

4 **Зобнин, В. Л.** Дальнейшее развитие полигонных технологий как инструмента повышения внутренней эффективности / В. Л. Зобнин // Бюллетень Объединенного ученого совета ОАО «РЖД». – 2017. – № 5–6. – С. 40–52.

5 О едином сетевом технологическом процессе грузовых перевозок : распоряжение ОАО «РЖД». – М. : Транспорт, 2012. – 120 с.

6 **Грунтов, П. С.** Эксплуатационная надежность станции / П. С. Грунтов. – М. : Транспорт, 1986. – 247 с.

Получено 05.06.2023

ISSN 2227-1155. Сборник студенческих научных работ.
Вып. 28. Гомель, 2023

УДК 004.056

Я. В. ГУРИНОВИЧ (ЗмТ-56)

Научный руководитель – канд. техн. наук *В. О. МАТУСЕВИЧ*

SNMP-ОПРОС УСТРОЙСТВ С ПОМОЩЬЮ PYTHON

Произведено исследование процесса мониторинга по протоколу SNMP, написана программа для получения различных данных от сетевых устройств с помощью Python.

Постоянный мониторинг локальной сети, который является основой любой корпоративной сети, необходим для полного бесперебойного функционирования. Контроль сетевых устройств – необходимая и важная часть мониторинга, которую нужно реализовывать при управлении сетью. Из-за важности этой функции ее часто отделяют от других функций систем управления и реализуют с помощью специальных средств. Такое разделение функций управления и контроля обычно используют для сетей малого и среднего размера, для которых установка интегрированной системы управления не имеет экономического смысла. Использование автономных инструментов управления помогает сетевому администратору выявлять проблемные области или отдельные проблемы на сетевых устройствах, и в этом случае он может вручную отключить или перенастроить их.

Задачи мониторинга решаются с помощью программно-аппаратных измерительных приборов, тестеров, сетевых анализаторов, интегрированных устройств мониторинга устройств связи, а также агентов систем управления. Также требуется более активное участие инженера-администратора и использование сложных инструментов, таких как экспертные системы, которые накапливают практический опыт многих сетевых специалистов.

В рамках автоматического управления возможно организовать централизованное управление, при котором результаты параметров мониторинга представляются центральному подразделению. В известном и довольно простом алгоритме управления технологическим процессом центральное

управление замыкается на блок управления, образуя централизованную систему управления. Если процесс протекает относительно медленно, либо контролируемый процесс сложен, и законы, подлежащие контролю, еще не сформулированы, или сложно автоматизировать учет всех определяющих факторов, контроль осуществляется с участием одного человека. При централизованном управлении без вмешательства человека дисплей для отображения информации теряет значение для оператора, а центральной функцией управления является генерация сигналов для блока управления.

Simple Network Management Protocol (SNMP) – это протокол прикладного уровня, который позволяет обмениваться данными между сетевыми устройствами. Является набором правил. Он определен Советом по архитектуре Интернета и является частью пакета TCP / IP. SNMP, управляется Internet Engineering Group (IETF).

Протокол позволяет системному администратору контролировать производительность сети и изменять конфигурацию подключенных устройств. SNMP используется в сетях любого размера: чем больше сеть, тем лучше раскрываются преимущества протокола. Это позволяет просматривать, отслеживать и управлять узлами через единый интерфейс с функциями пакетной команды и автоматическим уведомлением.

SNMP избавляет администратора от необходимости вводить команды вручную. Всего было разработано и реализовано три версии. Все они используются до сих пор, и наиболее распространенным стал второй – SNMPv2c.

Компоненты, составляющие архитектуру SNMP (рисунок 1), следующие:

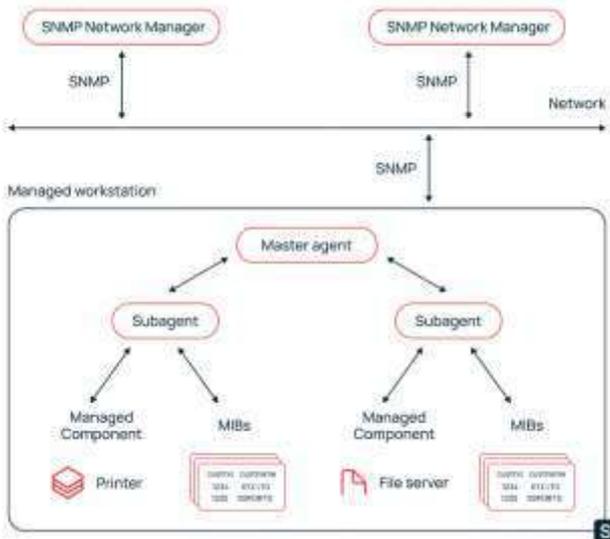


Рисунок 1 – Архитектура SNMP

- станция управления сетью, содержащая сетевой менеджер;
- агенты;
- основные агенты (Master agent);
- управляемые компоненты.

В настоящее время, когда программирование является одним из наиболее востребованных направлений в современном мире, в котором широко используются сетевые технологии, внедряются сети на основе намерений (ИБН) и виртуализация сетевых функций (NFV), протоколы RESTconf / NETconf, данные json, вводится язык моделирования и т. д. Чтобы соответствовать новым требованиям, сетевым инженерам необходимы новые компетенции, знания для программирования и автоматизации сетей в будущем.

Python – популярный язык программирования в различных областях, включая сетевую индустрию. Многие современные сетевые устройства поддерживают API и взаимодействуют с ними через Python. Существует множество сетевых библиотек Python: telnetlib, netmiko, paramiko и т. д.

Применение знаний и навыков программирования, особенно на Python, позволяет автоматизировать многие рутинные задачи по настройке и обслуживанию сети, которая может состоять как из «старых» (устаревших) устройств, так и из современных устройств, поддерживающих парадигму сетевой программируемости и открытые API [2].

У каждого есть рабочие задачи, которые несложно выполнить один раз. Однако по мере того, как они становятся постоянными и отнимают много времени, их необходимо автоматизировать.

Автоматизируя эти задачи, вы можете перестать тратить свое время на рутину. Это позволяет им сосредоточиться на действительно важных вещах, что значительно повышает эффективность их работы.

Листинг программы:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# Подключение модулей
import re # Модуль re предоставляет операции сопоставления
шаблонов регулярных выражений, аналогичные тем, которые встречаются
в языке Perl.
import json # Модуль JSON – это стандарт обмена данными. Он
позволяет легко сериализовать и десериализовать объекты.
from sys import argv # Функция argv модуля sys возвращает спи-
сок параметров командной строки, передаваемых скрипту Python.
from pysnmp.entity.rfc3413.oneliner import cmdgen
import logging # Модуль logging – пакет для логирования в
стандартной библиотеке Python
# Установка логов
logging.basicConfig(
    level=logging.DEBUG, filename='./log.txt',
    format='%(asctime)s %(name)s.%(funcName)s +%(lineno)s:
%(levelname)-8s [%s] %(message)s',
)
```



```

f"{errorStatus.prettyPrint(), errorIndex and varBindTable[-
1][int(errorIndex) - 1] or '?'}")
# Таким образом, если сетевое устройство недоступно или отдало
некорректный ответ, мы логируем исключение.
# Если нет ошибок в полученном ответе - записываем все пара-
метры в словарь. В varBindTable содержится ответ, который предстоит
распарсить.
# И, т.к это список объектов специфического формата - обращение
к их обрабатываемому виду выглядит так:
for varBindTableRow in varBindTable:
    for name, val in varBindTableRow:
        founds_mt_responce =
self.re_mt.search(name.prettyPrint())
        if founds_mt_responce is not None:
            port =
founds_mt_responce.group("port")
            self.result.setdefault('sign',
{})[port] = founds_mt_responce.group("sign")
            self.result.setdefault('link',
{})[port] = val.prettyPrint()
            found_if_responce =
self.re_if.search(name.prettyPrint())
            if found_if_responce is not None:
                port = found_if_responce.group('port')
                type_response =
self.types_response.get(found_if_responce.group('key'))
                if (type_response in ['ifAdminStatus',
'ifOperStatus']) and (val.prettyPrint() == '1'):
                    status = 'up' if val.prettyPrint()
== '1' else 'down'

self.result.setdefault(type_response, {})[port] = status
                continue
            self.result.setdefault(type_response,
{})[port] = val.prettyPrint()
            except BaseException as bex:
                logger.error(bex)
            return self.result
if __name__ == "__main__":
    name_script, ip, ro, oid = argv
    device = Device(ip, ro, oid)
    print(json.dumps(device.get_ifwalk()))

```

СПИСОК ЛИТЕРАТУРЫ

- 1 Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих : пер. с англ. – М. : ООО «ИД Вильямс», 2017. – 592 с.
- 2 Python для сетевых инженеров [Электронный ресурс]. – Режим доступа : <https://pyneng.readthedocs.io/ru/latest/about.html/>. – Дата доступа : 24.05.2023.

Получено 25.05.2023