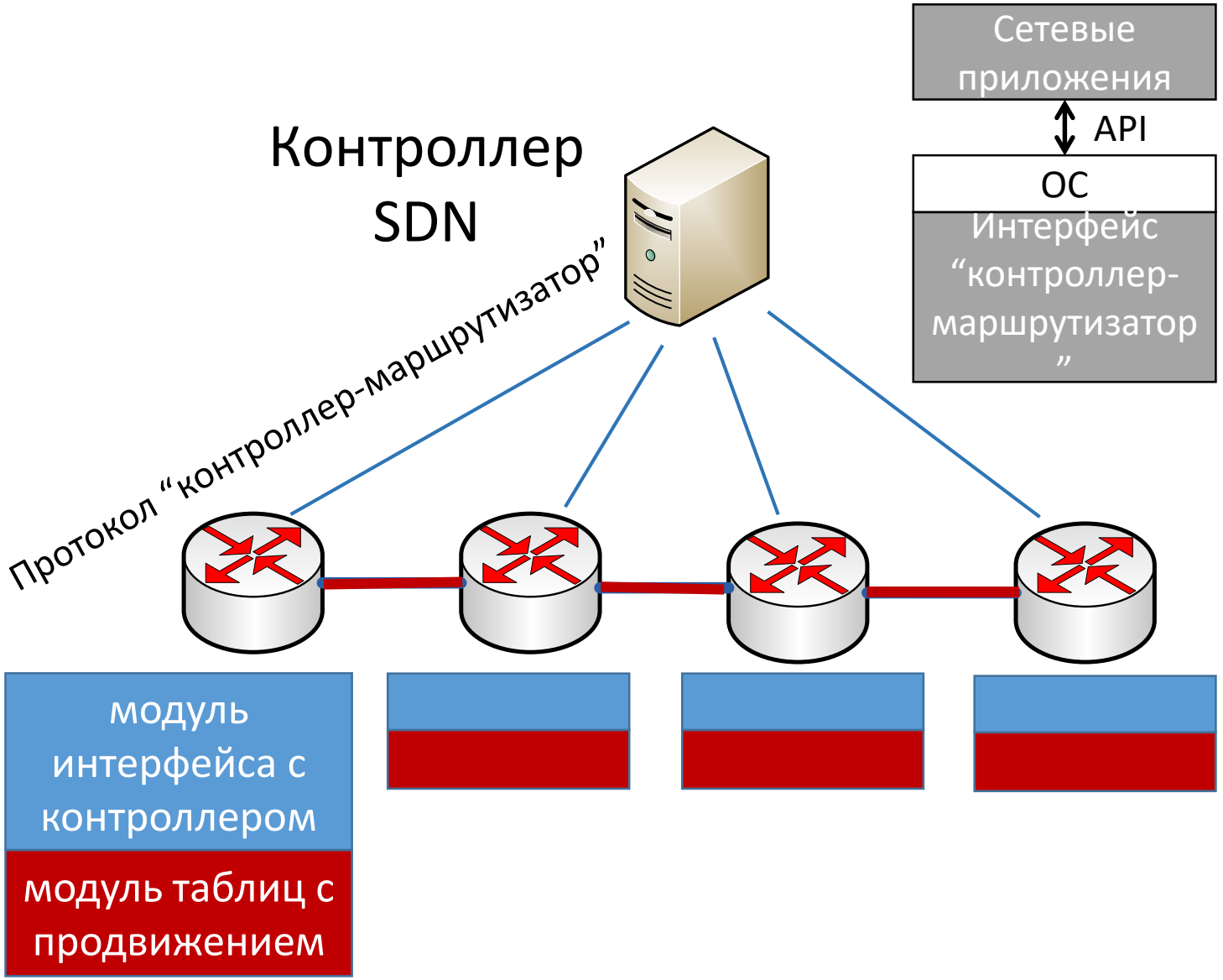
Лекция XII. SDN. ICMP. Протоколы управления: SNMP, NETCONF.

Курс читает Рогозин Н.О., кафедра ИУ-7


Контроллер SDN
Протокол “контроллер-маршрутизатор”
Сетевые приложения
API
ОС
Интерфейс “контроллер-маршрутизатор”
модуль интерфейса с контроллером
модуль таблиц с продвижением

SDN на основе OpenFlow v1 [RFC 7426]

- По версии OF 1.0 модель коммутатора использует одну таблицу продвижения, состоящую из ряда записей — правил обработки пакетов.
- В исходном состоянии таблица продвижения коммутатора пуста. Ее формирование — это обязанность приложений контроллера SDN.
- Полученные от приложений правила обработки пакетов контроллер передает коммутатору по протоколу OF.
- Помимо сообщений-правил в число возможных сообщений протокола OF входят также сообщения -запросы, с помощью которых контроллер запрашивает у коммутатора информацию о состоянии его портов (работоспособные или нет), а также статистику потоков.

SDN на основе OpenFlow v1

- В протоколе OF предполагается, что коммутатор не только отвечает на запросы контроллера, но может передать контроллеру сообщения по своей инициативе, например, в случае изменения состояния порта или удаления некоторого правила по таймауту.
- Канал обмена сообщениями между контроллером и коммутатором SDN называется **управляющим каналом**.
- Он представляет собой TCP-сессию, установленную в IP-сети контроллером и коммутатором.

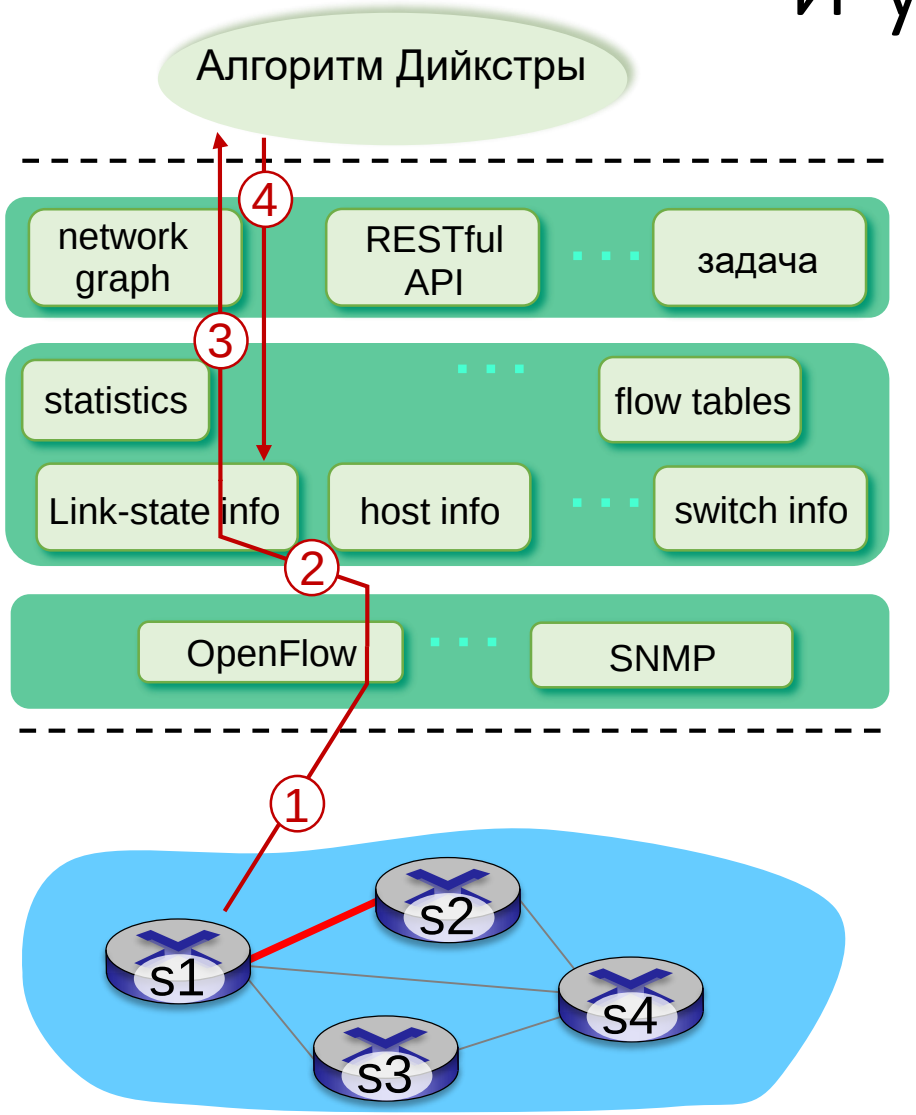
Принцип работы SDN на основе Openflow

- Определяется **правилами**, каждое правило состоит из:
  - **условий** выделения потока пакетов, к которым это правило должно быть применено;
  - **действий**, которые должны быть выполнены над пакетом, который удовлетворяет условиям данного правила;
  - **счетчиков**, измеряющих характеристики потока пакетов

Виртуальные порты

- Контроллер и его приложения должны иметь возможность реагировать на появление новых потоков в сети, иначе гибкость сети SDN не будет достаточной.
- С этой целью в технологии SDN предусмотрен виртуальный порт CONTROLLER
- Для того, чтобы пакеты, принадлежащие неизвестным коммутатору потокам (то есть для которых не нашлось условия в таблице, вызывающего совпадение) не отбрасывались, а обрабатывались особым, предусмотренным для них способом, необходимо поместить в таблицу следующее правило, имеющее нулевой приоритет:
  - Priority = O
  - Conditions: {}
  - Actions: {port=CONTROLLER}
- Если такое правило в таблице отсутствует, то все нераспознанные пакеты просто отбрасываются, но при его наличии они направляются в порт CONTROLLER.

SDN: пример взаимодействия уровней данных и управления



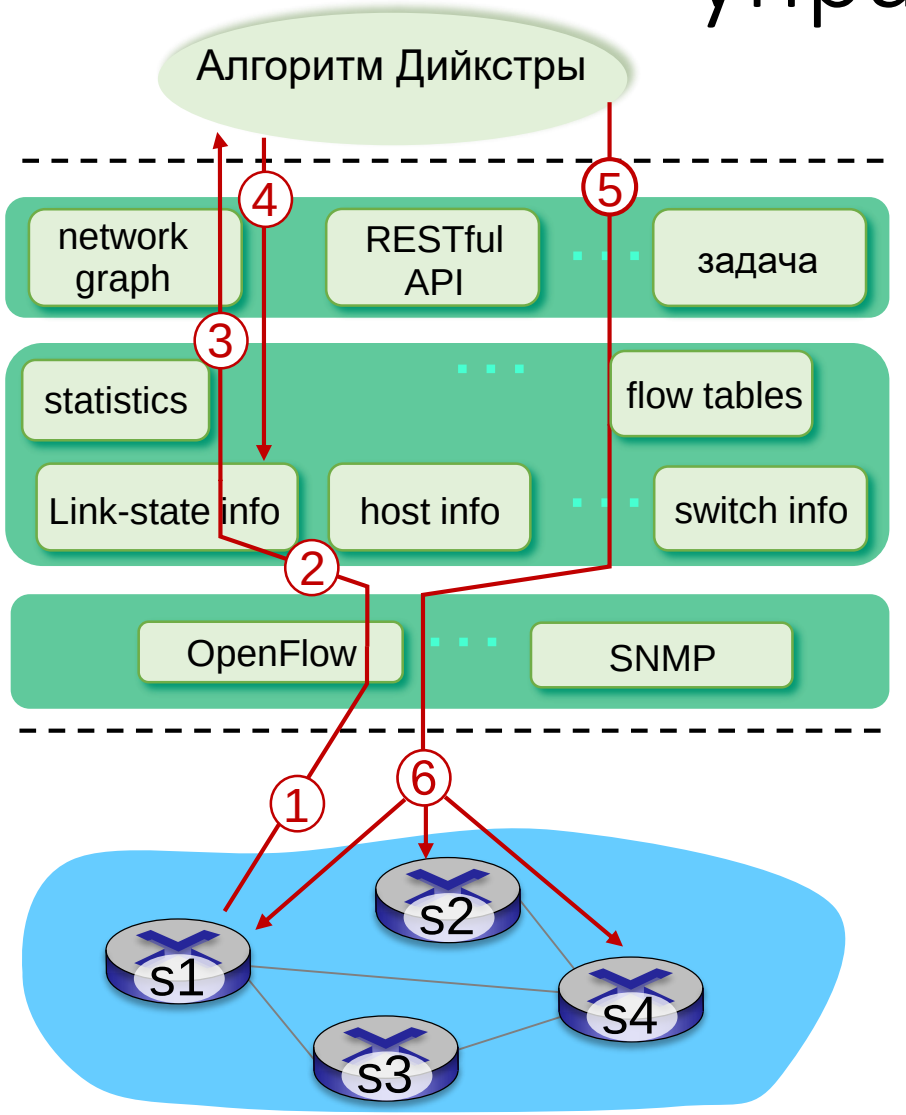
① S1, после падения канала использует статус порта OpenFlow для оповещения контроллера

② SDN контроллер получает OpenFlow сообщение, обновляет информацию о канале

③ Модуль алгоритма Дийкстры подписан на вызов в случае изменения состояния каналов. Происходит его вызов.

④ Модуль алгоритма Дийкстры получает информацию о графе сети, состоянии каналов от контроллера, вычисляет новые маршруты

SDN: пример взаимодействия уровней данных и управления



⑤ модуль маршрутизации передает информацию модулю вычисления таблицы потоков в SDN контроллере, последний вычисляет новые таблицы потоков

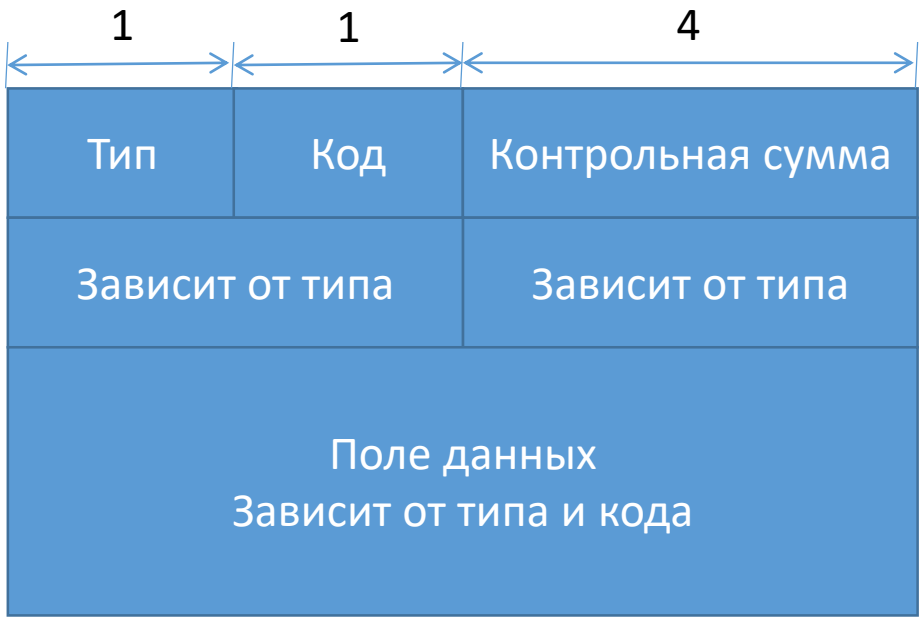
⑥ контроллер использует OpenFlow для установки новых таблиц в устройствах которым нужно обновление

ICMP [RFC 792]

- протокол межсетевых управляющих сообщений) — сетевой протокол, входящий в стек протоколов TCP/IP.
- Используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна, или хост, или маршрутизатор не отвечают.
- Также на ICMP возлагаются некоторые сервисные функции.

Заголовок ICMP

- **тип** (1 байт)
  - числовой идентификатор типа сообщения
- **код** (1 байт)
  - числовой идентификатор, более тонко дифференцирующий тип ошибки
- **контрольная сумма** (2 байта)
  - подсчитывается для всего ICMP-сообщения



Поле типа

Тип	Код	Описание
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Эхо-протокол

- Включает обмен двумя типами сообщений: эхо-запрос и эхо-ответ.
- Компьютер или маршрутизатор посылают по интерсети эхо-запрос, в котором указывают IP-адрес узла, достижимость которого нужно проверить.
- Узел, который получает эхо-запрос, формирует и отправляет эхо-ответ и возвращает сообщение узлу - отправителю запроса.
- В запросе могут содержаться некоторые данные, которые должны быть возвращены в ответе.
- Так как эхо-запрос и эхо-ответ передаются по сети внутри IP-пакетов, то их успешная доставка означает нормальное функционирование всей транспортной системы интерсети.

Примеры приложений

- ping
- traceroute/ tracert

Ключи запуска “ping”

- Ключ **-n** задает количество отправляемых эхо-запросов (по умолчанию 4).
- Ключ **–t** заставляет утилиту ping посылать запросы в бесконечном цикле до ее прерывания нажатием комбинации клавиш **<Ctrl-C>**.

```
D:\>ping bmstu.ru

Обмен пакетами с bmstu.ru [195.19.50.247] с 32 байтами данных:
Ответ от 195.19.50.247: число байт=32 время=5мс TTL=55
Ответ от 195.19.50.247: число байт=32 время=4мс TTL=55
Ответ от 195.19.50.247: число байт=32 время=8мс TTL=55
Ответ от 195.19.50.247: число байт=32 время=9мс TTL=55

Статистика Ping для 195.19.50.247:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 4мсек, Максимальное = 9 мсек, Среднее = 6 мсек

D:\>_
```

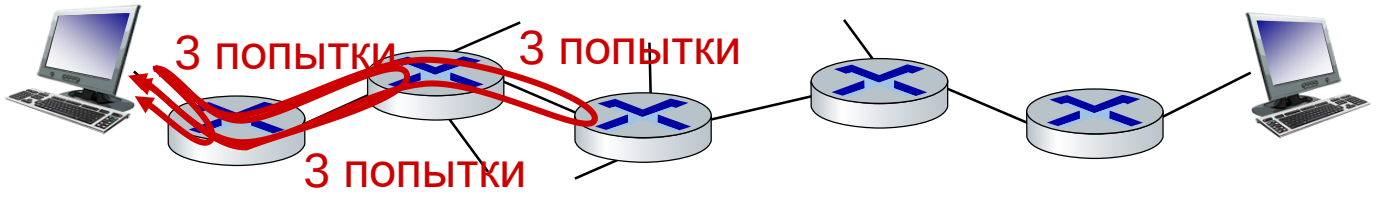
Ключи запуска “ping”

- Ключ **–l** задает размер дейтаграммы без учета длины заголовка (28 байт), посылаемой в эхо-запросе. Допустимыми являются значения от 0 до 65.500, включительно.
- Ключ **–f** устанавливает на дейтаграмме специальную пометку, запрещающую ее фрагментацию.
- Ключ **–i** задает TTL (***T**ime **T**o **L**ive*) пакета посылаемых дейтаграмм, измеряемое количеством узлов, которые может посетить пакет (по умолчанию 128).
  - Каждый промежуточный узел уменьшает значение TTL на единицу и, когда оно достигает нуля, пакет уничтожается

Утилита “traceroute/ tracert”

- Когда маршрутизатор не может передать или доставить IP-пакет, он отсылает узлу, отправившему этот пакет, сообщение о недостижимости узла назначения.
- Помимо причины ошибки, указанной в заголовке (в полях типа и кода), дополнительная диагностическая информация передается в поле данных ICMP-сообщения. Именно туда помещается заголовок IP и первые 8 байт данных того IP-пакета, который вызвал ошибку.
- Эта информация позволяет узлу-отправителю еще точнее диагностировать причину ошибки.

Утилита “traceroute/ tracert”



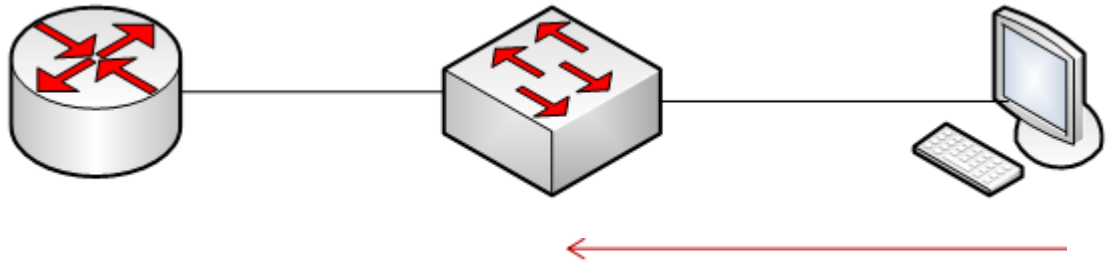
- Источник отправляет набор UDP пакетов получателю, 1 множество TTL =1, 2 множество TTL=2 и т.д.
- Пакет из набора n прибывает на роутер n:
  - роутер отбрасывает пакет и посылает отправителю ICMP пакет (тип 11, код 0)
  - ICMP пакет может включать имя маршрутизатора и IP-адрес

критерии остановки:

- UDP сегмент прибывает на узел назначения
- узел назначения возвращает ICMP сообщение “порт недоступен” (тип 3, код 3)
- отправитель перестает отправлять

- Когда ICMP прибывает на узел-отправитель, записывается RTT

ICMPv6 в SLAAC



Сообщение ICMPv6 **RA**
Включает: префикс, длину префикса. шлюз по умолчанию.
Используется multicast адрес IPv6 (**FF02::1**)
Рассылается всем узлам сети в группе

от хоста

от маршрутизаторов.

адрес IPv6 (**FF02::2**)

Сообщение ICMPv6 **RS**

Запрос сообщения RA

Используется multicast

Служба управления сетью

- Сложный программноаппаратный комплекс, который контролирует сетевой трафик и управляет коммуникационным оборудованием крупной компьютерной сети.
- Работает, как правило, в автоматизированном режиме, выполняя наиболее простые действия автоматически и оставляя человеку принятие сложных решений на основе подготовленной системой информации.

Задачи службы управления сетью

- **Управление конфигурацией сети и именованием** заключается в конфигурировании параметров как отдельных элементов сети, так и сети в целом.
- Для элементов сети (маршрутизаторы, мультиплексоры и т. п.) конфигурирование состоит в назначении сетевых адресов, идентификаторов (имен), географического положения и пр.
- **Обработка ошибок** включает выявление, определение и устранение последствий сбоев и отказов.

Задачи службы управления сетью

- **Анализ производительности и надежности** связан с оценкой на основе накопленной статистической информации таких параметров, как :
  - время реакции системы
  - пропускная способность реального или виртуального канала связи между двумя конечными абонентами сети
  - интенсивность трафика в отдельных сегментах и каналах сети
  - вероятность искажения данных при их передаче через сеть.

Задачи службы управления сетью

- **Управление безопасностью** подразумевает контроль доступа к ресурсам сети (данным и оборудованию) и сохранение целостности данных при их хранении и передаче через сеть.
- Базовыми элементами управления безопасностью являются процедуры
  - аутентификации пользователей
  - назначение и проверка прав доступа к ресурсам сети
  - распределение и поддержка ключей шифрования
  - управление полномочиями и т. п.

Задачи службы управления сетью

- **Учет работы сети** включает регистрацию времени использования различных ресурсов сети (устройств, каналов и транспортных служб) и ведение биллинговых операций (плата за ресурсы).

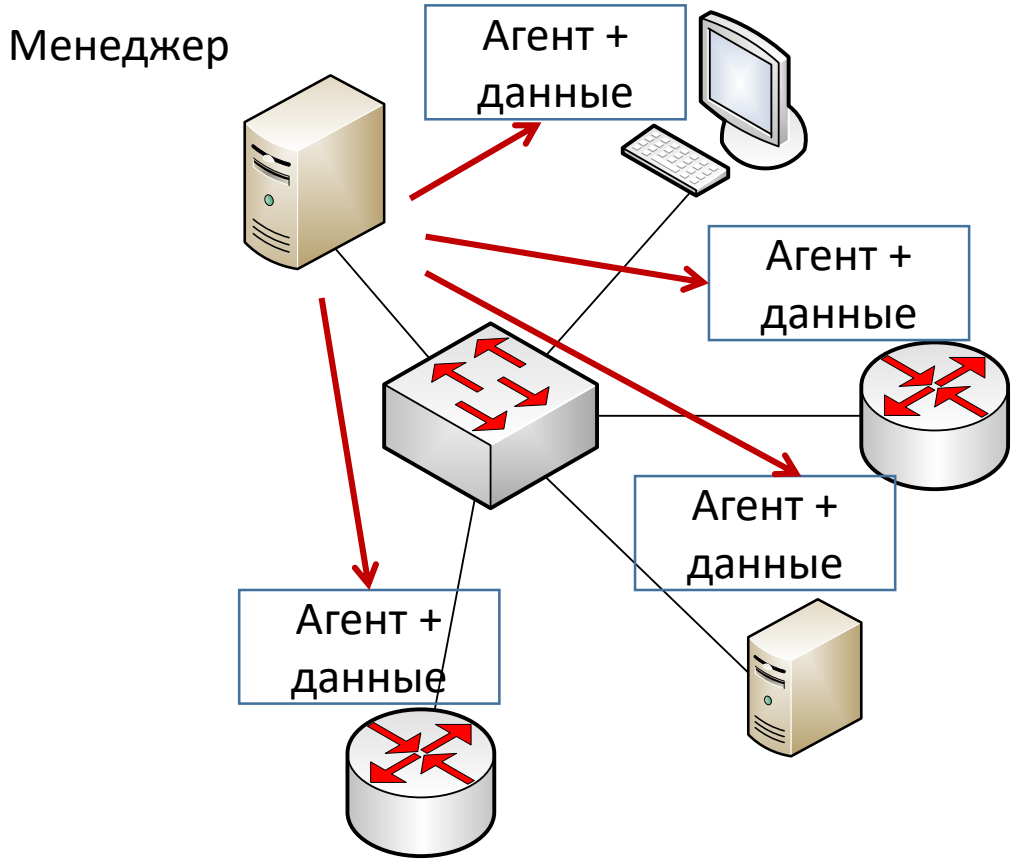
Агент

- каждое устройство, требующее достаточно сложного конфигурирования, производитель сопровождает автономной программой конфигурирования и управления, работающей в среде специализированной ОС, установленной на этом устройстве, называемый агентом.
- могут встраиваться в управляемое оборудование либо работать на устройстве, подключенном к интерфейсу управления такого устройства.

Функции агента

- **Хранить, извлекать и передавать по запросам извне информацию о технических и конфигурационных параметрах устройства**, включая модель устройства, число портов, тип портов, тип ОС, связи с другими устройствами и др.;
- **Выполнять, хранить и передавать по запросу извне измерения (подсчеты) характеристик функционирования устройства**: число принятых пакетов, число отброшенных пакетов, степень заполнения буфера, состояние порта (рабочее или нерабочее);
- **Изменять по командам, полученным извне, конфигурационные параметры.**

Распределение



Подходы к управлению сетью

CLI (Command Line Interface)

- ооператор, используя команды CLI вручную или через скрипт напрямую работает с отдельными устройствами (telnet, ssh)

SNMP/MIB

- оператор делает запросы к данным на устройстве через базу MIB , используя SNMP

NETCONF/YANG

- более абстрактный, общий подход
- управление множеством устройств
- Используется специальный язык YANG
- NETCONF: передает описанные на YANG действия и данные среди устройств сети

SNMP [RFC 1157]

- Простой протокол сетевого администрирования - используется в качестве стандартного протокола взаимодействия менеджера и агента.
- Относится к прикладному уровню стека TCP/IP.
- Для транспортировки своих сообщений он использует дейтаграммный транспортный протокол UDP, который, как известно, не обеспечивает надежную доставку.
- Протокол TCP, организующий надежную передачу сообщений на основе соединений, весьма загружает управляемые устройства и на этапе разработки было решено от него отказаться

Команды SNMP

- Команда **GetRequest** используется менеджером для запроса агента о значении какой-либо переменной по ее стандартному имени.
- Команда **GetNextRequest** применяется менеджером для извлечения значения следующего объекта (без указания его имени) при последовательном просмотре таблицы объектов
- С помощью команды **Response** SNMP-агент передает менеджеру ответ на команду GetRequest или GetNextRequest.

Команды SNMP

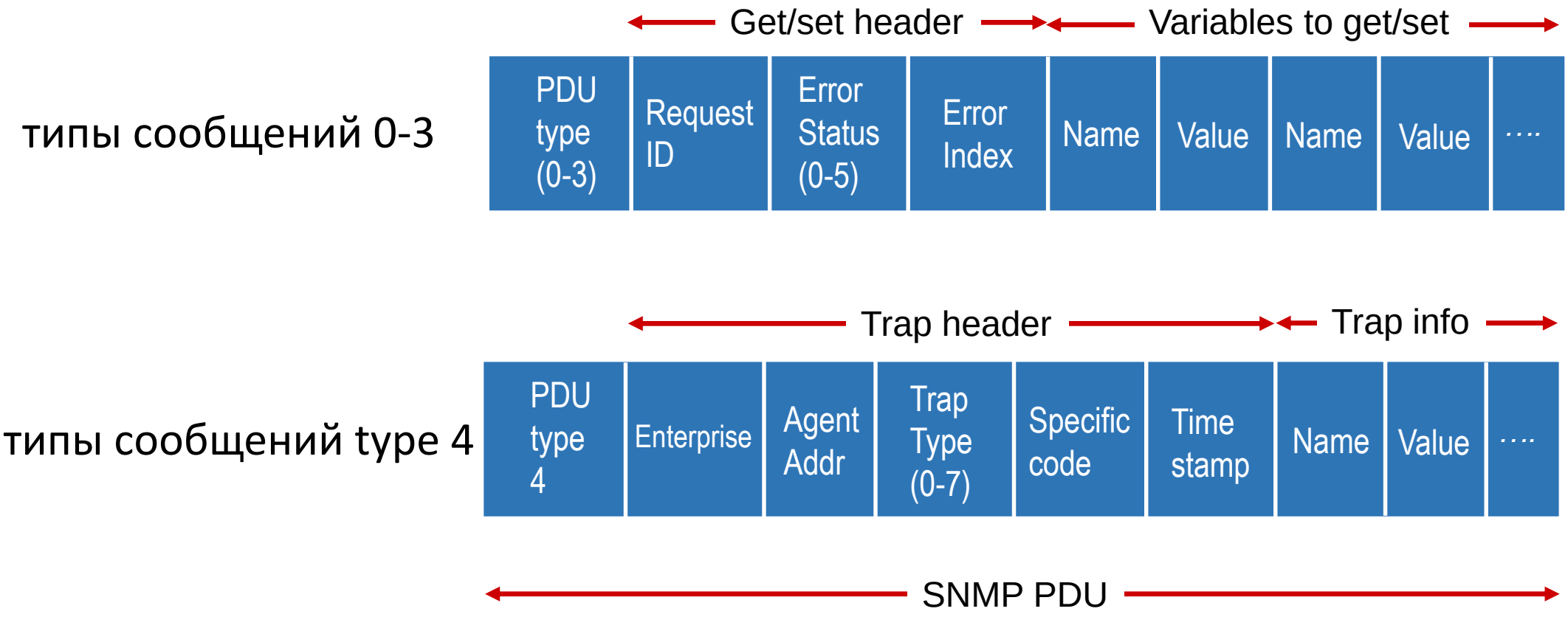
- Команда **SetRequest** позволяет менеджеру изменять значения какой-либо переменной или списка переменных. С помощью этой команды и происходит управление устройством.
- Агент должен «понимать» смысл значений переменной, которая используется для управления устройством, и на основании этих значений выполнять реальное управляющее воздействие — отключить порт, приписать порт определенной линии VLAN и т. п.
- Команда **SetRequest** пригодна также для задания условия, при выполнении которого SNMP-агент должен послать менеджеру соответствующее сообщение.
- Может быть определена реакция на такие события, как
  - инициализация агента
  - рестарт агента
  - обрыв связи
  - восстановление связи
  - неверная аутентификация и потеря ближайшего маршрутизатора

Команды SNMP

- Команда **Trap** используется агентом для сообщения менеджеру о возникновении особой ситуации.
- Команда **GetBulk** позволяет менеджеру получить несколько переменных за один запрос.

Формат пакета SNMP

Пакеты не имеют заголовков с фиксированными полями. Любое SNMP-сообщение состоит из трех основных частей: версии протокола, общей строки и области данных.



Общая строка/ Community string

- используется для группирования устройств, управляемых определенным менеджером.
- является своего рода паролем, так как для того, чтобы устройства могли взаимодействовать по протоколу SNMP, они должны иметь одно и то же значение этого идентификатора (по умолчанию часто употребляется строка «public»).
- Однако этот механизм служит скорее для «распознавания» партнеров, нежели для безопасности.

Область данных

- Содержит описанные команды протокола, а также имена объектов и их значения.
- Состоит из одного или более блоков, каждый из которых может относиться к одному из перечисленных типов команд протокола SNMP
- Для каждого типа команды определен свой формат.

База данных MIB

- База данных MIB содержит значения множества различных типов переменных, характеризующих конкретный управляемый объект.
- В самой первой версии стандарта (MIB-I) для характеристики устройства предлагалось использовать 114 типов переменных.
- Эти переменные организованы в виде дерева. Из корня выходит 8 ветвей, соответствующих восьми группам переменных

- **System** — общие данные об устройстве (например, идентификатор поставщика, время последней инициализации системы);
- **Interfaces** — параметры сетевых интерфейсов устройства (например, их количество, типы, скорости обмена, максимальный размер пакета);
- **Address Translation Table** — описание соответствия между сетевыми и физическими адресами (например, по протоколу ARP);
- **Internet Protocol** — данные, относящиеся к протоколу IP (адреса IP-шлюзов, хостов, статистика об IP-пакетах);
- **ICMP** — данные, относящиеся к протоколу ICMP;
- **TCP** — данные, относящиеся к протоколу TCP (число переданных, принятых и ошибочных TCP-сообщений);
- **UDP** — данные, относящиеся к протоколу UDP (число переданных, принятых и ошибочных UPD-дейтаграмм);
- **EGP** — данные, относящиеся к протоколу EGP (число принятых с ошибками и без ошибок сообщений).

RMON

- Протокол дистанционного мониторинга сети, расширение SNMP.
- Системы управления, построенные на основе RMON, имеют такую же архитектуру, элементами которой являются менеджеры, агенты и управляемые объекты.
- Отличие состоит в том, что SNMP-системы собирают информацию только о событиях, происходящих на тех объектах, на которых установлены агенты, а RMON-системы — также о сетевом трафике.
- С помощью RMON-агента можно провести достаточно детальный анализ работы сетевого сегмента.
- Собрав информацию о наиболее часто встречающихся типах ошибок в кадрах, а затем получив зависимость интенсивности этих ошибок от времени, можно сделать некоторые предварительные выводы об источнике ошибочных кадров и на этом основании сформулировать более тонкие условия захвата кадров со специфическими признаками, соответствующими выдвинутой версии.

NETCONF [RFC 4741]/ [RFC 6241]

- Протокол, ориентированный на соединение
  - SSH, TLS как транспорт
- Клиент Netconf (“manager”) устанавливает сессию с сервером (“agent”)
- Данные кодируются в виде **XML**
- Базируется на **RPC**
  - <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id=”100">
- Определен в RFC4741 (NETCONF 1.0) и RFC6241 (NETCONF 1.1)
- Функция Call-home в процессе стандартизации
  - Возможность инициировать соединение со стороны устройства

Пример сообщения NETCONF RPC

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <rpc message-id="101"
03   xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
04   <edit-config>
05     <target>
06       <running/>
07     </target>
08     <config>
09       <top xmlns="http://example.com/schema/
         1.2/config">
10         <interface>
11           <name>Ethernet0/0</name>
12           <mtu>1500</mtu>
13         </interface>
14       </top>
15     </config>
16   </edit-config>
17 </rpc>
```

изменить конфигурацию

изменить текущую конфигурацию

Изменить MTU на интерф. Ethernet 0/0 на значение 1500

YANG

- YANG использует XML для кодирования данных
  - Определяет правила генерации XML
  - Использует некоторые расширенные возможности XML (например, Xpath) § Хорошо увязывается с NETCONF
- YANG не является XML
  - Акцент на читаемости документа: структура, удобная для программистов. Похожа на C/C++ или Java
  - Грамматика XM определена в YIN (Yang-Independent Notation)
    - Похожая семантика
    - Трансляция семантики YANG <-> YIN

Определены альтернативные методы кодирования (например, JSON для RESTconf)

Использованные источники

- В. Олифер, Н. Олифер “Компьютерные сети. Принципы, технологии, протоколы”
- Д. Куроуз, К. Росс “Компьютерные сети. Нисходящий подход.”
- https://www.cisco.com/c/dam/assets/global/RU/events/cisco-connect/presentation/kon3/17/17_55-18_55nso_netconf_yang_vpatenko_ru.pdf