 379
  характеристики провода, 376
качество обслуживания в IP-сетях, 249
  поле
    IP Precedence, 249, 251
    TOS, 250
  рекомендации RFC 1812, 251
клавиша Tab, при вводе команд, 37
ключевое слово
  any, 351
  chap, 369
  config, 32
  default, 370
  DF, 169
  exec, 372
  host, 351
  local, 369
  log, 357
  no, 37
  no ip unreachable, 168
  none, 372
  overload, 241
  password, 371
  radius-server, 373
  running-config, 65
  start-stop, 372
  startup-config, 65
  tacacs+, 371
  tacacs-server, 373
команда
  aaa accounting, 372
  aaa authentication, 369
  aaa authorization network, 372
  aaa new-model, 369
  access-list, 348, 355
  arp timeout, 121, 124
  bandwidth, 45, 229
  cdp holdtime, 136
  cdp timer, 136
  clear arp-cache, 121
  clear host, 128
  clear line, 132
  clock, 36
  configure memory, 62
  configure network, 63
  configure terminal, 31, 37, 62
  configure overwrite-network, 66
  custom-queue-list, 265
  debug, 137
  debug ip rip, 296
  description, 131
  disable, 30
  enable, 30
  encapsulation, 131
  encapsulation frame-relay, 326, 337
команда (*продолжение*)
  end, 63
  erase, 66
  exec timeout, 38
  exec-timeout, 132
  exit, 32
  fair-queue, 268
  frame-relay adaptive-shaping, 339
  frame-relay class, 338
  frame-relay custom-queue-list, 343
  frame-relay interface-dlci, 331
  frame-relay traffic-rate, 338
  group-range, 370
  hostname, 66
  interface, 31
  ip access-group, 351
  ip address, 117
  ip host, 127
  ip ICMP rate-limit unreachable, 169
  ip mask-reply, 180
  ip nat outside, 242
  ip redirects, 176
  ip rsyp bandwidth, 229
  ip tcp intercept list, 201
  ip tcp path-mtu-discovery, 234
  ip tcp selective-ack, 220
  ip tcp timestamp, 221
  ip tcp window-size, 214
  ip unnumbered, 118, 317
  line, 132
  login, 68
  map-class frame-relay, 337
  network, 299, 306
  no cdp enable, 136
  no cdp run, 136
  no ip domain-lookup, 126
  no ip proxy-arp, 122
  password, 68
  ping, 127, 236
  ppp authentication, 370
  priority-group, 259, 260
  priority-list, 258
  queue-list, 263
  queue-list queue limit, 266
  radius-server host, 373
  random-detect, 255, 256
  random-detect precedence, 255
  redistribute static, 305
  resume, 134
  router igrp, 306
  service password-encryption, 70, 371
  service timestamps log datetime, 358
  show, 37
  show arp, 121
  show buffers, 53
  show cdp entry, 136
  show cdp neighbors, 135
  show cdp neighbors detail, 136

команда (*продолжение*)
show clock, 58
show controllers, 41
show flash, 39
show frame-relay pvc, 332
show hosts, 128
show interface serial1, 311
show interfaces, 121
show ip interface, 49
show ip route, 56
show ip route summary, 56
show line, 51
show logging, 357
show memory, 53
show process memory, 52
show processes cpu, 52
show protocols, 56
show queue, 268
show queuing priority, 260
show running-config, 60, 70
show sessions, 134
show sntp, 59
show stacks, 55
show startup-config, 60
show traffic-shape, 342
show user, 51
show users, 133
show version, 38
terminal, 34
terminal length, 38
terminal monitor, 137
terminal no history, 35
traceroute, 127
tracert, 162, 166, 182, 313
traffic-shape rate bit-rate, 336
username, 371
write erase, 66
write memory, 64, 65
write network, 64
команды
длина, 34
отмена введенной ранее, 37
получение подсказки, 35
получение справки, 35
принцип сокращения, 35
проверки аппаратных ресурсов, 52
коммуникационные шкафы, 380
коммутаторы Hewlett-Packard, 377
компания
RAD Data Communication, 307
RSA Data Security Inc., 401
SurfControl, 387
консольный порт (CTY/CON), 18, 50
защита доступа, 28
кабель, 19
контроллеры, 41
просмотр информации, 41
конфигурационный регистр, 39
конфигурация
загрузочная, 20, 59
просмотр, 60
удаление, 66
хранение, 60
рабочая, 20, 59
копирование в загрузочную, 59
обновление, 62, 63
полная замена, 61
просмотр, 60
резервирование, 64, 65
удаленных маршрутизаторов, 21
файлы, 20
корпоративная сеть, требования к технологии, 323

Л

логические интерфейсы, 129
loopback, 129
tunnel, 129

М

маршрут
критерии выбора, 275
метрика, 14
по умолчанию, 302
настройка, 303
статический, 399
маршрутизатор, 14
Cisco 1720, 310
безопасность доступа, 68
буферы сетевых пакетов, 53, 54
ведение журнала, 57
получатели сообщений, 58
уровни сообщений, 57
имя
ограничения длины, 66
по умолчанию, 66
рекомендации RFC 1035, 23
интерфейсы, 18
логические, 129
настройка описания, 131
физические, 128
классы, 14, 16
комплектация, 310
настройка
начальная, 22
пошаговая, 25
память, 17
NVRAM, 18, 60
RAM, 18
ROM, 18
флэш-память, 18
подинтерфейсы, 130
схема именования, 130
проверка аппаратных ресурсов, 52

маршрутизатор (*продолжение*)
работа со встроенным отладчиком, 137
режимы работы, 33
таблицы маршрутизации, 55, 56
управление через сеанс Telnet, 133
маршрутизация
административно заданное расстояние, 300
значения по умолчанию, 301
алгоритмы, 273, 274
внутридоменные, 275
динамические, 274
междоменные, 275
многоадресные, 275
одноадресные, 275
одномаршрутные, 274
статические, 274
требования, 273
бесклассовая, 109
в виртуальных частных сетях, 396, 399
выбор маршрута, 275
динамическая, 310
избыточные каналы связи, 285
маршрут по умолчанию, 302
настройка, 303
перенаправление, 172
протоколы, 275
длины вектора, 275
политики маршрутизации, 276
состояния канала, 276
стоимость доставки пакета, 273
счет до бесконечности, 289
технология
мгновенного обновления, 291
расщепления горизонта, 290
маска подсети, 84, 178
для стандартных классов IP-адресов, 84
механизм формирования, 85
переменной длины, 96, 179
преимущества, 97
условия, 100
масштабирование скользящего окна, 212
в Windows 2000, 213
мгновенное обновление (triggered updates), 291
медленный старт, 229
междоменные алгоритмы маршрутизации, 275
метрика маршрута, 14, 300
механизм
ip unnumbered, 118
Path MTU, 46
Path MTU Discovery, 234
параметр реестра
EnablePMTUBHDetect, 236
параметр реестра
EnablePMTUDiscovery, 236
SACK, 219
TCP Intercept, 200
разрешения адресов ARP, 120
механизм (*продолжение*)
скользящих окон TCP, 205
узкие места в сети, 208
фрагментации, 183
многоадресные алгоритмы маршрутизации, 275
модель OSI, 140, 382
модем
ASMi-50, 307
FlexDSL, 376
PairGain, 376
Watson, 310, 376
Натекс FlexDSL, 310
модемы
интерфейсы, 376
регулировка скорости, 377

Н

надежность виртуального соединения (VPN), 413
настраиваемые очереди, 261
глубина, 264
настройка
Frame Relay, 326
управление трафиком, 336, 337, 338, 340
адресного пространства, 313
адресов серверов DNS, 127
внутренних часов, 58
заголовков (баннеров), 67
защитного экрана
рекомендации, 181, 188
маршрута по умолчанию, 303
маршрутизатора
рекомендации, 164, 168, 179
начальная, 22
пошаговая, 25
модемов
рекомендации, 312
Натекс, 310
очередей, 246
RED, 255
WRED, 255
взвешенной справедливой, WFQ, 267
на основе классов, 267
настраиваемых (custom queuing), 261
с приоритетами, 258, 261
подинтерфейса, 331
протокола
IGRP, 306
IP, 116
RIP, 299
терминальных портов, 132
технологии AAA, 369
авторизация, 371
ведение учета, 372
список аутентификации, 369
туннеля PPTP, 390

немаршрутизируемые протоколы, 274
номер подсети, ограничения, 87

О

обновление
  загрузочной конфигурации, 65
    с сервера TFTP, 65
  рабочей конфигурации
    из флэш-памяти, 63
    командой configure memory, 62
    командой copy startup-config running-config, 62
    с сервера rcp, 63
    с сервера TFTP, 62, 63
обратное разрешение адресов, 120
одноадресные алгоритмы маршрутизации, 275
одномаршрутные алгоритмы маршрутизации, 274
окно, скользящее, 205
  узкие места в сети, 208
  изменение размера на маршрутизаторе, 214
  масштабирование, 212
описание интерфейса (description), 131
организация сети Fame Relay, 323
отключение
  рассылки сообщений протокола RIP, 295, 296
открытые адреса, 238
отладчик Cisco IOS, встроенный, 137
отображение портов, 408
отчеты по группам пользователей, 388
очереди
  FIFO, 246
    задержка выпуска, 247
    перегрузка канала связи, 247
  RED, 252
    настройка, 255
  WRED
    настройка, 255
  взвешенная справедливая (WFQ), 267
    вес пакета, 271
    настройка, 267
    режим DWFQ, 271
  выбор стратегии, 271
  на основе классов (CBQ), 267
  настраиваемые, 261
    глубина, 264
    и очереди с приоритетами, 262
  настройка, 246
  проблема голодания
    в настраиваемой очереди, 261
    в очереди с приоритетами, 258
  с приоритетами, 256
    глубина очереди, 261
    настройка, 258
    список приоритетов, 258

П

пакет, 147
память
  NVRAM, 18, 60
  RAM, 18
  ROM, 18
  проверка состояния, 52, 53
  флэш-память, 18
параметр реестра
  DynamicBacklogGrowthDelta, 199
  EnableDynamicBacklog, 198
  EnablePMTUBHDetect, 236
  EnablePMTUDiscovery, 236
  GlobalMaxTcpWindowSize, 214
  MaximumDynamicBacklog, 199
  MinimumDynamicBacklog, 198, 200
  SackOpts, 220
  SynAttackProtect, 196, 197
  Tcp1323Opts, 213, 214
  TcpDelAckTicks, 221
  TcpipbleICMPRedirects, 173
  TcpMaxConnectResponseRetransmissions, 197
  TcpMaxConnectRetransmissions, 225
  TcpMaxDataRetransmissions, 225
  TcpMaxDupAcks, 228
  TcpMaxHalfOpen, 196
  TcpMaxHalfOpenRetried, 196, 197
  TcpWindowSize, 212, 213
пароль
  enable password, 30, 69
  enable secret password, 30, 69, 70
  пользовательского режима, 68
  шифрование
    one-way, 70
    two-way, 70
перегрузки в IP-сети
  контроль, 228
  предотвращение, 229
  протокол RSVP, 228
передача
  повторная
    алгоритм быстрой повторной передачи, 226
    таймер, 221
  широковещательная, 80
    направленная, 80
    ограниченная, 81
переменные TCB, 204
  RCV.NXT, 204
  SND.NXT, 204
  SND.UNA, 204
перенаправление
  сообщений, 395
  трафика, 318
перехват трафика
  фильтр, 142
Петербургская телефонная сеть, 310
подавление источника, 160
подавление сеанса Telnet, 133

подинтерфейсы, 130
настройка, 331
схема именования, 130
подсеть
маска
для стандартных классов IP-адресов, 84
механизм формирования, 85
переменной длины, 96, 97, 100
формирование, 82
трехуровневая концепция, 82
поле
DLCI, 322
FCS, 322
IP Precedence, 249, 251
Timestamp Echo Reply, 224
Timestamp Value, 224
Time-to-Live, 181
TOS, 250
полносвязная топология в сети Frame Relay, 328, 329
порт
1900, 170
80, 387
8080, 387
порты
терминальные, 50, 68
AUX, 19, 50
асинхронный (TTY), 50
виртуальный (VTY), 50
кабель консольного порта, 19
консольный (CTY/CON), 18, 28, 50
настройка, 132
пошаговая настройка маршрутизатора, 25
префикс, сетевой, 77
расширенный, 84
приглашение
Cisco IOS, 29
в глобальном режиме конфигурации, 31
в режиме подконфигурации, 32
проверка
аппаратных ресурсов, 52
память, 52, 53
процессор, 52
состояния буферов сетевых пакетов, 53
программа
Cisco CallManager, 378
Cisco ConfigMaker, 22, 67, 117, 119
Firewalk, 183
nmap, 154, 187
routed, 292
SuperScoute for ISA Server, 387
System Configuration Dialog, 25
tracert, 188
WinDump, 414
прокси-сервер, 387, 388
пропускная способность
TCP, максимальная, 210
методы определения, 210
канала связи на интерфейсе, 45
просмотр
информации об интерфейсе, 56
конфигурации
рабочей, 60
загрузочной, 60
поддерживаемых протоколов, 56
списков доступа, 350
таблицы
маршрутизации, 55
протокола CDP, 135
протокол
ARP, 119, 120
proxy ARP, 120, 121
reverse ARP, 120
атака ARP spoofing, 123
разрешение адресов, 120
сообщения, 122
BGP-4, 113
CDP, 135
просмотр информации о соседних маршрутизаторах, 135
CHAP, 369
DHCP, 178
EIGRP, 96
GRE, 391
формат пакета, 392
HDLC, 311
ICMP, 139, 235, 242
IGRP, настройка, 306
IP, 72
атака TearDrop Fragmentation, 75
дейтаграмма, 72
настройка на маршрутизаторах Cisco, 116
списки доступа, 346
формат адреса, 77
IPCP, 424
IPSec, 382, 383
IS-IS, 100
L2F, 382, 383
L2TP, 383
L2TP , 382
LMI, 327
MS-CHAP, 418, 421
OSPF, 96, 100, 310, 311, 312, 319
PAP, 369
PPP, 369, 383, 391
PPP LCP, 418
PPTP, 374, 382, 383, 389
аутентификация удаленного партнера, 417
знаки внимания, 420
недостатки, 405

протокол (*продолжение*)
сжатие данных, 401
сообщение
OUTGOING-CALL-REPLY, 416
OUTGOING-CALL-REQUEST, 416
START-CONTROL-REPLY, 415
START-CONTROL- REQUEST, 414
уровень безопасности шифрования, 424
фильтрация трафика, 410
шифрование данных, 401
Probe, 120
RADIUS, 366
аутентификация, 368
взаимодействие клиент-сервер, 367
и протокол TACACS+, 368, 369
RIP, 88
атака
RIP Entry Added, 306
RIP Entry Timeout, 306
мгновенное обновление, 291
настройка, 299
отключение рассылки сообщений, 295, 296
расщепление горизонта, 290
RIP, 276
RIP-1, 96, 97, 295
RIP-2, 100
RSVP, 228
SNMP, 378
SSDP, 170
TACACS+, 366, 368
и протокол RADIUS, 368, 369
TCP, 189
Path MTU Discovery, 234
алгоритм временных отметок, 224
алгоритм масштабирования окна, 212
атака Land, 201
атака stream.c, 201
атака TCP SYN flooding, 194, 195, 196
блок управления передачей (TCB), 204
выборочное подтверждение (SACK), 219
задержанное подтверждение, 221
квазисвободное соединение, 199
механизм скользящего окна, 205
пропускная способность, 210
расширения для Windows 2000, 219, 221
стратегии повторной передачи, 218
таймер повторной передачи, 221
узкие места в сети, 208
управление потоком, 214
установление соединения, 191
флаг PSH, 217
флаги заголовка, 190
UDP, 386
протоколы
немаршрутизируемые, 274
просмотр поддерживаемых протоколов, 56
протоколы (*продолжение*)
маршрутизации, 275, 276
длины вектора, 275, 293
политики маршрутизации, 276
состояния канала, 276
процессор, проверка состояния, 52
прямой провод
простая проверка, 376

P

размер сегмента, максимальный (MSS), 212
разрешение адресов, 120
ARP, 120
локальная таблица соответствия, 127
обратное, 120
разъем
DB-25, 42
V.35, 42
расстояние, административно заданное значения по умолчанию, 301
расстояние, административное, 300
расстояние, заданное административно, 300
расширение TCP для Windows 2000
выборочное подтверждение (SACK), 219
параметр реестра SackOpts, 220
задержанное подтверждение, 221
параметр реестра TcpDelAckTicks, 221
расширенные списки доступа, 354
расширенный сетевой префикс, 84
расщепление горизонта (split horizon), 290
регистр конфигурации, 39
режимы
конфигурации
Cisco IOS, 29, 31
интерфейсов, 31
портов, 68
подконфигурации, 31
работы маршрутизатора, 33
ROM Monitoring, 32
глобальный, 31
пользовательский, 29
привилегированный, 30
пароль, 30
разрешения, 30
резервирование каналов связи, 310, 318
резервирование рабочей конфигурации
в памяти NVRAM
командой copy running-config startup-config, 64
командой write memory, 64
во флэш-памяти, 65
на сервере rcp, 65
на сервере TFTP
командой copy running-config tftp, 64
командой write network, 64
репликация информации домена, 411

С

сеанс
  Telnet
    завершение, 135
    подавление, 133
    управление маршрутизатором, 133
  принудительное завершение, 132
  просмотр отладочной информации, 137
сегмент TCP, 147
сервер
  DNS, настройка адресов, 127
  RADIUS, 367
  RARP, 120
  rcp, 61
  Syslog, 58
  TACACS+, 371
  VPN, 389, 411
    размещение, 411
  авторизации, внешний, 69
сервер имен, 170
сетевой префикс, 77
  расширенный, 84
сеть
  Frame Relay
    организация, 323
    подинтерфейсы, 130
  IP, качество обслуживания, 249
    поле IP Precedence, 249, 251
    поле TOS, 250
    рекомендации RFC 1812, 251
  IP, контроль перегрузок, 228
  IP, предотвращение перегрузок
    алгоритмы, 229
    медленный старт, 229
  корпоративная, требования к технологии, 323
синхронизация внутренних часов, 59
система SuperScout, 388
система обнаружения атак, 363
  Snort, 144
система обнаружения атак (IDS), 361
система обнаружения вторжений (IDS), 198
сканирование
  определение доступных портов, 181
  определение номеров открытых портов, 187
  определение схемы маршрутизации, 179
  определение типов операционных систем, 178
сканирование портов
  подстановка адресов, 171
скользящее окно, 205
  в Windows 2000, 213
    GlobalMaxTcpWindowSize, 214
    Tcp1323Opts, 213, 214
    TcpWindowSize, 212, 213
  изменение размера на маршрутизаторе, 214
  масштабирование, 212
  узкие места в сети, 208
служба
  DNS, 126
  RRAS, 386, 389, 390
согласования параметров соединения
  протокола PPP, 418
соединение
  Frame Relay, мультиплексирование, 321
  PVC, 130
  TCP
    квазисвободное, 199
    установление, 191
сообщение
  IPCP Configuration Ack, 425
  PPP CHAP Success, 422
  PPP IPCP Request, 424
  PPP LCP Configuration Request, 418
сообщения
  keepalive, 43, 319, 391
  Syslog, 58
  инкапсуляция ICMP, 140
  маршрутизатора
    получатели, 58
    уровни, 57
  протокола RIP, отключение рассылки, 295, 296
сообщения ICMP redirect, 172
списки доступа, 344
  ведение журнала, 357
  входящие, 345
  диапазоны номеров, 347
  добавление правил, 350
  задачи, 344
  защита от атаки
    address spoofing, 358
    Cisco ident, 362
    fraggle, 360
    Ping Of Death, 363
    smurf, 359
    основанной на фрагментации IP-дейтаграмм, 362
  исходящие, 345
  маска шаблона, 348
    и маска подсети для протокола IP, 350
  просмотр, 350
  протокола IP, 346
  разрешающее правило (permit), 351
  расширенные, 354
  создание правил, 348
  счетчики пакетов, 358
список
  аутентификации, 369
  приоритетов, 258
статистика работы интерфейса, 43
статическая трансляция адреса, 242
статические алгоритмы маршрутизации, 274
статус интерфейса, 43
стоимость доставки пакета, 273

стратегии повторной передачи в TCP
индивидуальная, 218
пакетная, 218
только первый, 218
счет до бесконечности, при обрыве в сети, 289
счетчики пакетов, для списков доступа, 358

Т

таблица маршрутизации, 55, 56, 173, 319
добавление маршрута, 173
запись, 293, 300
петля, 287
просмотр, 55
содержимое, 273
эффект прыжка, 287
таблица протокола CDP, 135
таймер повторной передачи, 221
RTO, 223
оценка SRTT, 223
телефон Cisco ATA 186, 379
терминальные порты, 50
настройка, 132
технология
AAA, 365
ведение учета, 372
контроль доступа к маршрутизатору, 370
настройка, 369
настройка авторизации, 371
список аутентификации, 369
ABC, 377
ADSL, 308
практика применения, 384
Frame Relay, 320
в корпоративных сетях, 323
в сравнении с VPN, 377
виртуальное соединение, 321
гарантированный объем кадров Bc, 326
и ATM, 322
и X.25, 321
и протокол LMI, 327
мультиплексирование, 321
настройка, 326
организация сети, 323
параметр Be, 326
параметры сервиса, 326
полносвязная топология сети, 328, 329
постоянное виртуальное соединение, 321
пропускная способность CIR, 324, 326
расщепление горизонта, 330
структура кадра, 322
управление трафиком, 333, 334
MPPC, 401
MPPE, 401
настройка ключей, 402
NAR, 245
NAT, 238, 312
совместно с технологией PAT, 315
таймер неактивности, 240
NAT/PAT, 408
технология (*продолжение*)
PAT, 166
SACK, 414
TC-PAM, 310
VoIP, 378
VPN
и Microsoft Windows 2000, 386
xDSL, 286
преимущества, 375
бесклассовой маршрутизации (CIDR), 109
виртуальной частной сети (VPN), 374
мгновенного обновления (triggered updates), 291
наибольшего совпадения, 108
расщепления горизонта (split horizon), 290
в сети Frame Relay, 330
топология, полносвязная, в сети Frame Relay, 328, 329
трансляция адреса, 238
динамическая, 242
приоритеты, 244
статическая, 242
таймер неактивности, 240
трассировка пакетами UDP, 183
туннель
PPTP, 391
восстановление при обрыве, 432
криптоустойчивость, 405
параметры безопасности, 432
параметры учетной записи, 429
политика удаленного доступа, 435
рабочий режим, 426
установка, 416
VPN, 389
шифрование, 401

У

удаленный доступ, 365
авторизация, 366
аутентификация, 365
учет, 366
управление потоком в TCP, 214
управление трафиком Frame Relay, 333, 334
базовая реализация GTS, 334
и FRTS, 334
настройка, 336
настройка, 338, 340
технология CAR, 334
технология FRTS, 336
настройка, 337
установка и профилактика оборудования, 380
установка туннеля PPTP, 416
учет по технологии AAA, 372
start-stop, 372
stop-only, 372
wait-start, 372
настройка, 372

Ф

файлы
  Cisco IOS
    схема именования, 40
  конфигурации, 20, 59, 62
    host, 62
    network, 62
    резервные копии, 63
  образ Cisco IOS, 39
физические интерфейсы, 128
флаг ACK, 155
флаг DF, 165
флаг Don't fragment (DF), 187
флаг More Fragments (MF), 183
флаг RST, 155, 167
флаги заголовка TCP, 190
  ACK, 190
  PSH, 217
  RST, 362
  SYN, 190
флэш-память, 18
  для обновления рабочей конфигурации, 63
  проверка содержимого, 39
формирование подсетей, 82
  трехуровневая концепция, 82
фрагментация дейтаграмм IP, 183

Х

хэширование при шифровании, 420

Ч

часы, внутренние, 58
  настройка, 58
  синхронизация на всех маршрутизаторах, 59

Ш

широковещательная передача, 80
  направленная, 80
  ограниченная, 81
шифрование и дешифрование трафика, 388
шифрование
  Microsoft для канала «точка-точка», 383
  RAS, 401
  RC4, 401
  с разделяемым закрытым ключом, 401
шлюз по умолчанию, 173, 318, 394
шторм запросов TCP SYN, 194–196, 364