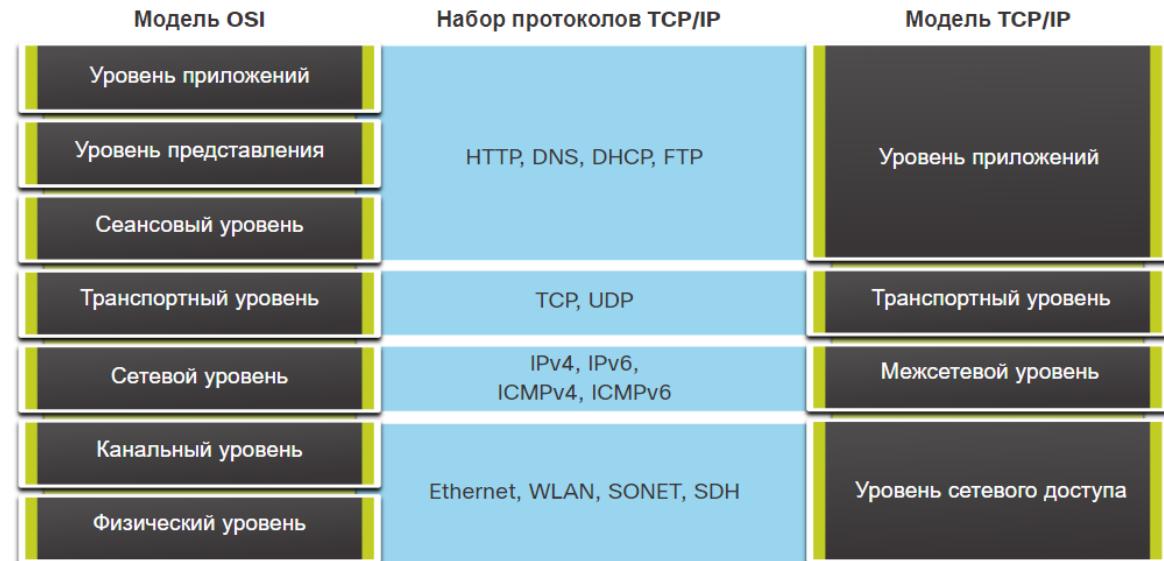


1. Принципы коммутации:	3
a. Канальный уровень: задачи, подуровни, назначение подуровней.	3
b. Кадры канального уровня, инкапсуляция Ethernet.....	5
c. MAC-адрес: определение, типы.....	7
d. Коммутация в сети, работа коммутатора, способы пересылки.....	9
e. Физические характеристики порта: дуплекс, скорость	14
f. Широковещательный домен и домен коллизий.	16
2. Базовая настройка оборудования:	17
a. Среда передачи данных.....	17
b. Сетевое оборудование. Виды, принцип работы, ОС, устройство сетевого оборудования. Порты и интерфейсы, интерфейс loopback.	17
c. Доступ к сетевому оборудованию, управление оборудованием.....	19
d. Командные режимы CLI Cisco IOS.....	22
e. Работа в CLI IOS.....	25
3. Введение в маршрутизацию:	32
a. Сетевой уровень: задачи.	32
b. Маршрутизация на хостах. Таблица маршрутизации хоста.....	35
c. Маршрутизация в общем смысле. Типы маршрутов.	40
d. Статическая и динамическая маршрутизации	42
e. Механизмы пересылки пакетов.	46
4. Маршрутизация:	53
a. Статическая маршрутизация. Типы статических маршрутов.....	53
b. Динамическая маршрутизация. Классификация протоколов маршрутизации.....	58
c. Задачи протоколов маршрутизации. Компоненты.	59
d. Алгоритм работы протокола маршрутизации.	60
e. RIP: определение, описание, версии, принцип работы.....	61
f. Записи таблицы маршрутизации. Вид и устройство таблицы маршрутизации.	66
g. Процесс поиска маршрута по таблице маршрутизации.	69
5. IP-адресация:	70
a. Определение IP-протокола. Его задачи.	70
b. Версии IP. Формат адресов.....	76
c. Типы рассылки IPv4 и IPv6.	79
d. Области действия IPv4 и IPv6.....	81
f. Способы назначения адресов IPv4 и IPv6.....	85
6. DHCP:	86
a. Определение. Алгоритм работы DHCP. Ретрансляция DHCP.....	86
b. Методы назначения IPv6. Алгоритмы работы.....	96
c. Использование сообщений ICMPv6 для работы IPv6.	112
7. Управление оборудованием:	115

a. Протоколы обнаружения устройств: определение, принцип работы, характеристики.	115
b. Службы времени: способы настройки системных часов, протокол NTP, его характеристика и принцип работы.	124
c. Системный журнал: протокол Syslog, характеристика и принцип работы, формат сообщений Syslog, уровни важности.....	129
8. Обслуживание сетевого оборудования:	136
a. Операционные и файловые системы сетевого оборудования.	136
b. Загрузка коммутатора и маршрутизатора. Начальный загрузчик.	140
c. Резервное копирование и восстановление конфигурации.	144
d. Управление образами IOS. Имена файлов образов. Резервное копирование и восстановление образов IOS.....	152
9. Проектирование сети:.....	157
a. Конвергентные сети.....	157
b. Модели построения сетей без границ. Описание, характеристики. Задачи уровней.	158
c. Выбор оборудования при построении сети.	165
10. VLAN:	173
a. Определение, типы VLAN.	173
b. Добавление тега. Trunk-порты.	178
c. Диапазоны VLAN: стандартные и расширенные.....	183
d. Управление портами в VLAN: добавление, удаление, изменение принадлежности.	183
e. Маршрутизация между VLAN: способы, описание.....	194
f. Многоуровневый коммутатор. Маршрутизация на многоуровневом коммутаторе.	204
g. Поиск и устранение неисправностей при работе с VLAN.....	210



Уровни модели TCP/IP	Описание
4 – Прикладной уровень	Представляет данные пользователю, а также обеспечивает кодирование и управление диалоговыми окнами.
3 – Транспортный уровень	Поддерживает связь между различными устройствами в разных сетях.
2 – Межсетевой уровень	Определяет наилучший путь через сеть.
1 – Уровень сетевого доступа	Управляет устройствами и средствами подключения, формирующими сеть.

Физический уровень



Физический уровень OSI обеспечивает средства транспортировки битов, образующих кадр данных канального уровня, по средствам сетевого подключения. Этот уровень принимает от канального уровня целый кадр данных и кодирует его в виде последовательности сигналов, которые затем пересыпаются по средству подключения локальной сети. Закодированные биты, из которых состоит кадр, принимаются либо оконечным, либо промежуточным устройством.

Темы вопросов к экзамену по Сетевым технологиям

1. Принципы коммутации:

а. Канальный уровень: задачи, подуровни, назначение подуровней.

Задача – связать между собой 2 сетевые карты конечных устройств либо сетевого оборудования и передать данные полученные от сетевого уровня (данные IPv4 или IPv6) на физический уровень (витая пара, беспроводная среда). Передача данных осуществляется с помощью кадров.

Канальный уровень модели OSI (Уровень 2) подготавливает сетевые данные для физической сети. Уровень канала передачи данных отвечает за связь между сетевыми интерфейсными картами (NIC). Канальный уровень выполняет следующие функции:

- Обеспечение доступа вышестоящих уровней к среде подключения. Протокол верхнего уровня полностью не знает тип среды, которая используется для пересылки данных.
- Принимает данные, обычно пакеты уровня 3 (например, IPv4 или IPv6), и инкапсулирует их в кадры уровня 2.
- Управление передачей и приемом данных в среде передачи данных.
- Обмен кадрами между узлами по физическим среде сетевого подключения.

- Получает инкапсулированные данные, обычно пакеты уровня 3, и направляет их на соответствующий протокол верхнего уровня.
- Обнаружение ошибок и отклонение любого поврежденного кадра.

Подуровни канала передачи данных IEEE 802 LAN/MAN

Стандарт IEEE 802 LAN/MAN специфичны для сетей Ethernet, беспроводных локальных сетей (WLAN), беспроводных персональных сетей (WPAN) и других типов локальных и городских сетей. Уровень канала данных IEEE 802 LAN/MAN состоит из следующих двух подуровней:

- **Управление логическим соединением (Logical Link Control, LLC)** - Этот подуровень IEEE 802.2 взаимодействует между сетевым программным обеспечением на верхних слоях и аппаратным обеспечением устройства на нижних слоях. Он помещает в кадр информацию, указывающую, какой протокол сетевого уровня используется для данного кадра. Данная информация позволяет различным протоколам 3-го уровня, таким как IPv4 и IPv6, использовать один и тот же сетевой интерфейс и одно и то же средство подключения.
- **Управление доступом к среде (Media Access Control, MAC)** — этот подуровень (IEEE 802.3, 802.11 или 802.15) определяет процессы доступа к среде, выполняемые оборудованием. Он отвечает за инкапсуляцию данных и управление доступом к среде передачи данных. Он обеспечивает адресацию уровня передачи данных и интегрирован с различными технологиями физического уровня.

Сетевой уровень	Протоколы сетевого уровня			
Канальный уровень	Подуровень LLC	LLC «Подуровень» - IEEE 802.2		
	Подуровень MAC	Ethernet IEEE 802.3	WLAN IEEE 802.11	WPAN IEEE 802.15
Физический уровень		Различные стандарты Ethernet для Fast Ethernet, Gigabit Ethernet и т.д.	Различные стандарты WLAN для различных типов беспроводной связи	Различные стандарты WPAN для Bluetooth, RFID и т.д.

Подуровень LLC использует данные сетевых протоколов, которые обычно представлены в виде IPv4 или IPv6 пакета, и добавляет управляющую информацию 2 уровня для доставки пакета к узлу назначения.

Подуровень MAC управляет сетевым адаптером NIC и другим оборудованием, отвечающим за отправку и получение данных на проводном или беспроводном носителе LAN/MAN.

Подуровень MAC обеспечивает инкапсуляцию данных:

- **Разделение кадра:** процесс формирования кадров предоставляет важные разделители для идентификации полей в кадре. Эти разграничитывающие биты обеспечивают синхронизацию между передающими и получающими узлами.
- **Адресация:** обеспечивает адресацию источника и назначения для переноса кадра уровня 2 между устройствами в одной и той же общей среде.
- **Обнаружение ошибок:** каждый кадр содержит концевик, позволяющий выявлять ошибки передачи.

б. Кадры канального уровня, инкапсуляция Ethernet.



Информация, добавленная к кадру, определяется используемым протоколом.

Канальный уровень подготавливает инкапсулированные данные (обычно пакет IPv4 или IPv6) для перемещения по среде передачи данных локальной сети, добавляя к нему заголовок и концевик с целью создать кадр.

Протокол передачи данных отвечает за связь между одной NIC и другой NIC в одной сети. Хотя кадры канального уровня описываются множеством различных протоколов канального уровня, кадры любого типа состоят из трех основных компонентов.

- Заголовок
- Данные
- Концевик

В отличие от других протоколов инкапсуляции, канальный уровень добавляет информацию в виде концевика в конце кадра.

Общие поля кадра показаны на рисунке. Не каждый протокол включает в себя все эти поля. Фактический формат кадра определяется стандартами для конкретного канального протокола.

Поля кадра включают следующее:

- **Флаги начала и конца кадра:** используются для определения границ начала и конца кадра.
- **Адресация:** указывает узлы источника и назначения в среде передачи данных.
- **Тип:** Указывает протокол уровня 3 в поле данных.
- **Управление:** указывает особые службы управления потоком, например качество обслуживания (QoS). Служба QoS используется для приоритетной пересылки определенных типов сообщений. Например, кадры протокола VoIP, обычно пользуются приоритетом, поскольку они чувствительны к задержкам.
- **Данные:** Содержит полезные данные кадра (т.е. заголовок пакета, заголовок сегмента и данные).
- **Обнаружение ошибок:** Идет после данных, чтобы сформировать концевик.



Ethernet определяется протоколами канального и физического уровня.

Подуровень MAC отвечает за инкапсуляцию данных и доступ к среде передачи данных.

Инкапсуляция данных

Инкапсуляция данных IEEE 802.3 включает следующее:

- **Кадр Ethernet** - это внутренняя структура кадра Ethernet.
- **Адресация Ethernet** - Кадр Ethernet включает MAC-адрес источника и назначения для доставки кадра Ethernet из Ethernet NIC в Ethernet в одной локальной сети.
- **Обнаружение ошибок Ethernet**. Кадр Ethernet включает трейлер последовательности проверок кадров (FCS), используемый для обнаружения ошибок.

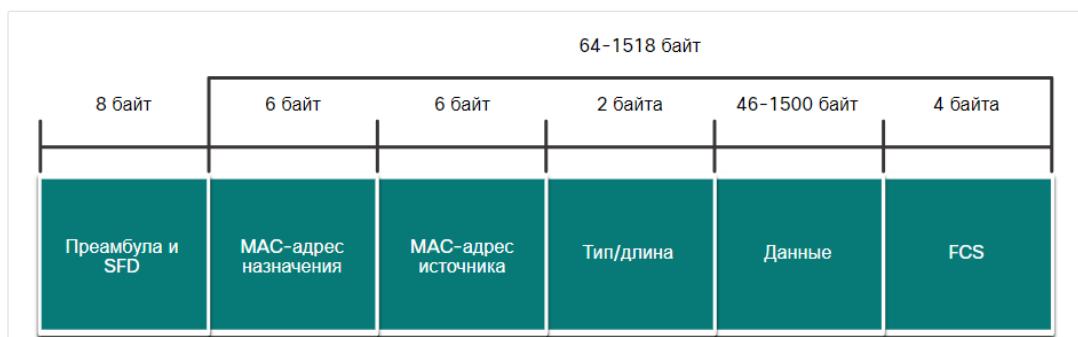
Поля кадра Ethernet

Минимальный размер кадра Ethernet — 64 байта, максимальный — 1518 байт. К этому количеству относятся все байты, начиная с поля «MAC-адрес назначения» и заканчивая полем «Проверочная последовательность кадра (FCS)». Поле «Преамбула» при описании размера кадра не включено.

Любой кадр с длиной менее 64 байтов считается «фрагментом коллизии» или «карликовым кадром» и автоматически отклоняется принимающими станциями. Кадры с длиной более 1500 байт называются Jumbo-кадрами (значительно превышающие допустимый размер) или Baby Giant (слегка превышающие допустимый размер).

Если размер передаваемого кадра меньше минимального значения или больше максимального значения, получающее устройство сбрасывает такой кадр. Отброшенные кадры, скорее всего, являются результатом коллизий или других нежелательных сигналов и, следовательно, считаются недействительными. Кадры Jumbo обычно поддерживаются большинством коммутаторов Fast Ethernet и Gigabit Ethernet и сетевых адаптеров.

Ethernet Frame Fields



Поле	Описание
Поля «Преамбула» и «Начальный разделитель кадра»	Преамбула (7 байт) и Начальный разделитель кадров (SFD), также называемое Начало кадра (1 байт), поля используются для синхронизации между устройствами отправки и приема. Эти первые восемь байтов кадра используются, чтобы привлечь внимание принимающих узлов. По сути, первые несколько байт сообщают получателям о необходимости подготовиться к поступлению нового кадра.
Поле «MAC-адрес назначения»	Это 6-байтное поле является идентификатором для предполагаемого получателя. Как вы помните, этот адрес используется на уровне 2, чтобы помочь устройствам определить, адресован ли им кадр. Адрес в кадре сравнивается с MAC-адресом в устройстве. Если есть совпадение, то устройство принимает кадр. Адрес может быть предназначен для одноадресной, многоадресной и широковещательной рассылок.
Поле «MAC-адреса источника»	Это 6-байтное поле определяет сетевую интерфейсную плату или интерфейс, отправившие кадр.
Тип/длина	Это 2-байтное поле определяет протокол верхнего уровня, инкапсулированный в кадр Ethernet. Характерные значения – значения в шестнадцатеричном формате 0x800 для IPv4, 0x86DD для IPv6 и 0x806 для ARP. Примечание. Вы также можете увидеть это поле, называемое Тип EtherType, тип или длина.
Поле «Данные»	Это поле (46 – 1500 байт) содержит инкапсулированные данные из более высокого уровня, который представляет собой общий PDU уровня 3 или, что более часто, из пакета IPv4. Длина всех кадров должна быть не менее 64 байт. Если небольшой пакет инкапсулирован, дополнительные биты, называемые пэдом, используются для увеличения размера кадра до этого минимального размера.
Поле FCS (Проверочная последовательность кадра)	Поле FCS (Проверочная последовательность кадра) (4 байта) используется для обнаружения ошибок в кадре. В нем используется циклический избыточный код (CRC). Отправляющее устройство включает само результаты CRC в поле FCS кадра. Принимающее устройство получает кадр и генерирует CRC для поиска ошибок. Если расчеты совпадают, ошибки отсутствуют. Расчеты, которые не совпадают, указывают на то, что данные изменились; следовательно, кадр опущен. Изменение данных может быть результатом нарушения электрических сигналов, которые представляют биты.

с. МАС-адрес: определение, типы.

MAC-адрес Ethernet состоит из 48-битного двоичного значения. Шестнадцатеричный используется для идентификации адреса Ethernet, так как одна шестнадцатеричная цифра представляет четыре бита. Таким образом 48-разрядный MAC-адрес Ethernet может быть выражен только с помощью 12 шестнадцатеричных значений.

Каждое устройство в сети Ethernet подключено к одной и той же общей среде передачи данных. MAC-адрес используется для определения источника и места назначения в локальной сети Ethernet. MAC-адресация предоставляет метод идентификации устройств на более низком уровне модели OSI.

MAC-адрес Ethernet — это 48-битный адрес, выраженный с использованием 12 шестнадцатеричных цифр, как показано на рисунке. Поскольку байт равен 8 битам, мы также можем сказать, что MAC-адрес имеет длину 6 байтов.

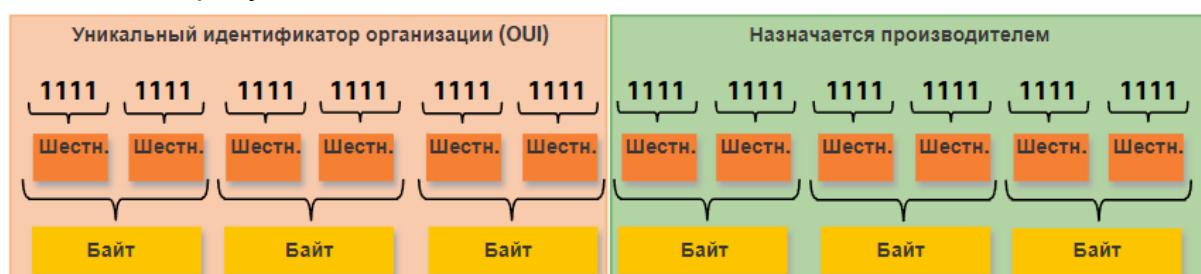


Все MAC-адреса должны быть уникальными для устройства Ethernet или интерфейса Ethernet. Для этого все поставщики, продающие устройства Ethernet, должны зарегистрироваться в IEEE, чтобы получить уникальный 6 шестнадцатеричный (т.е. 24-битный или 3-байтовый) код, называемый организационно уникальным идентификатором (OUI).

Когда поставщик назначает MAC-адрес устройству или интерфейсу Ethernet, поставщик должен выполнить следующие действия:

- Используют свой назначенный OUI в качестве первых 6 шестнадцатеричных цифр.
- Назначая уникальное значение в последних 6 шестнадцатеричных цифрах.

Таким образом MAC-адрес Ethernet состоит из 6 шестнадцатеричного кода OUI поставщика, за которым следует 6 шестнадцатеричных значений, назначенных поставщиком, как показано на рисунке.



При запуске компьютера сетевая плата сначала копирует MAC-адрес из ПЗУ в ОЗУ. Когда устройство пересыпает сообщение в сеть Ethernet, оно добавляет к кадру информацию заголовка.

- **МАС-адрес источника** - Это MAC-адрес сетевой платы устройства источника.
- **МАС-адрес назначения** - Это MAC-адрес сетевой карты устройства назначения.

При поступлении кадра Ethernet на сетевую плату она проверяет MAC-адрес назначения, чтобы определить, совпадает ли он с физическим MAC-адресом устройства, сохраненным в ОЗУ. Если не удается обнаружить совпадения, устройство отклоняет кадр. При наличии совпадения сетевая плата передает кадр вверх по уровням модели OSI, где происходит процесс деинкапсуляции.

Примечание: Сетевые платы устройств Ethernet принимают кадры также в том случае, если MAC-адрес назначения является широковещательной рассылкой или группой многоадресной рассылки, в которую включен узел.

Индивидуальный (одноадресный) MAC-адрес

В сети Ethernet для одноадресной, многоадресной и широковещательной рассылки уровня 2 используются разные MAC-адреса.

Индивидуальный MAC-адрес — это уникальный адрес, который используется при отправке кадра от одного передающего устройства к одному устройству назначения.

Для определения MAC-адреса назначения на узле источника используется протокол разрешения адресов (ARP). Процесс, который использует хост-источник для определения MAC-адреса назначения, связанного с адресом IPv6, называется Neighbor Discovery (ND).

Примечание: MAC-адрес источника всегда должен быть адресом одноадресной рассылки (индивидуальным).

MAC-адрес широковещательной рассылки

Кадр широковещательной передачи Ethernet принимается и обрабатывается каждым устройством в локальной сети Ethernet. Функции широковещательной сети Ethernet заключаются в следующем:

- MAC-адрес назначения — это адрес FF-FF-FF-FF-FF-FF в шестнадцатеричном формате (48 разрядов в двоичном формате).
- Он пересыпается через все порты коммутатора Ethernet, кроме входящего порта.
- Он не пересыпается маршрутизатором.

Если инкапсулированные данные являются широковещательным пакетом IPv4, это означает, что пакет содержит целевой IPv4-адрес, который имеет все единицы (1) в хост-части. Эта нумерация в адресе означает, что все узлы в локальной сети (домене широковещательной рассылки) получат и обработают пакет.

MAC-адрес многоадресной рассылки(групповой)

Кадр многоадресной передачи Ethernet принимается и обрабатывается группой устройств в локальной сети Ethernet, принадлежащих к той же группе многоадресной рассылки. Функции многоадресной рассылки Ethernet заключаются в следующем:

- Существуют другие зарезервированные MAC-адреса назначения многоадресной рассылки для тех случаев, когда инкапсулированные данные не являются IP-адресами, например протокол STP и протокол обнаружения уровня канала (LLDP).
- Он рассыпается на все порты коммутатора Ethernet, за исключением входящего порта, если коммутатор не настроен для многоадресного отслеживания.
- Он не пересыпается маршрутизатором, если маршрутизатор не настроен на маршрутизацию многоадресных пакетов.

Если инкапсулированные данные являются многоадресным IP-пакетом, устройствам, которые принадлежат многоадресной группе, назначается IP-адрес многоадресной группы. Диапазон IPv4-адресов многоадресной рассылки — от 224.0.0.0 до 239.255.255.255.

Диапазон IPv6-адресов многоадресной рассылки начинается с FF00:: /8. Поскольку адреса многоадресной рассылки представляют собой группу адресов (которая иногда называется также группой узлов), они используются только как адреса назначения пакета. Источник всегда имеет адрес одноадресной рассылки.

Как и в случае с адресами для одноадресной и широковещательной рассылки, IP-адресу для многоадресной рассылки требуется соответствующий MAC-адрес, чтобы фактически передавать кадры по локальной сети. MAC-адрес многоадресной рассылки связан и использует информацию адресации от адреса многоадресной рассылки IPv4 или IPv6.

Принадлежность группе определяется работой протокола, которые используются!!!!

d. Коммутация в сети, работа коммутатора, способы пересылки.

Коммутатор Ethernet уровня 2 использует MAC-адреса для принятия решения о пересылке. Устройство не имеет информации о протоколе, передаваемом в части кадра, выделенной для данных, например, в IPv4-пакете или ND-пакет IPv6. Коммутатор пересылает пакеты только на основе MAC-адресов Ethernet уровня 2.

В отличие от устаревших концентраторов Ethernet, которые повторяют биты на всех портах, кроме входящего, коммутатор Ethernet обращается к таблице MAC-адресов для пересылки каждого конкретного кадра. На рисунке показан только что включенный 4-портовый коммутатор. Таблица показывает таблицу MAC-адресов, которая еще не изучила MAC-адреса для четырех подключенных компьютеров.

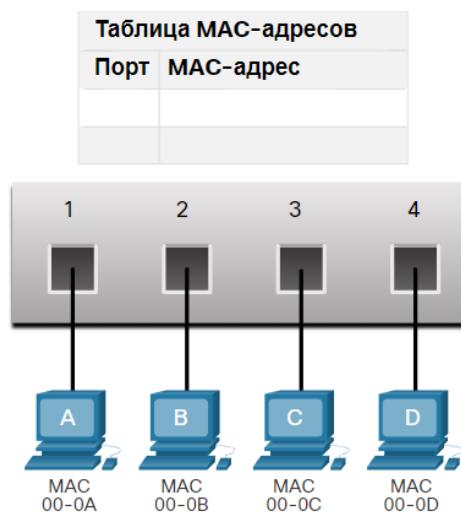


Таблица MAC-адресов коммутатора пуста.

Коммутатор создает таблицу MAC-адресов динамически, проверяя MAC-адрес источника в кадрах, принимаемых портом. Он пересыпает кадры на основе совпадения между MAC-адресом назначения в кадре и записью в таблице MAC-адресов.

Получение информации

При каждом поступлении кадра в коммутатор выполняется проверка на наличие новой информации. Проверяются MAC-адрес источникa, указанный в кадре, и номер порта, по которому кадр поступает в коммутатор. Если MAC-адрес источника отсутствует, он добавляется в таблицу вместе с номером входящего порта. Если MAC-адрес источника уже существует, коммутатор обновляет таймер обновления для этой записи. По умолчанию в большинстве коммутаторов Ethernet данные в таблице хранятся в течение 5 минут.

Например, на рисунке PC-A отправляет кадр Ethernet на PC-D. Таблица показывает, что коммутатор добавляет MAC-адрес для PC-A в таблицу MAC-адресов.

Примечание: Если MAC-адрес источника указан в таблице, но с другим портом, коммутатор считает эту запись новой. Запись заменяется на тот же MAC-адрес, но с более актуальным номером порта.

Таблица MAC-адресов	
Порт	MAC-адрес
1	00-0A



1. PC-A отправляет кадр Ethernet.

2. Коммутатор добавляет номер порта и MAC-адрес PC-A в таблицу MAC-адресов.

Перенаправление

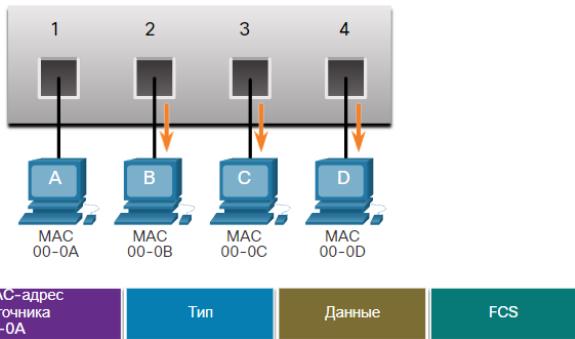
Если MAC-адрес назначения является адресом одноадресной рассылки, коммутатор ищет совпадения между MAC-адресом назначения кадра и записью в таблице MAC-адресов.

Если MAC-адрес назначения есть в таблице, коммутатор пересыпает кадр через указанный порт. Если MAC-адреса назначения нет в таблице, коммутатор пересыпает кадр через все порты, кроме входящего порта. Это называется одноадресной рассылкой неизвестному получателю.

Как показано на рисунке, в таблице коммутатора нет MAC-адреса назначения для компьютера PC-D, поэтому он пересыпает кадр через все порты, кроме порта 1.

Примечание: Если MAC-адрес назначения является адресом широковещательной или многоадресной рассылки, коммутатор также пересыпает кадр через все порты, кроме входящего.

Таблица MAC-адресов	
Порт	MAC-адрес
1	00-0A



1. MAC-адреса назначения нет в таблице

2. перешлет кадр на все порты

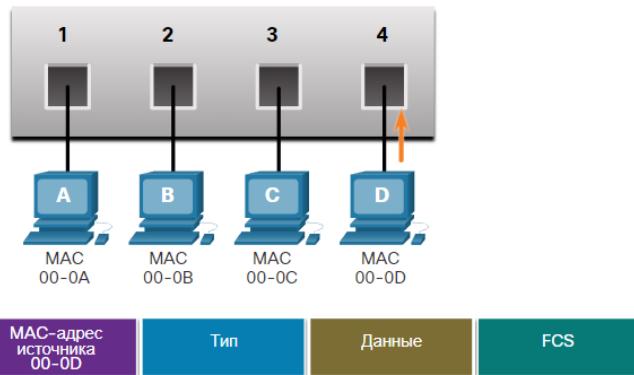
Фильтрация кадров

Поскольку коммутатор получает кадры от разных устройств, его таблица MAC-адресов заполняется через проверку MAC-адреса источника каждого кадра. Если в таблице MAC-адресов коммутатора есть MAC-адрес назначения, он может выполнять фильтрацию кадров и пересыпать его через один порт.

На рисунке PC-D отвечает на PC-A. Коммутатор видит MAC-адрес PC-D во входящем кадре на порту 4. Затем коммутатор помещает MAC-адрес PC-D в таблицу MAC-адресов, связанную с портом 4.

Таблица MAC-адресов

Порт	MAC-адрес
1	00-0A
4	00-0D

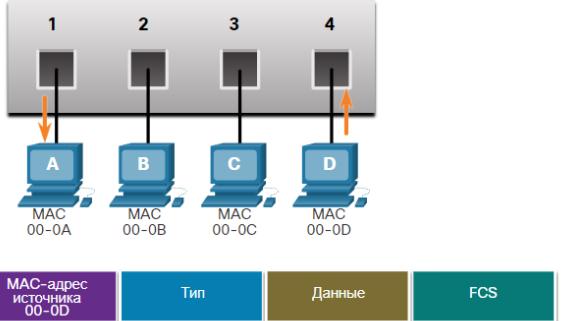


Коммутатор добавляет номер порта и MAC-адрес PC-D в свою таблицу MAC-адресов.

Далее, поскольку коммутатор имеет MAC-адрес назначения для PC-A в таблице MAC-адресов, он отправит кадр только через порт 1, как показано на рисунке.

Таблица MAC-адресов

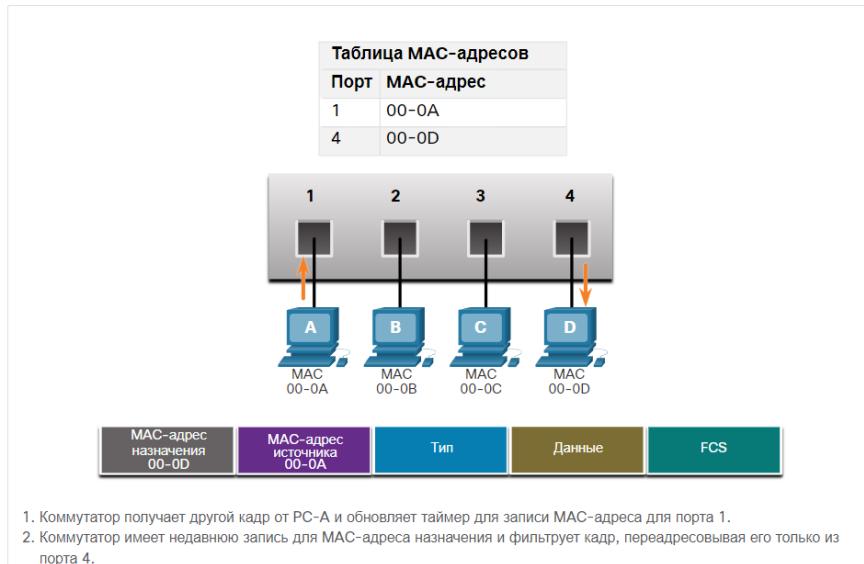
Порт	MAC-адрес
1	00-0A
4	00-0D



1. Коммутатор имеет запись MAC-адреса для указанного получателя.

2. Коммутатор фильтрует кадр, отправляя его только из порта 1.

Затем PC-A отправляет другой кадр PC-D, как показано на рисунке. Таблица MAC-адресов уже содержит MAC-адрес для PC-A; следовательно, пятиминутный таймер обновления для этой записи сбрасывается. Поскольку в таблице коммутатора уже есть MAC-адрес компьютера PC-D, он пересыпает кадр только через порт 4.



1. Коммутатор получает другой кадр от PC-A и обновляет таймер для записи MAC-адреса для порта 1.
2. Коммутатор имеет недавнюю запись для MAC-адреса назначения и фильтрует кадр, переадресовывая его только из порта 4.

В таблице коммутатора может быть несколько MAC-адресов, связанных с одним портом.

Обычно это происходит тогда, когда коммутатор соединен с другим коммутатором. В таблице MAC-адресов коммутатора вводится отдельная запись для каждого кадра, получаемого с другого MAC-адреса источника.

Если IP-адрес устройства находится в удаленной сети, отправить кадр Ethernet в устройство назначения напрямую невозможно. Вместо этого кадр Ethernet отправляется по MAC-адресу шлюза по умолчанию, т. е. маршрутизатора.

Способы переадресации кадра на коммутаторах Cisco

В коммутаторах Cisco существует два метода переадресации кадров, и есть веские причины использовать один вместо другого, в зависимости от ситуации.

Коммутаторы используют один из двух способов пересылки для коммутации данных между сетевыми портами:

- **Коммутация с промежуточным хранением** - В этом методе пересылки кадров коммутатор получает кадр целиком и вычисляет циклический избыточный код (CRC). CRC использует математическую формулу, основанную на количестве бит (единиц) в кадре, что позволяет определить наличие ошибок в полученном кадре. Если значение CRC допустимо, коммутатор ищет адрес назначения, который определяет выходной интерфейс. Затем кадр перенаправляется к правильному порту.
- **Коммутация со сквозной пересылкой** - В этом режиме коммутатор пересыпает данный кадр до его полного получения. Рекомендуется указать адрес назначения кадра в начале, прежде чем кадр может быть переадресован.

Большим преимуществом коммутации с промежуточным хранением является то, что она определяет, есть ли у кадра ошибки перед передачей кадра. Если же в кадре обнаружена ошибка, коммутатор отклонит его. Отклонение кадров с ошибками позволяет уменьшить ширину полосы пропускания, потребляемую поврежденными данными. Коммутация с промежуточным хранением необходима для анализа качества обслуживания (QoS) в конвергентных сетях, в которых требуется классификация кадра для назначения приоритетов проходящего трафика. Например, при передаче речи по IP потоки данных должны иметь больший приоритет, чем трафик, используемый для просмотра веб-страниц. При использовании сквозной коммутации коммутатор обрабатывает данные по мере их поступления даже в том случае, если передача еще не завершена. Коммутатор добавляет в буфер только ту часть кадра, которая требуется для чтения MAC-адреса назначения, чтобы он смог определить, на какой порт пересыпать данные. MAC-адрес назначения указан в

первых 6 байтах кадра после преамбулы. Коммутатор ищет MAC-адрес назначения в своей таблице коммутации, определяет порт исходящего интерфейса и направляет кадр на узел назначения через определенный порт коммутатора. Коммутатор не проверяет кадр на наличие каких-либо ошибок.

Существуют два варианта сквозной коммутации.

- **Коммутация с быстрой пересылкой.** Коммутация с быстрой пересылкой обеспечивает наименьший уровень задержки. При такой коммутации пакет пересыпается сразу же после чтения адреса назначения. Поскольку при коммутации с быстрой пересылкой (fast-forward) передача начинается до того, как будет получен весь пакет, могут быть моменты, когда пакеты ретранслируются с ошибками. Это происходит редко, а сетевой адаптер назначения отклоняет пакет с ошибками после его получения. В режиме быстрой пересылки задержка измеряется с момента получения первого бита до передачи первого бита. Коммутация с быстрой пересылкой является типичным способом сквозной коммутации.
- **Коммутация с исключением фрагментов.** При коммутации с исключением фрагментов коммутатор сохраняет первые 64 байта кадра перед его отправкой. Коммутацию с исключением фрагментов можно рассматривать как компромиссный вариант между коммутацией с промежуточным хранением и коммутацией с быстрой пересылкой. Причина, по которой при коммутации с исключением фрагментов сохраняются только первые 64 байта кадра, заключается в том, что большинство сетевых ошибок и коллизий происходит именно в первых 64 байтах. Коммутация с исключением фрагментов позволяет повысить эффективность коммутации с быстрой пересылкой благодаря выполнению небольшой проверки ошибок в первых 64 байтах кадра, чтобы перед пересылкой кадра убедиться в отсутствии коллизии. Коммутация с исключением фрагментов представляет собой компромисс между большой задержкой с высокой целостностью (коммутация с промежуточным хранением) и малой задержкой с меньшей целостностью (коммутация с быстрой пересылкой).

Некоторые коммутаторы настроены на использование сквозной коммутации для каждого порта до тех пор, пока не будет достигнуто указанное пользователем предельное количество ошибок, после чего автоматически устанавливается коммутация с промежуточным хранением. После того, как частота повторения ошибок снизится до установленного предельного значения, порт автоматически переключится на использование сквозной коммутации.

Буферизация памяти на коммутаторах

Коммутатор Ethernet может использовать метод буферизации для хранения кадров до их пересылки. Буферизация также может использоваться, когда порт назначения занят из-за перегрузки. Коммутатор сохраняет кадр до тех пор, пока он не будет передан.

Метод	Описание
Буферизация памяти на основе портов	<ul style="list-style-type: none">Кадры хранятся в очередях, связанных с определенными входящими и исходящими портами..Кадр передается на исходящий порт только тогда, когда все кадры впереди в очереди были успешно переданы.Для одного кадра возможно задержать передачу всех кадров в памяти из-за занятого порта назначения..Такая задержка возникает и в том случае, если другие кадры можно передать на открытые порты назначения.
Буферизация совместно используемой памяти	<ul style="list-style-type: none">Помещает все кадры в общий буфер памяти, совместно используемый всеми портами коммутатора, и объем буферной памяти, требуемой для порта, распределяется динамически.Кадры в буфере динамически связываются с портом назначения. позволяющий получить пакет на одном порту, а затем передается на другой порт, не перемещая его в другую очередь.

е. Физические характеристики порта: дуплекс, скорость.

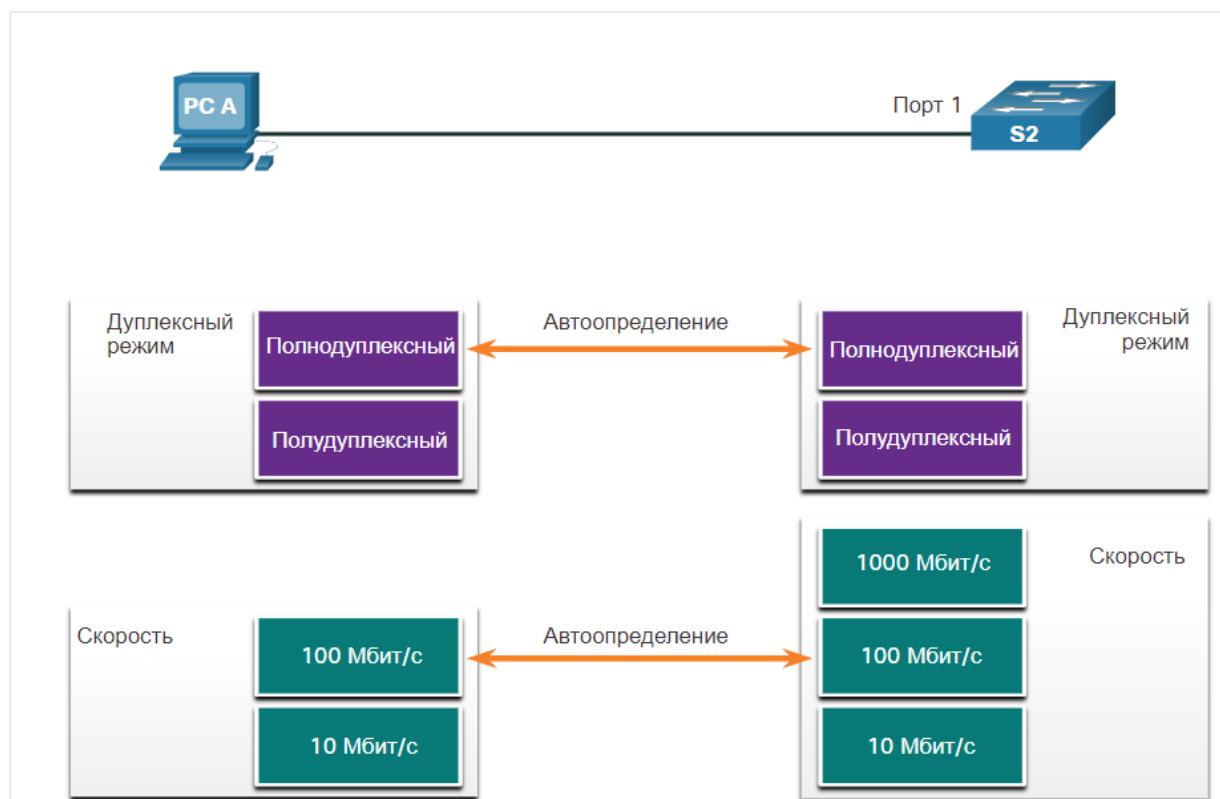
К двум базовым параметрам коммутатора относятся пропускная способность и дуплексный режим, которые задаются для каждого отдельного порта коммутатора. Важно, чтобы настройки дуплексного режима и пропускной способности порта коммутатора и подключенных устройств, таких как компьютер или другой коммутатор, совпадали.

Для обмена данными в сетях Ethernet используются два типа настроек дуплексного режима.

- **Полнодуплексный режим** : одновременная отправка и получение данных в обе стороны.
- **Полудуплексный режим** : отправка данных только одной стороной.

Автоопределение — это дополнительная функция, которой оснащено большинство коммутаторов и сетевых плат Ethernet. Автоопределение позволяет двум устройствам автоматически обмениваться информацией о скорости и возможностях дуплексного режима. Если оба устройства поддерживают полнодуплексный режим, для работы выбирается этот режим вместе с максимальной пропускной способностью, общей для двух устройств.

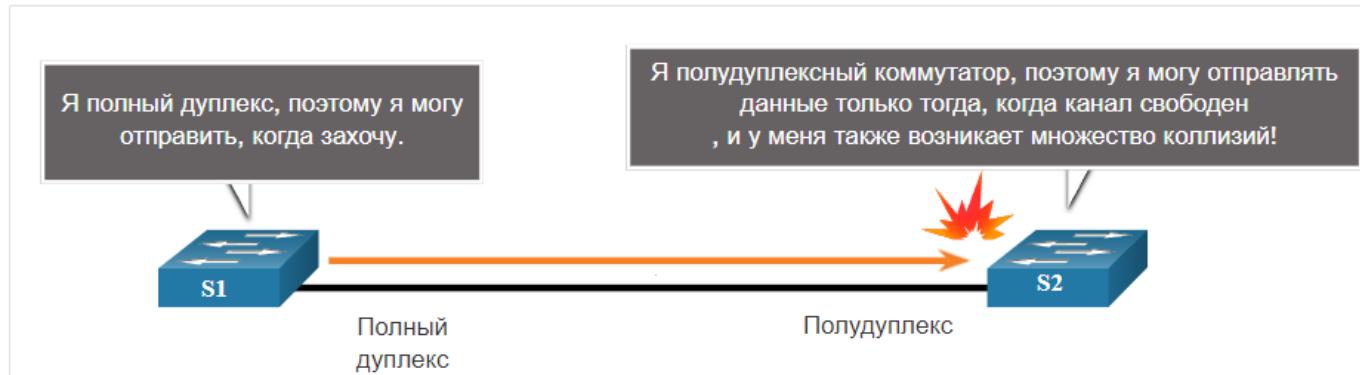
Например, сетевая плата Ethernet компьютера PC-A, показанная на рисунке, может работать в полнодуплексном или полудуплексном режиме на скорости 10 или 100 Мбит/с.



Компьютер PC-A соединен через порт 1 с коммутатором S1, который может работать в полнодуплексном или полудуплексном режиме на скорости 10, 100 или 1 000 Мбит/с (1 Гбит/с). Если в обоих устройствах есть автоопределение, то будет выбраны полнодуплексный режим и скорость 100 Мбит/с.

Примечание: В большинстве коммутаторов и сетевых плат Ethernet компании Cisco используется автоопределение скорости и настроек дуплексного режима. Порты Gigabit Ethernet работают только в полнодуплексном режиме.

Несоответствие дуплексных режимов является наиболее распространенной причиной снижения производительности каналов Ethernet. Это происходит, когда один порт канала работает в полудуплексном режиме, а другой – в полнодуплексном.



В коммутаторе S2 будут по-прежнему возникать коллизии, поскольку коммутатор S1 непрерывно отправляет имеющиеся кадры.

Настройка на физическом уровне

Настройка дуплекса

```
S(config-if)# duplex type  
S(config-if)# duplex full
```

Настройка скорости

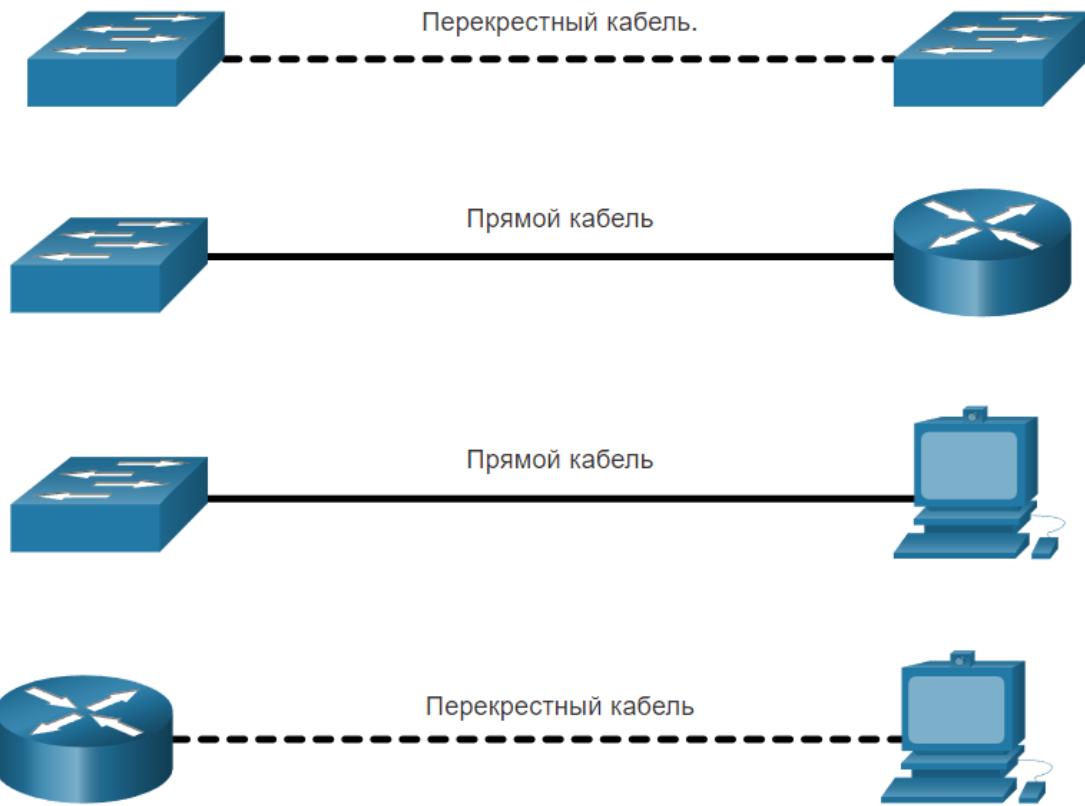
```
S(config-if)# speed value  
S(config-if)# speed 100
```

```
S1# configure terminal  
S1(config)# interface FastEthernet 0/1  
S1(config-if)# duplex full  
S1(config-if)# speed 100  
S1(config-if)# end  
S1# copy running-config startup-config
```

Для соединения устройств когда-то требовалось использование либо перекрестного, либо прямого кабеля. Тип необходимого кабеля зависит от типа соединяемых устройств. Например, на рисунке указан правильный тип кабеля, необходимый для соединения устройств коммутатор-коммутатор, коммутатор-маршрутизатор, коммутатор-хост или

маршрутизатор-хост. Переходный кабель используется при подключении подобных устройств, а прямой кабель используется для подключения неподобных устройств.

Примечание: Прямое соединение между маршрутизатором и узлом требует переходного подключения.



Теперь большинство устройств поддерживают функцию автоматического определения перекрещивания пар на зависящем от среды передачи интерфейсе (Auto-MDIX). Если функция Auto-MDIX включена, коммутатор определяет необходимый тип кабеля, подключенного к порту, и настраивает интерфейс соответствующим образом. Таким образом, для подключения к медным портам 10/100/1000 Мбит/с на коммутаторе можно использовать либо переходный, либо прямой кабель независимо от типа устройства на другом конце соединения.

Функция Auto-MDIX включена по умолчанию на коммутаторах с операционной системой Cisco IOS 12.2 (18) SE или более поздней версии. Однако эта функция может быть отключена. По этой причине всегда следует использовать правильный тип кабеля и не полагаться на функцию Auto-MDIX. Функция Auto-MDIX может быть повторно включена с помощью команды конфигурации интерфейса **mdix auto**.

f. Широковещательный домен и домен коллизий.

Как коммутаторы работают друг с другом и с другими устройствами для устранения коллизий и снижения перегрузки сети.

В старых сегментах Ethernet на основе концентратора сетевые устройства соревновались за общий носитель. Сегменты сети, в которых устройства совместно используют полосу пропускания, называются коллизионными доменами. Если два или более устройств в одном коллизионном домене одновременно пытаются передавать данные, возникает коллизия. Если коммутационный порт Ethernet работает в полудуплексном режиме, каждый сегмент находится в своем собственном коллизионном домене. При работе портов коммутатора в

полнодуплексном режиме не существует доменов столкновений. Тем не менее может существовать домен коллизии, если порт коммутатора работает в полу duplexном режиме. Порт коммутатора ограничивает домен коллизий, то есть за порт коммутатора коллизия не проходит.

Домены широковещательной рассылки

Совокупность соединенных коммутаторов формирует единый широковещательный домен.

Только устройство сетевого уровня, например, маршрутизатор, может разделить широковещательный домен уровня 2. Маршрутизаторы используются для сегментации доменов широковещательной рассылки, но они также сегментируют домен коллизий.

Когда устройство отправляет широковещательную рассылку уровня 2, MAC-адрес назначения в кадре представлен единицами в двоичном формате.

Широковещательный домен уровня 2 называют широковещательным доменом MAC-адресов. В широковещательный домен MAC-адресов входят все устройства локальной сети, которые получают кадры широковещательной рассылки от узла.

Когда коммутатор получает широковещательный кадр, он пересыпает кадр из всех своих портов, за исключением входного порта, на котором широковещательный кадр был получен. Каждое устройство, подключенное к коммутатору, получает копию широковещательного кадра и обрабатывает ее.

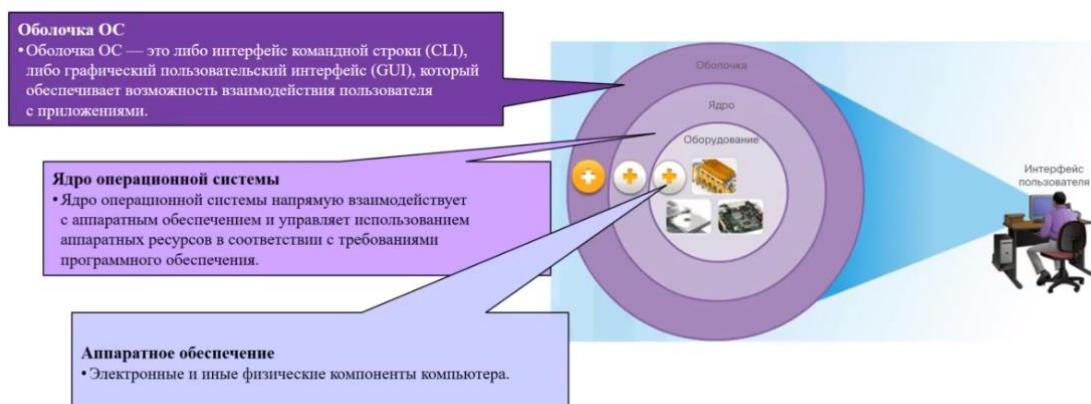
В некоторых случаях широковещательные рассылки необходимы для первоначального местонахождения других устройств и сетевых сервисов, но, кроме этого, они снижают эффективность сети. Полоса пропускания сети используется для распространения широковещательного трафика. Чрезмерное количество широковещательных рассылок и высокая интенсивность трафика в сети могут привести к перегруженности и в результате к снижению производительности сети.

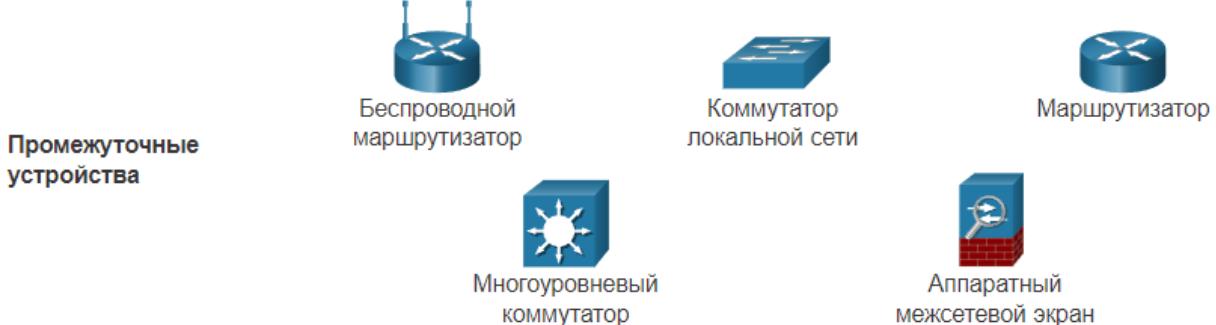
Когда два коммутатора соединены, широковещательный домен увеличивается.

2. Базовая настройка оборудования:

- Среда передачи данных.
- Сетевое оборудование. Виды, принцип работы, ОС, устройство сетевого оборудования. Порты и интерфейсы, интерфейс loopback.

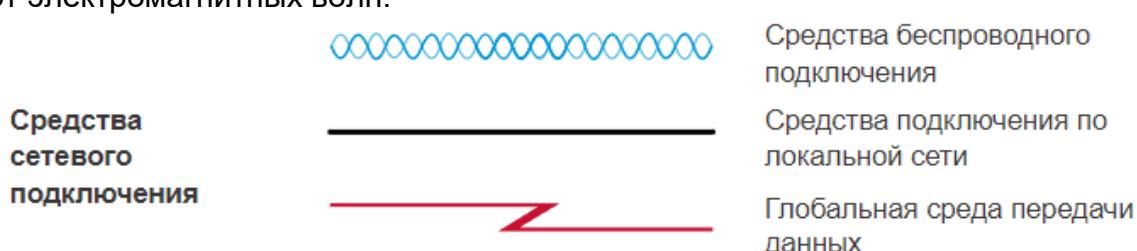
Операционная система





Коммуникация передается по среде передачи данных. Среда передачи данных предоставляет собой канал, по которому сообщение передается от источника к адресату. Современные сети используют в первую очередь три типа носителей для соединения устройств, как показано на рисунке:

- **Металлические провода в кабелях** — данные кодируются в электрические импульсы.
- **Стеклянные или пластиковые волокна (оптоволоконный кабель)** — данные кодируются в световые импульсы.
- **Беспроводная передача** - Данные кодируются посредством модуляции конкретных частот электромагнитных волн.



В дополнение к этим представлениям используется специальная терминология для описания того, как каждое из этих устройств и носители соединяются друг с другом:

- **Сетевые интерфейсные платы (Network Interface Card, NIC)** служат для подключения устройства к сети.
- **Физический порт** — разъем на сетевом устройстве, через который кабели подключены к компьютеру или другому сетевому устройству.
- **Интерфейс** — специализированные порты в сетевом устройстве, которые подключаются к отдельным сетям. Поскольку маршрутизаторы используются для связывания сетей, порты маршрутизатора называются сетевыми интерфейсами.

Примечание: Термины «порт» и «интерфейс» часто взаимозаменяются.

Интерфейс loopback

- Логический, внутренний по отношению к маршрутизатору интерфейс
- Не назначается физическому порту и не может быть подключен к другому устройству
- Считается программным интерфейсом, который автоматически переводится в состояние up (активен) во время работы маршрутизатора

```
Router(config)# interface loopback number
```

```
Router(config-if)# ip address ip-address subnet-mask
```

с. Доступ к сетевому оборудованию, управление оборудованием.

Назначение ОС



```
analyst@secOps ~]$ ls
Desktop Downloads lab.support.files second_drive
[analyst@secOps ~]$
```

При помощи GUI пользователь операционной системы ПК может выполнять следующие задачи.

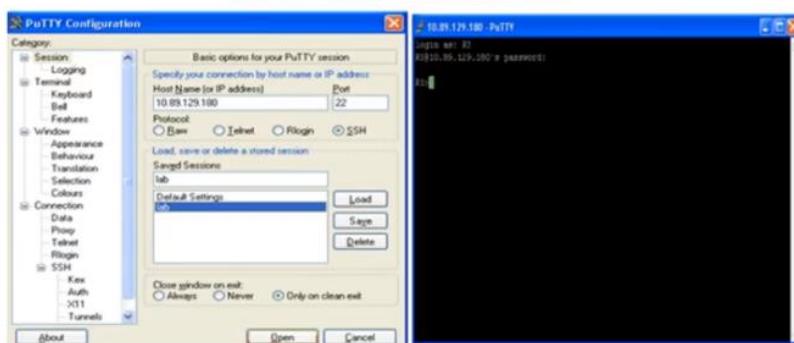
- Выбирать различные объекты и запускать программы, используя мышь
- Вводить текст и текстовые команды
- Просматривать выходные данные на экране монитора

Сетевая операционная система на основе интерфейса командной строки позволяет сетевому специалисту выполнять следующие действия:

- Запускать сетевые программы на базе CLI, используя клавиатуру
- Вводить текст и текстовые команды с клавиатуры
- Просматривать выходные данные на экране монитора

3

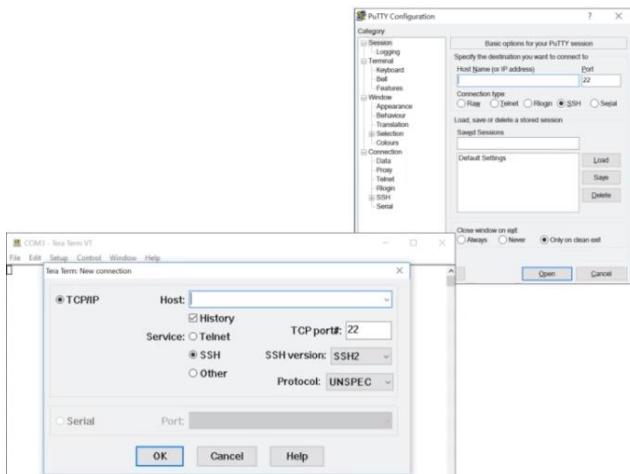
Способы доступа



- **Консоль** — физический порт управления, используемый для доступа к устройству для обслуживания, например для выполнения начальных конфигураций
- **Secure Shell (SSH)** — метод, позволяющий удаленно установить защищенное подключение CLI через виртуальный интерфейс по сети
- **Telnet** — устанавливает небезопасное удаленное подключение CLI к устройству по сети. Данные для аутентификации пользователя, пароли и команды передаются по сети в виде простого текста

4

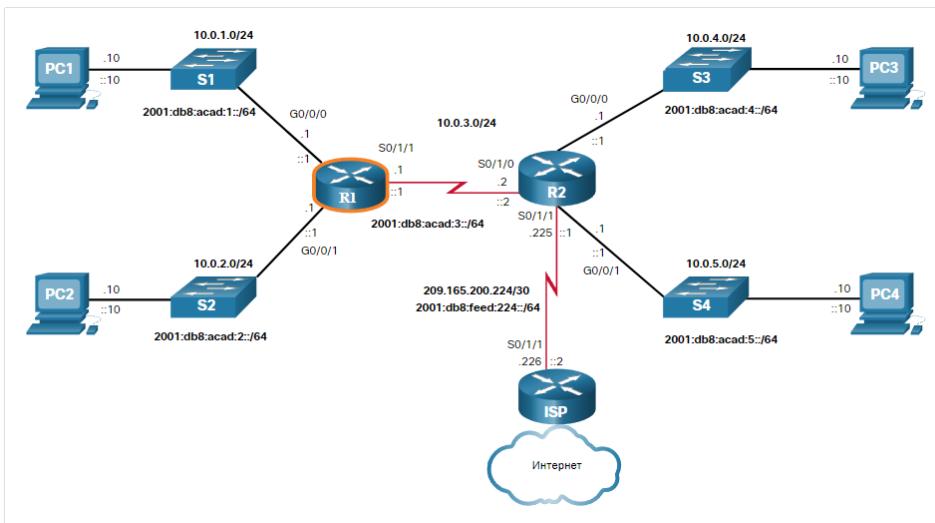
Программа эмуляции терминала



- Программы эмуляции терминала используются для подключения к сетевому устройству с помощью консольного порта или соединения SSH/Telnet
- Есть несколько программ эмуляции терминала, такие как PuTTY, Tera Term и SecureCRT

5

Базовые настройки маршрутизатора



```

Router> enable
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# hostname R1
R1(config)# enable secret class
R1(config)# line console 0
R1(config-line)# logging synchronous
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# exit
R1(config)# line vty 0 4
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# transport input ssh telnet
R1(config-line)# exit
R1(config)# service password-encryption
R1(config)# banner motd #
Enter TEXT message. End with a new line and the #
*****WARNING: Unauthorized access is prohibited!*****
*****#

```

```

R1(config)# ipv6 unicast-routing
R1(config)# interface gigabitethernet 0/0/0
R1(config-if)# description Link to LAN 1
R1(config-if)# ip address 10.0.1.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:acad:1::1/64
R1(config-if)# ipv6 address fe80::1:a link-local
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface gigabitethernet 0/0/1
R1(config-if)# description Link to LAN 2
R1(config-if)# ip address 10.0.2.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:acad:2::1/64
R1(config-if)# ipv6 address fe80::1:b link-local
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface serial 0/1/1
R1(config-if)# description Link to R2
R1(config-if)# ip address 10.0.3.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:acad:3::1/64
R1(config-if)# ipv6 address fe80::1:c link-local
R1(config-if)# no shutdown
R1(config-if)# exit
R1# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
R1#

```

Общие команды проверки включают следующее:

- **show ip interface brief**
- **show running-config interface interface-type number**
- **show interfaces**
- **show ip interface**
- **show ip route**
- **ping**

В каждом случае замените **ip** на версию команды IPv6 **ipv6**.

```

R1# show ip interface brief
Interface IP-Address OK? Method Status Protocol
GigabitEthernet0/0/0 10.0.1.1 YES manual up up
GigabitEthernet0/0/1 10.0.2.1 YES manual up up
Serial0/1/0 unassigned YES unset administratively down down
Serial0/1/1 10.0.3.1 YES manual up up
GigabitEthernet0 unassigned YES unset down down
R1#

```

```
R1# show running-config interface gigabitethernet 0/0/0
Building configuration...
Current configuration : 189 bytes
!
interface GigabitEthernet0/0/0
  description Link to LAN 1
  ip address 10.0.1.1 255.255.255.0
  negotiation auto
  ipv6 address fe80::1:a link-local
  ipv6 address 2001:DB8:ACAD:1::1/64
end
R1#
```

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
< output omitted>
Gateway of last resort is not set
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C 10.0.1.0/24 is directly connected, GigabitEthernet0/0/0
L 10.0.1.1/32 is directly connected, GigabitEthernet0/0/0
C 10.0.2.0/24 is directly connected, GigabitEthernet0/0/1
L 10.0.2.1/32 is directly connected, GigabitEthernet0/0/1
C 10.0.3.0/24 is directly connected, Serial0/1/1
L 10.0.3.1/32 is directly connected, Serial0/1/1
R1#
```

Удобство работы с интерфейсом командной строки также можно повысить с помощью фильтрации выходных данных команды **show**. Для отображения определенных разделов выходных данных можно использовать команды фильтрации. Чтобы включить фильтрацию, введите вертикальную черту | после команды **show**, а затем введите параметр и выражение фильтрации.

К параметрам фильтрации, которые следует указывать после вертикальной черты, относятся:

- **section** — показать целый раздел, который начинается с заданного фильтра.
- **include** — включить все строки выходных данных, которые соответствуют заданному фильтру.
- **exclude** — исключить все строки выходных данных, которые соответствуют заданному фильтру.
- **begin** — показать все строки выходных данных от конкретного места, начиная с линии, которая соответствует заданному фильтру.

Примечание: Фильтры выходных данных можно использовать в сочетании с любой командой **show**.

```

R1# show running-config | section line vty
line vty 0 4
password 7 121A0C0411044C
login
transport input telnet ssh
R1#
R1# show ipv6 interface brief | include up
GigabitEthernet0/0/0 [up/up]
GigabitEthernet0/0/1 [up/up]
Serial0/1/1 [up/up]
R1#
R1# show ip interface brief | exclude unassigned
Interface IP-Address OK? Method Status Protocol
GigabitEthernet0/0/0 192.168.10.1 YES manual up
GigabitEthernet0/0/1 192.168.11.1 YES manual up
Serial0/1/1 209.165.200.225 YES manual up
R1#
R1# show ip route | begin Gateway
Gateway of last resort is not set
    192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.10.0/24 is directly connected, GigabitEthernet0/0/0
L      192.168.10.1/32 is directly connected, GigabitEthernet0/0/0
    192.168.11.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.11.0/24 is directly connected, GigabitEthernet0/0/1
L      192.168.11.1/32 is directly connected, GigabitEthernet0/0/1
    209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
C      209.165.200.224/30 is directly connected, Serial0/1/1
L      209.165.200.225/32 is directly connected, Serial0/1/1
R1#

```

d. Командные режимы CLI Cisco IOS.

По соображениям безопасности Cisco IOS использует два отдельных командных режима для доступа к административным функциям.

- **Пользовательский режим EXEC** — это режим с ограниченными возможностями, но он удобен для базовых операций. В пользовательском режиме доступно только ограниченное число основных команд мониторинга, но невозможно выполнять какие-либо команды, которые могут изменить конфигурацию устройства. Пользовательский режим EXEC можно определить по командной строке CLI, оканчивающейся символом «>».
- **Привилегированный режим EXEC** — этот режим должен использовать сетевой администратор для выполнения команд настройки. Режимы конфигурации более высокого уровня, например, режим глобальной конфигурации, доступны только из привилегированного режима EXEC. Привилегированный режим EXEC можно определить по командной строке, оканчивающейся символом # .

Командный режим	Описание	Командная строка устройства по умолчанию
Пользовательский режим EXEC	<ul style="list-style-type: none"> Обеспечивает доступ к ограниченному количеству базовых команд мониторинга. Этот режим часто называется «режимом только для просмотра». 	Switch> Router>
Привилегированный режим EXEC	<ul style="list-style-type: none"> В этом режиме предоставляется доступ ко всем командам и функциям. Пользователь может использовать любые команды мониторинга, а также имеет доступ ко всем конфигурациям и командам управления. 	Switch# Router#

• Пользовательский режим EXEC (User EXEC)

- hostmane>
- Команды, не меняющие конфигурацию
- Базовые команды мониторинга
- show

Router>
Switch>

• Привилегированный режим EXEC (Privileged EXEC)

- hostname#
- Команды настройки и управления
- enable / disable
- clear

Router#
Switch#

• Режим глобальной конфигурации (Global Configure)

- hostname (config)#+
- configure terminal / exit

Switch(config)#+

Для настройки устройства пользователь должен перейти в режим глобальной конфигурации.

В режиме глобальной конфигурации выполняются изменения конфигурации CLI, влияющие на работу устройства в целом. Режим глобальной конфигурации можно определить по командной строке с именем устройства, после которого следует (config)#+, например Switch(config)#.

Перед тем как перейти в другие специализированные режимы конфигурации, нужно войти в режим глобальной конфигурации. Из режима глобальной конфигурации пользователь может перейти в различные дополнительные режимы конфигурации. Каждый из этих режимов позволяет настроить конфигурацию отдельной части или функции устройства IOS. Два распространенных вложенных режима конфигурации

- **Режим настройки линии** - предназначен для настройки доступа через одну из физических или виртуальных линий (консоль, SSH, Telnet или AUX).
- **Режим настройки интерфейса** - предназначен для настройки порта коммутатора или сетевого интерфейса маршрутизатора.

Когда используется CLI, режим определяется приглашением командной строки, которое является уникальным для этого режима. По умолчанию каждый диалог начинается с имени устройства. После имени следует остальная часть диалога, которая определяет режим. Например, запрос по умолчанию для режима конфигурации — **Switch(config-line)#** а для режима интерфейсной настройки — **Switch(config-if)#**.

- Специальные режимы конфигурации (Configure Specific)

- interface, vlan, router etc.
- hostname (config-if) #

Переключение между режимами IOS

```
Switch> enable  
Switch#
```

```
Switch(config)#line console 0  
Switch(config-line)#exit  
Switch(config) #
```

```
Switch(config)#line console 0  
Switch(config-line)#end  
Switch#
```

Для выхода из режима подконфигурации и возврата в режим глобальной конфигурации используйте команду **exit**. Для возврата в привилегированный режим EXEC используйте команду **end** или комбинацию клавиш **Ctrl + Z**.

Для переключения диалогов командной строки используются различные команды. Чтобы перейти из пользовательского режима EXEC в привилегированный, введите команду **enable**. Чтобы вернуться в пользовательский режим EXEC, используйте команду привилегированного режима **disable**.

Примечание: Привилегированный режим EXEC иногда называют режимом включения (**enable**).

Для входа в режим глобальной конфигурации и выхода из него используйте команду привилегированного режима EXEC **configure terminal**. Чтобы вернуться в привилегированный режим EXEC, введите команду режима глобальной конфигурации **exit**. Есть множество разных вложенных режимов конфигурации. Например, для перехода в режим подконфигурации линии введите команду **line**, а затем тип и номер нужной линии управления. Для выхода из режима подконфигурации и возврата в режим глобальной конфигурации используйте команду **exit**.

```
Switch(config)# line console 0  
Switch(config-line)# exit  
Switch(config) #
```

Чтобы перейти из любого вложенного режима в рамках режима глобальной конфигурации на один уровень выше в иерархии режимов, введите команду **exit**.

Чтобы перейти из любого вложенного режима в привилегированный режим EXEC, введите команду **end** или используйте сочетание клавиш **Ctrl+Z**.

```
Switch(config-line)# end  
Switch#
```

Также можно напрямую переходить из одного вложенного режима конфигурации в другой. Обратите внимание, что после выбора интерфейса команда строка изменяется **(config-line) #** на **(config-if) #**.

```
Switch(config-line)# interface FastEthernet 0/1
Switch(config-if)#

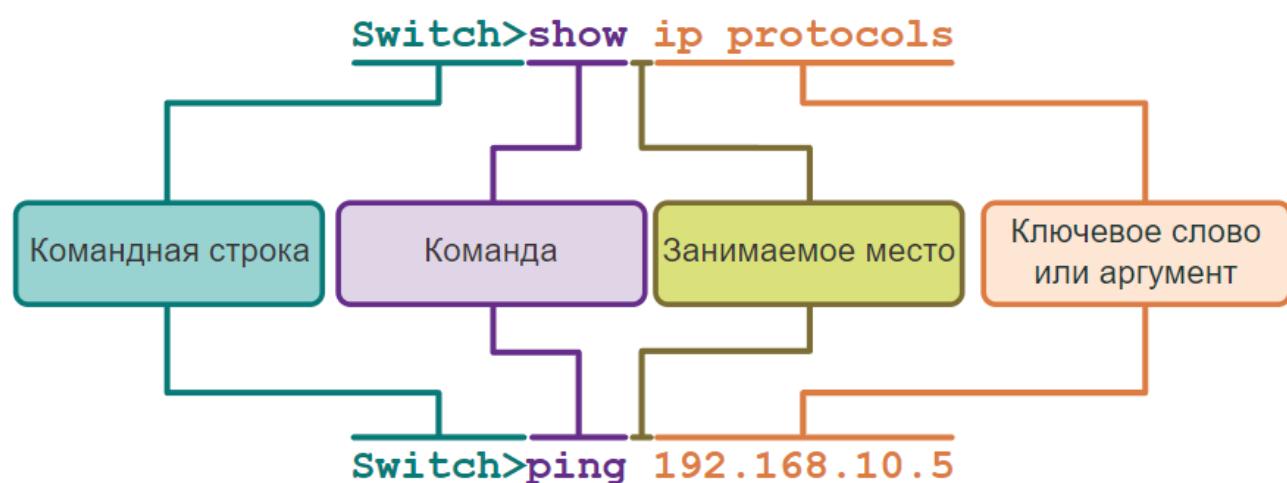
```

е. Работа в CLI IOS.

Базовая структура команд IOS

В этом разделе рассматривается основная структура команд для Cisco IOS. Администратор сети должен знать базовую структуру команд IOS, чтобы иметь возможность использовать интерфейс командной строки для конфигурации устройства.

Устройства Cisco IOS поддерживают множество команд. Каждая команда IOS имеет определенный формат или синтаксис и выполняется только в соответствующем режиме. Общий синтаксис команд представляет собой команду, за которой следуют любые необходимые ключевые слова и аргументы:



- **Ключевое слово** – это параметр, определенный в операционной системе (на рисунке **ip-протоколы**).
- **Аргумент** – Это значение или переменная, определенная пользователем (на рисунке **192.168.10.5**). После ввода каждой полной команды, включая все ключевые слова и аргументы, нажмите клавишу **Enter**, чтобы отправить эту команду в командный процессор.

Синтаксис команд

Подсказка (режим) # команда аргументы
[необязательный элемент]
{ обязательный элемент}
[элемент | {на | выбор}]

Условное обозначение	Описание
полужирный	Вводимые команды и ключевые слова отображаются полужирным шрифтом, как показано на рисунке.
курсив	Курсивом отображаются аргументы, для которых нужно указать значения.
[x]	В квадратных скобках отображаются дополнительные элементы (ключевое слово или аргумент).
{x}	В фигурных скобках отображаются обязательные элементы (ключевое слово или аргумент).
[x {y z}]	Фигурные скобки и вертикальные линии в квадратных скобках означают, что необходимо выбрать дополнительный элемент. Пробелы используются для четкого разграничения частей команды.

Ниже представлены примеры условных обозначений для документирования и использования команд IOS:

- **ping ip-address** – команда **ping**, а пользовательский аргумент – это ip-адрес целевого устройства. Например, **ping 10.10.10.5**.
- **traceroute ip-address** – команда traceroute, а определяемый пользователем аргумент является ip-адресом целевого устройства. Например, **traceroute 192.168.254.254**. Если команда сложная с несколькими аргументами, вы можете увидеть ее в следующем виде:

```
Switch(config-if)# switchport port-security aging { static | time time | type {absolute | inactivity}}
```

В сетевой операционной системе IOS предусмотрены две формы предоставления справочной информации: контекстная справка и проверка синтаксиса команд.

Контекстная справка позволяет быстро найти ответы на следующие вопросы:

- Какие команды доступны в каждом командном режиме?
- Какие команды начинаются с определенных символов или группы символов?
- Какие аргументы и ключевые слова доступны для определенных команд?

Для доступа к контекстной справке просто введите вопросительный знак **?** в командной строке CLI.

Проверка синтаксиса команды подтверждает, что пользователь ввел допустимую команду. Командный процессор анализирует введенную команду слева направо. Если процессор распознает команду, то выполняется требуемое действие и интерфейс CLI возвращается к соответствующей командной строке. Если процессор не распознает введенную команду, он отображает возможные ошибки.

Контекстная справка

В сетевой операционной системе IOS предусмотрены две формы предоставления справочной информации: **контекстная справка** и **проверка синтаксиса команд**

Контекстская справка позволяет быстро найти ответы на следующие вопросы:

- Какие команды доступны в каждом командном режиме?
- Какие команды начинаются с определенных символов или группы символов?
- Какие аргументы и ключевые слова доступны для определенных команд?

```
Router#ping ?
WORD  Ping destination address or hostname
ip    IP echo
ipv6  IPv6 echo
```

Проверка синтаксиса команды подтверждает, что пользователь ввел допустимую команду

- Если процессор не распознает введенную команду, он отображает возможные ошибки

```
Switch#interface fastEthernet 0/1
^
% Invalid input detected at '^' marker.
```

Имя по умолчанию должно быть изменено на более описательное. Если выбрать имена со смыслом, то запоминать, документировать и идентифицировать сетевые устройства будет легче. Вот несколько важных рекомендаций по выбору имени хоста:

- начинаться с буквы
- не содержать пробелов
- оканчиваться на букву или цифру
- содержать только буквы, цифры и тире
- состоять не более чем из 64 символов

После определения соглашения об именовании нужно присвоить устройствам имена с помощью CLI. Как показано в примере, из привилегированного режима EXEC перейдите в режим глобальной конфигурации с помощью команды **configure terminal**. Обратите внимание на изменение в диалоге командной строки.

```
Switch# configure terminal  
Switch(config)# hostname Sw-Floor-1  
Sw-Floor-1(config)#
```

В режиме глобальной конфигурации введите команду **hostname**, а затем имя коммутатора и нажмите клавишу **Enter**. Обратите внимание на изменение имени в диалоге командной строки.

Примечание: Чтобы удалить настроенное имя узла и вернуть стандартный диалог командной строки для коммутатора, используйте команду глобальной конфигурации **no hostname**.

```
Switch(config)# hostname hostname
```

Удаление имени устройства

```
S1(config)# no hostname Sw-Floor-1
```

Правила выбора паролей

В Cisco IOS можно настроить пароли иерархических режимов, чтобы предоставлять разные права доступа к сетевому устройству.

Все сетевые устройства должны ограничивать административный доступ, защищая привилегированный доступ EXEC, пользовательский EXEC и удаленный доступ Telnet с помощью паролей. Кроме того, все пароли должны быть зашифрованы и должны быть настроены уведомления о том, что лишь авторизованным пользователям можно получить доступ к устройству. Используйте надежные пароли, которые сложно подобрать. При выборе паролей необходимо учитывать некоторые ключевые моменты:

- Используйте пароли длиной более 8 символов.
- Используйте сочетание букв в верхнем и нижнем регистре, цифр, специальных символов и (или) числовых последовательностей.
- Не используйте одинаковый пароль для всех устройств.
- Не используйте часто употребляющиеся слова, поскольку их легко подобрать.

Используйте поиск в Интернете, чтобы найти генератор паролей. Многие из них позволят вам установить длину, набор символов и другие параметры.

Настройка паролей

Обеспечение безопасности пользовательского режима EXEC

```
S(config)# line console 0  
S(config-line)# password password  
S(config-line)# login
```

```
Sw-Floor-1# configure terminal  
Sw-Floor-1(config)# line console 0  
Sw-Floor-1(config-line)# password cisco  
Sw-Floor-1(config-line)# login  
Sw-Floor-1(config-line)# end  
Sw-Floor-1#
```

Безопасный привилегированный доступ EXEC

```
S(config)# enable secret password
```

```
Sw-Floor-1# configure terminal  
Sw-Floor-1(config)# enable secret class  
Sw-Floor-1(config)# exit  
Sw-Floor-1#
```

Для обеспечения доступа пользователя в режиме EXEC введите режим конфигурации консоли линии с помощью команды глобальной конфигурации **line console 0**, как показано в примере. Ноль используется для обозначения первого (а в большинстве случаев — единственного) интерфейса консоли. Затем задайте пароль пользовательского режима EXEC с помощью

команды **password** password. Наконец, включите доступ к пользовательскому режиму EXEC с помощью команды **login**.

Теперь для доступа к пользовательскому режиму EXEC с консоли будет необходим пароль. Чтобы иметь доступ администратора ко всем командам IOS, включая настройку устройства, необходимо получить привилегированный доступ режима EXEC. Это самый важный метод доступа, поскольку он обеспечивает полный доступ к устройству.

Для защиты доступа к привилегированному режиму EXEC используйте команду глобальной конфигурации **enable secret** password.

Линии виртуального терминала (VTY) обеспечивают удаленный доступ к устройству через Telnet или SSH. Большинство коммутаторов Cisco поддерживают до 16 линий VTY, пронумерованных от 0 до 15.

Чтобы защитить VTY, войдите в режим line VTY, используя команду глобальной конфигурации **line vty 0 15**. Затем задайте пароль VTY с помощью команды **password** password . Наконец, включите доступ к VTY с помощью команды **login**.

Пример защиты линий VTY на коммутаторе приведен ниже.

```
Sw-Floor-1# configure terminal
Sw-Floor-1(config)# line vty 0 15
Sw-Floor-1(config-line)# password cisco
Sw-Floor-1(config-line)# login
Sw-Floor-1(config-line)# end
Sw-Floor-1#
```

Шифрование паролей

Шифрование паролей в файле конфигурации

```
S(config)# service password-encryption
```

```
Sw-Floor-1# configure terminal
Sw-Floor-1(config)# service password-encryption
Sw-Floor-1(config)# exit
Sw-Floor-1#
```

```
Sw-Floor-1# show running-config
!
line con 0
password 7 094F471A1A0A
login
!
Line vty 0 4
Password 7 03095A0F034F38435B49150A1819
Login
!
end
```

Файлы конфигурации startup-config и running-config отображают большинство паролей в виде простого текста. Это создает угрозу безопасности, поскольку при наличии доступа к этим файлам любой пользователь может увидеть пароли.

Чтобы зашифровать все пароли открытого текста, используйте команду глобальной конфигурации **service password-encryption**, как показано в примере.

Команда применяет слабый алгоритм шифрования ко всем незашифрованным паролям.

Шифрование применяется только к паролям в файле конфигурации, но не к паролям, которые отправлены по сети. Эта команда не позволяет неавторизованным пользователям прочитать пароли в файле конфигурации.

С помощью команды **show running-config** убедитесь, что пароли зашифрованы.

Баннерные сообщения

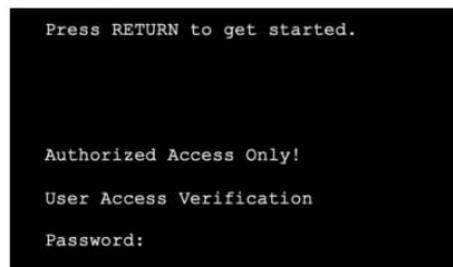
Баннерное сообщение важно для предупреждения несанкционированного персонала о попытке доступа к устройству

Настройка

```
S(config)# banner motd # the message of the day #
```

Разделителем может быть любой уникальный символ, которого нет в самом сообщении (например, #\\$%^&)

```
Sw-Floor-1# configure terminal  
Sw-Floor-1(config)# banner motd #Authorized Access Only!#
```



Хотя пароли защищают сеть от несанкционированного доступа, необходимо использовать уведомления о том, что лишь авторизованным пользователям можно получить доступ к устройству. Для этого нужно добавить баннер в выходные данные устройства. Баннеры могут стать важной частью судебного процесса, если пользователь был обвинен в несанкционированном доступе. Отдельные законодательства не разрешают возбуждать судебные дела против пользователей или даже просто следить за их действиями без предупреждения.

Чтобы создать баннерное сообщение дня на сетевом устройстве, используйте команду глобальной конфигурации **banner motd # the message of the day #**. Символ «#» в синтаксисе команды называется разделителем. Он вводится до и после сообщения. Разделителем может быть любой символ, которого нет в самом сообщении. Поэтому часто используются такие символы, как «#».

После выполнения команды баннер будет отображаться при всех последующих попытках доступа к устройству, пока не будет удален.

В следующем примере показаны шаги по настройке баннера на Sw-Floor-1.

Конфигурация устройства хранится в двух системных файлах.

- **startup-config** - Это сохраненный файл конфигурации, который хранится в NVRAM. Он содержит все команды, которые будут использоваться при загрузке или перезагрузке. Содержимое Флеш-накопителя не теряется при выключении питания устройства.
- **running-config** - Это файл текущей конфигурации, хранится в оперативной памяти (RAM). Он отражает текущую конфигурацию. Изменения текущей конфигурации незамедлительно влияют на работу устройства Cisco. ОЗУ — энергозависимая память. После отключения питания или перезагрузки устройства ОЗУ теряет все свое содержимое.

Команда привилегированного режима EXEC **show running-config** используется для просмотра текущей конфигурации. Как показано в примере, команда выведет список полной конфигурации, хранящейся в настоящее время в ОЗУ.

```
Sw-Floor-1# show running-config
Building configuration...
Current configuration : 1351 bytes
!
! Last configuration change at 00:01:20 UTC Mon Mar 1 1993
!
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname Sw-Floor-1
!
(output omitted)
```

Для просмотра файла загрузочной конфигурации используйте команду привилегированного режима EXEC **show startup-config**.

Однако при отключении питания или перезапуске устройства все не сохраненные изменения конфигурации будут потеряны. Чтобы сохранить изменения текущей конфигурации в файле загрузочной конфигурации, используйте команду привилегированного режима EXEC **copy running-config startup-config**.

Если изменения текущей конфигурации не принесли желаемых результатов и файл **running-config** пока не был сохранен, можно сделать следующее. Удалите команды по отдельности или перезагрузите устройство с помощью команды привилегированного режима EXEC **reload** для загрузки из файла начальной конфигурации.

Недостатком использования команды **reload** для удаления не сохраненной текущей конфигурации является кратковременный переход устройства в автономный режим и, как следствие, простой сети. Выполняя перезагрузку, IOS определит, что изменения текущей конфигурации не были сохранены в файл начальной конфигурации. Появится сообщение с вопросом, нужно ли сохранить изменения. Для отмены изменений введите **n** или **no**.

Если нежелательные изменения были сохранены в файл начальной конфигурации, возможно, придется удалить все конфигурации. Для этого нужно удалить начальную конфигурацию и перезапустить устройство. Загрузочную конфигурацию можно удалить с помощью команды привилегированного режима EXEC **erase startup-config**. После ввода команды появится запрос о подтверждении. Нажмите клавишу **Enter** для подтверждения.

Интерфейсы и порты

Сетевой обмен данными зависит от интерфейсов оконечных пользовательских устройств, интерфейсов сетевых устройств и кабелей, при помощи которых они соединены. Каждый физический интерфейс определяется своими техническими характеристиками (стандартами). Соединяющий кабель должен соответствовать физическим стандартам интерфейса. Существует несколько типов средств сетевого подключения: медные кабели на основе витой пары, оптоволоконные кабели, коаксиальные кабели или средства беспроводного подключения, как показано на рисунке.

Каждый канал связи в Интернете не только требует особого типа средства сетевого подключения, но и отдельной сетевой технологии. Например, Ethernet — наиболее распространенная технология локальной сети на сегодняшний день. Порты Ethernet есть на устройствах конечных пользователей, коммутаторах и других сетевых устройствах, которые поддерживают физическое подключение к сети с помощью кабеля.

Коммутаторы Cisco IOS уровня 2 оснащены физическими портами для подключения устройств. Эти порты не поддерживают IP-адреса уровня 3. Поэтому коммутаторы имеют один или несколько виртуальных интерфейсов (switch virtual interface, SVI). Такие интерфейсы называются виртуальными, поскольку на устройстве нет связанного с ними физического оборудования. SVI создается в программном обеспечении.

Виртуальный интерфейс позволяет удаленно управлять коммутатором по сети с помощью протокола IPv4 или IPv6. Каждый коммутатор поставляется с одним интерфейсом SVI в конфигурации по умолчанию. Виртуальным интерфейсом по умолчанию является VLAN 1.

Примечание: Для работы коммутатора уровня 2 не нужен IP-адрес. IP-адрес, назначенный SVI, используется для удаленного доступа к коммутатору. Для работы коммутатора IP-адрес не нужен.

Настройка виртуального интерфейса коммутатора (SVI)

Для удаленного доступа к коммутатору на интерфейсе (SVI) нужно настроить IP-адрес и маску подсети. Чтобы настроить SVI на коммутаторе, используйте команду глобальной конфигурации **interface vlan 1**. Vlan 1 — это не реальный физический интерфейс, а виртуальный. Затем назначьте адрес IPv4 с помощью команды конфигурации интерфейса **ip address ip-address subnet-mask**. Наконец, включите виртуальный интерфейс с помощью команды конфигурации интерфейса **no shutdown**.

После настройки этих команд все элементы IPv4 в коммутаторе будут готовы для передачи данных по сети.

```
Sw-Floor-1# configure terminal
Sw-Floor-1(config)# interface vlan 1
Sw-Floor-1(config-if)# ip address 192.168.1.20 255.255.255.0
Sw-Floor-1(config-if)# no shutdown
Sw-Floor-1(config-if)# exit
Sw-Floor-1(config)# ip default-gateway 192.168.1.1
```

Команды **show ip interface brief**. Эта команда позволяет проверить состояние интерфейсов коммутаторов.

FastEthernet0/24	unassigned	YES manual down	down
GigabitEthernet0/1	unassigned	YES manual down	down
GigabitEthernet0/2	unassigned	YES manual down	down
Vlan1	192.168.1.2	YES manual administratively down	down

Проверка настроек

Команды проверки

```
S1# show ip interface brief
S1# show running-config
S1# show running-config interface Vlan1
S1# show history

> ipconfig
```

show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.254.254	YES	NVRAM	up	up
FastEthernet0/1	unassigned	YES	unset	down	down
Serial0/0/0	172.16.0.254	YES	NVRAM	up	up
Serial0/0/1	unassigned	YES	unset	administratively down	down

Удаление конфигурации

```
S1# erase startup-config
```

Перезагрузка

```
S1# reload
```

«Джентльменский» набор команд

Синхронизация логирования

```
R(config)# line console 0
R(config-line)# logging synchronous
```

Отключение разрешения доменного имени

```
R(config)# no ip domain-lookup
```

3. Введение в маршрутизацию:

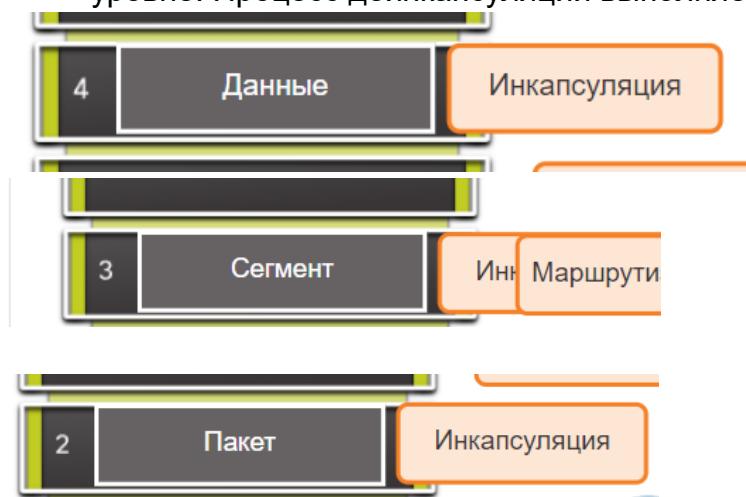
a. Сетевой уровень: задачи.

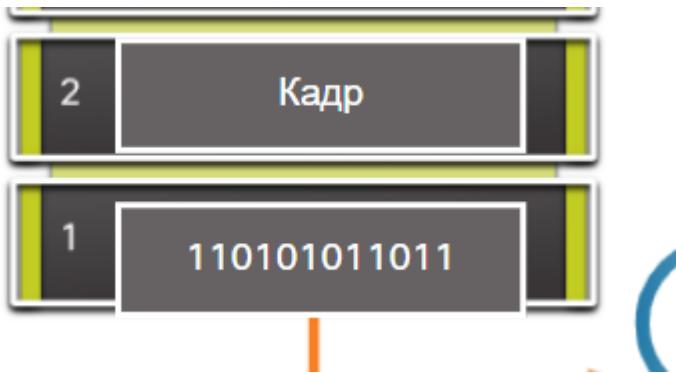
Сетевой уровень, или третий уровень модели OSI, предоставляет сервисы, позволяющие оконечным устройствам обмениваться данными по сети. Как показано на рисунке, IP версии 4 (IPv4) и IP версии 6 (IPv6) являются протоколами связи основного сетевого уровня. Другие протоколы сетевого уровня включают протоколы маршрутизации, такие как Open Shortest Path First (OSPF), и протоколы обмена сообщениями, такие как Internet Control Message Protocol (ICMP).



Для выполнения сквозных коммуникаций через границы сети протоколы сетевого уровня выполняют четыре основные операции:

- **Адресация оконечных устройств** - Оконечным устройствам необходимо назначить уникальный IP-адрес для возможности их идентификации в сети.
- **Инкапсуляция** - Сетевой уровень получает единицу данных протокола (PDU) от транспортного уровня. Во время выполнения процесса, который называется инкапсуляцией, сетевой уровень добавляет информацию заголовка IP, например IP-адрес узла источника (отправляющего) и узла назначения (получающего). Процесс инкапсуляции выполняется источником IP-пакета.
- **Маршрутизация** - Сетевой уровень предоставляет сервисы, с помощью которых пакеты направляются к узлу назначения в другой сети. Для перемещения к другим сетям пакет должен быть обработан маршрутизатором. Роль маршрутизатора заключается в том, чтобы выбрать пути для пакетов и направить их к узлу назначения. Такой процесс называется маршрутизацией. До того как достигнуть узла назначения, пакет может пройти через несколько маршрутизаторов. Каждый маршрут на пути пакета к узлу назначения называется переходом.
- **Деинкапсуляция** - По прибытии пакета на сетевой уровень узла назначения этот узел проверяет IP-заголовок пакета. Если IP-адрес назначения в заголовке совпадает с его собственным IP-адресом, заголовок IP удаляется из пакета. После деинкапсуляции пакета, выполняемой сетевым узлом, полученная единица данных протокола (PDU) уровня 4 пересыпается соответствующей службе на транспортном уровне. Процесс деинкапсуляции выполняется конечным узлом IP-пакета.

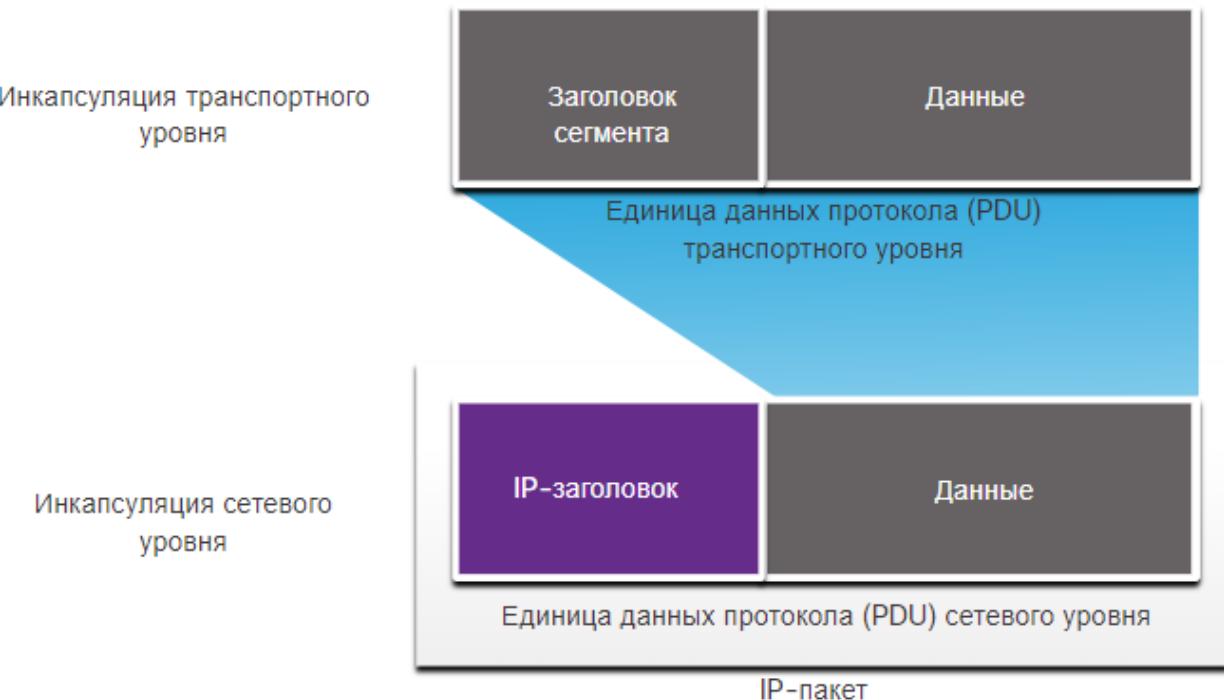




Инкапсуляция IP

Протокол IP инкапсулирует сегмент транспортного уровня (уровень чуть выше сетевого уровня) или другие данные путем добавления заголовка IP. IP заголовок используется для доставки пакета на узел назначения.

На рисунке показан последующий процесс создания единицы данных протокола (PDU) сетевого уровня и IP-пакета.



Процесс инкапсуляции данных от уровня к уровню обеспечивает возможность разрабатывать и масштабировать сервисы на различных уровнях без влияния на другие уровни. Это означает, что сегменты транспортного уровня можно легко упаковать с помощью протоколов IPv4 или IPv6 или любого нового протокола, который может быть создан в будущем.

IP-заголовок проверяется устройствами уровня 3 (т.е. маршрутизаторами и коммутаторами уровня 3), когда он перемещается по сети к месту назначения. Важно отметить, что информация об IP-адресации остается неизменной с момента выхода пакета с исходного хоста до момента его прибытия на хост назначения, за исключением случаев, когда она переводится устройством, выполняющим преобразование сетевых адресов (NAT) для IPv4.

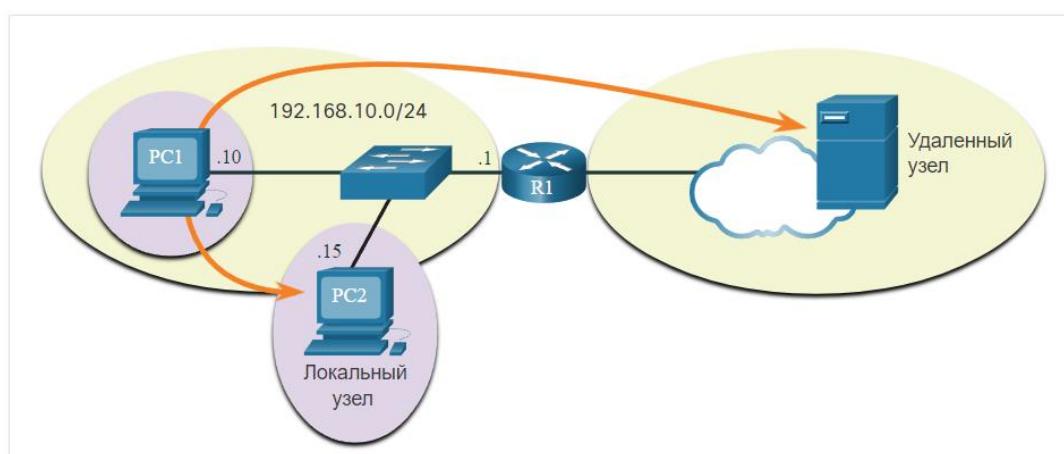
b. Маршрутизация на хостах. Таблица маршрутизации хоста.

С IPv4 и IPv6 пакеты всегда создаются на исходном хосте. Хост источника должен иметь возможность направлять пакет на хост назначения. Для этого хост-конечные устройства создают собственную таблицу маршрутизации

Другим предназначением сетевого узла является пересылка пакетов между узлами. Узел может отправить пакет на следующие адреса:

- **Себе** - хост может пинговать себя посыпая пакеты на специальный IPv4-адрес 127.0.0.1 или IPv6-адрес ::1, который называется интерфейсом обратной связи. Отправка эхозапроса на интерфейс loopback тестирует стек протокола TCP/IP на узле.
- **Локальный узел**. Узел в той же локальной сети, в которой также находится отправляющий узел. Хосты источника и назначения используют один и тот же сетевой адрес.
- **Удаленный узел**. Узел в удаленной сети. Хосты источника и назначения не используют один и тот же сетевой адрес.

На рисунке показано подключение PC1 к локальному узлу в той же сети и к удаленному узлу, расположенному в другой сети.



Определяет, предназначен ли пакет для локального узла или удаленного узла, определяется конечным устройством источника. Конечное устройство источника определяет, находится ли конечный IP-адрес в той же сети, в которой находится само устройство источника. Метод определения варьируется в зависимости от версии IP:

- **В IPv4** исходное устройство использует собственную маску подсети вместе с собственным адресом IPv4 и адресом назначения IPv4 для определения того, находится ли узел назначения в одной сети.

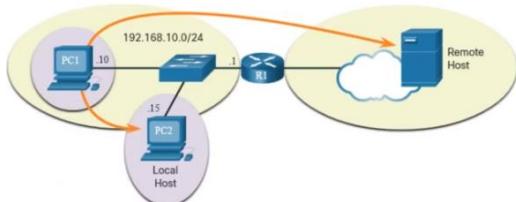
Берем свою маски и ip, смотрим какая сеть, потом смотрим тоже самое для ip адреса назначения, если сеть совпадает, то устройство находится в локальной сети.

- **В IPv6** — локальный маршрутизатор объявляет адрес локальной сети (префикс) всем устройствам в сети.

В домашней или корпоративной сети могут находиться несколько проводных и беспроводных устройств, соединенных друг с другом с помощью промежуточного устройства, такого как коммутатор локальной сети (LAN) или точка беспроводного доступа (WAP). Это промежуточное устройство обеспечивает соединение между локальными узлами в локальной сети. Локальные узлы могут получать доступ друг к другу и обмениваться информацией без использования каких-либо дополнительных устройств. Если узел отправляет пакет устройству, которое настроено в этой же IP-сети в качестве главного устройства, пакет просто пересыпается из интерфейса узла через промежуточное устройство прямо на устройство назначения.

Разумеется, в большинстве случаев нам требуется, чтобы наши устройства могли устанавливать соединения за пределами сегмента локальной сети: подключаться к другим домам, офисам и Интернету. Устройства, которые не входят в сегмент локальной сети, называются удаленными узлами. Если исходное устройство отправляет пакет к удаленному устройству назначения, то в этом случае требуется помочь маршрутизаторов и выполнение маршрутизации. Маршрутизация — это процесс определения оптимального пути к хосту назначения. Маршрутизатор, подключенный к сегменту локальной сети, называется шлюзом по умолчанию.

Маршрутизация на хостах



Исходное устройство определяет, является ли место назначения локальным или удаленным

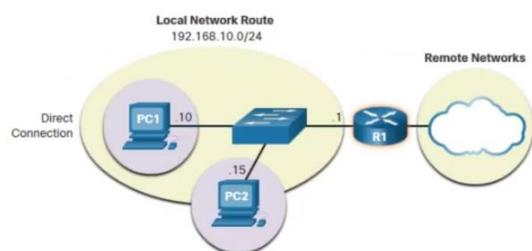
Метод определения:

- IPv4 — Источник использует свой собственный IP-адрес и маску подсети, а также IP-адрес назначения
- IPv6 — Источник использует сетевой адрес и префикс, объявленные локальным маршрутизатором

Локальный трафик отправляется из интерфейса хоста для обработки промежуточным устройством

Удаленный трафик перенаправляется непосредственно на шлюз по умолчанию в локальной сети

Маршрутизация на хостах



Маршрутизатор или коммутатор уровня 3 может быть шлюзом по умолчанию.

Особенности шлюза по умолчанию (DGW):

- Он должен иметь IP-адрес в том же диапазоне, что и остальная часть локальной сети
- Он может принимать данные из локальной сети и способен перенаправлять трафик из локальной сети
- Он может маршрутизировать в другие сети

Если устройство не имеет шлюза по умолчанию, его трафик не сможет покинуть локальную сеть.

Шлюз по умолчанию — это сетевое устройство (т.е. маршрутизатор или коммутатор уровня 3), которое направляет трафик в другие сети. Если в качестве аналогии сети использовать комнату, шлюзом по умолчанию будет входная дверь. Для того чтобы попасть в другую комнату или сеть, нужно найти входную дверь.

В сети шлюзом по умолчанию обычно является маршрутизатор со следующими функциями:

- Имеет локальный IP-адрес в том же диапазоне адресов, что и другие хосты в сети.
- Он может принимать данные в локальную сеть и пересыпать данные из локальной сети.
- Направляет трафик в другие сети.

Шлюз по умолчанию необходим для отправки трафика за пределы локальной сети. Трафик не может быть перенаправлен за пределы локальной сети, если отсутствует шлюз по умолчанию, адрес шлюза по умолчанию не настроен или шлюз по умолчанию отключен.

Таблица маршрутизации узла, как правило, содержит шлюз по умолчанию. В IPv4, узел получает IPv4-адрес шлюза по умолчанию динамически от протокола динамической настройки узла (DHCP) или из вручную настроенных параметров. В IPv6 маршрутизатор объявляет адрес шлюза по умолчанию, или узел можно настроить вручную.

На рисунке компьютеры PC1 и PC2 настроены на использование шлюза по умолчанию с IPv4-адресом 192.168.10.1.



В таблице маршрутизации ПК при наличии настроенного шлюза по умолчанию создается маршрут по умолчанию. Маршрут по умолчанию — маршрут или путь, по которому идет компьютер, когда он пытается связаться с удаленной сетью.

И компьютер PC1, и компьютер PC2 будут иметь маршрут по умолчанию для отправки всего трафика, предназначенного для удаленных сетей, к маршрутизатору R1.

Таблицы маршрутизации хоста

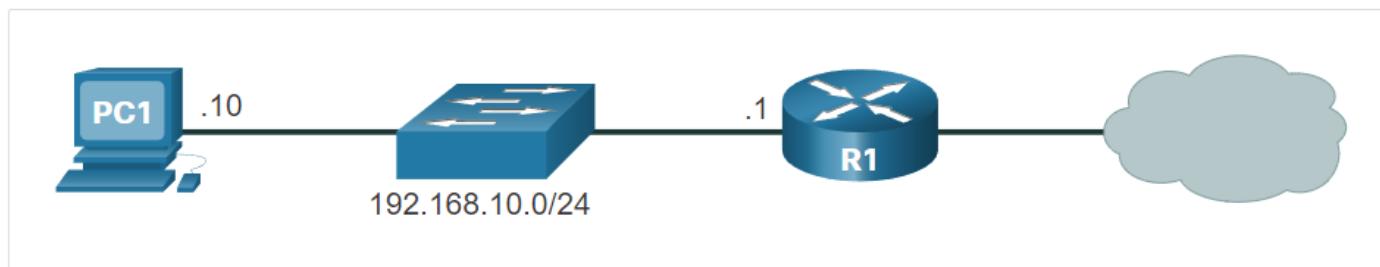


IPv4 Routing Table for PC1

IPv4 Route Table					
<hr/>					
Active Routes:					
Network Destination	Netmask	Gateway	Interface	Metric	
0.0.0.0	0.0.0.0	192.168.10.1	192.168.10.10	25	
127.0.0.0	255.0.0.0	On-link	127.0.0.1	306	
127.0.0.1	255.255.255.255	On-link	127.0.0.1	306	
127.255.255.255	255.255.255.255	On-link	127.0.0.1	306	
192.168.10.0	255.255.255.0	On-link	192.168.10.10	281	
192.168.10.10	255.255.255.255	On-link	192.168.10.10	281	
192.168.10.255	255.255.255.255	On-link	192.168.10.10	281	
224.0.0.0	240.0.0.0	On-link	127.0.0.1	306	
224.0.0.0	240.0.0.0	On-link	192.168.10.10	281	
255.255.255.255	255.255.255.255	On-link	127.0.0.1	306	
255.255.255.255	255.255.255.255	On-link	192.168.10.10	281	

На узле Windows команда **route print** или **netstat -r** можно использовать для отображения таблицы маршрутизации узла. Обе команды выдают одинаковый результат.

На рисунке отображается образец топологии и выходные данные, созданные командой **netstat -r**.



- **Список интерфейса.** Содержит адрес управления доступом к среде (MAC-адрес) и присвоенный номер интерфейса с поддержкой сети на узле, включая адAPTERы Ethernet, Wi-Fi и Bluetooth.
- **Таблица маршрутизации IPv4.** Содержит все известные маршруты IPv4, включая прямые подключения, локальные сети и локальные маршруты, используемые по умолчанию.
- **Таблица маршрутизации IPv6.** Содержит все известные маршруты IPv6, включая прямые подключения, локальные сети и локальные маршруты, используемые по умолчанию.

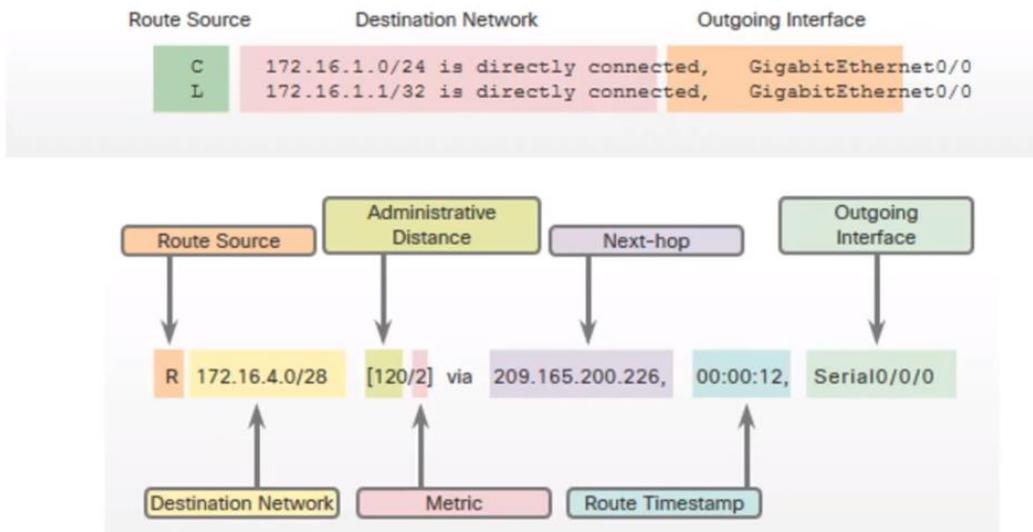
Таблица маршрутизации содержит список маршрутов к известным сетям (префиксы и длины префикса). Источником этой информации являются:

- непосредственно подключенные сети;
- статические маршруты;
- протоколов динамической маршрутизации.

Существует три принципа таблицы маршрутизации, как описано в таблице. Это проблемы, которые решаются правильной конфигурацией протоколов динамической маршрутизации или статических маршрутов на всех маршрутизаторах между устройствами источника и назначения.

Принципы таблицы маршрутизации	Пример
Каждый маршрутизатор принимает решение самостоятельно, основываясь на информации, которую он имеет в своей собственной таблице маршрутизации.	R1 может пересыпать пакеты только с помощью собственной таблицы маршрутизации. R1 не знает, какие маршруты находятся в таблицах маршрутизации других маршрутизаторов (например, R2).
Информация в таблице маршрутизации одного маршрутизатора не обязательно схожа с таблицей маршрутизации другого маршрутизатора.	Просто потому, что R1 имеет маршрут в своей таблице маршрутизации к сети в Интернет через R2, это не означает, что R2 знает об этой же сети.
Информация о маршруте в одну сторону, не гарантирует информацию о маршруте в обратном направлении.	R1 получает пакет с IP-адресом назначения PC1 и IP-адрес источником PC3. Просто потому, что R1 знает, как переслать пакет из интерфейса G0/0/0, не обязательно означает, что он знает как перенаправлять пакеты, исходящие от PC1, обратно в удаленную сеть PC3.

Записи таблицы маршрутизации



1. Источник маршрута (удаленная(static, rip) или напрямую подключенная (local, connected))
2. Сеть назначения
3. Степень доверия источнику маршрута. Чем меньше число, тем больше доверяют
4. Метрика (расстояние до сети). Чем меньше метрика, тем оптимальнее маршрут.
5. Адрес следующего маршрутизатора
6. Исходящий интерфейс

Administrative Distance

Route Source	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
Internal BGP	200

Команда ip route

```
R(config)# ip route network-address subnet-mask  
{ip-address | exit-intf} [distance]
```

с. Маршрутизация в общем смысле. Типы маршрутов.

Маршрутизатор



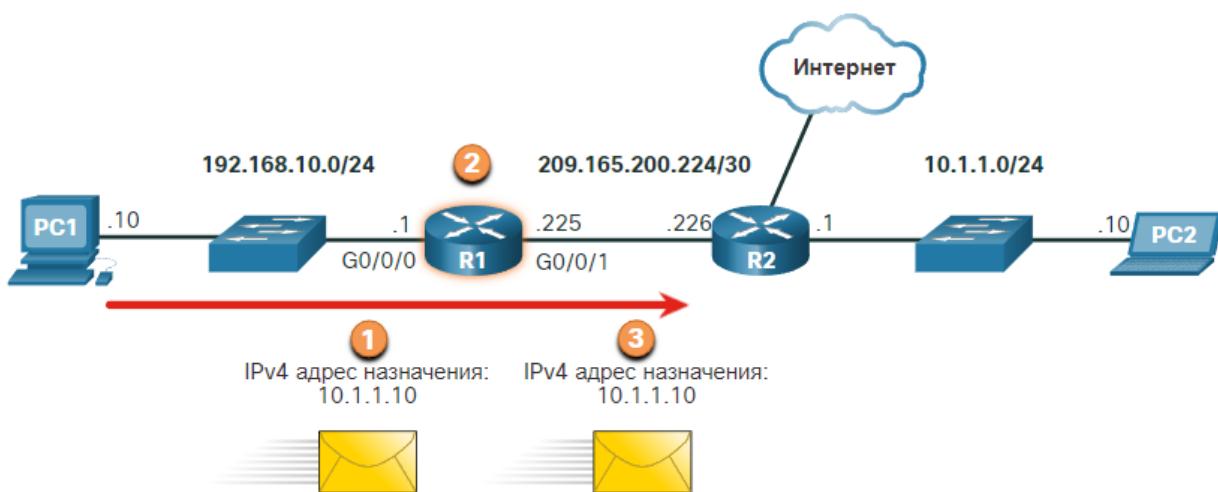
Маршрутизатор:

- Подключает одну сеть к другой
- Определяет оптимальный маршрут к месту назначения перед пересылкой трафика на следующий маршрутизатор в пути
- Отвечает за трафик маршрутизации между сетями
- Для определения наиболее эффективного пути к месту назначения используется таблица маршрутизации

Когда маршрутизатор получает IP-пакет на одном интерфейсе, он определяет, какой интерфейс следует использовать для пересылки пакета до места назначения. Это называется маршрутизация

Когда пакет поступает на интерфейс маршрутизатора:

Маршрутизатор считывает IP-адрес назначения и просматривает свою таблицу маршрутизации, определяя, куда нужно переслать пакет. Таблица маршрутизации содержит список всех известных сетевых адресов (префиксов) и куда пересылать пакет. Эти записи известны как записи маршрута или маршруты. Маршрутизатор пересыпает пакет, используя наилучшую соответствующую запись маршрута.



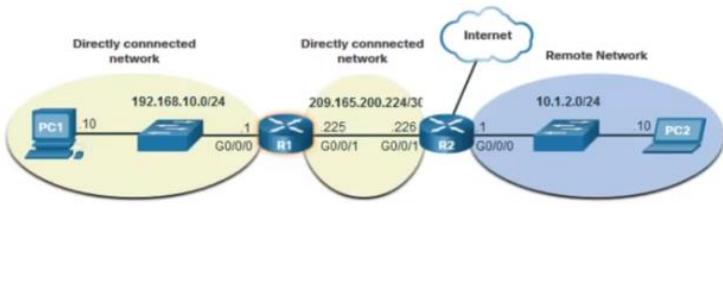
- Пакет поступает на интерфейс Gigabit Ethernet 0/0/0 маршрутизатора R1. R1 деинкапсулирует заголовок Ethernet уровня 2 и концевик.
- Маршрутизатор R1 проверяет адрес назначения IPv4 пакета и ищет наилучшее соответствие в своей таблице маршрутизации IPv4. Запись маршрута указывает, что этот пакет должен быть перенаправлен на маршрутизатор R2.
- Маршрутизатор R1 инкапсулирует пакет в новый заголовок и концевик Ethernet и пересыпает пакет на маршрутизатор следующего хопа - R2.

В следующей таблице приведены соответствующие сведения из таблицы маршрутизации R1.

R1 Routing Table

Маршрут	Адрес следующего перехода или исходящий интерфейс
192.168.10.0 /24	G0/0/0
209.165.200.224/30	G0/0/1
10.1.1.0/24	через R2
Маршрут по умолчанию 0.0.0.0/0	через R2

Типы маршрутов

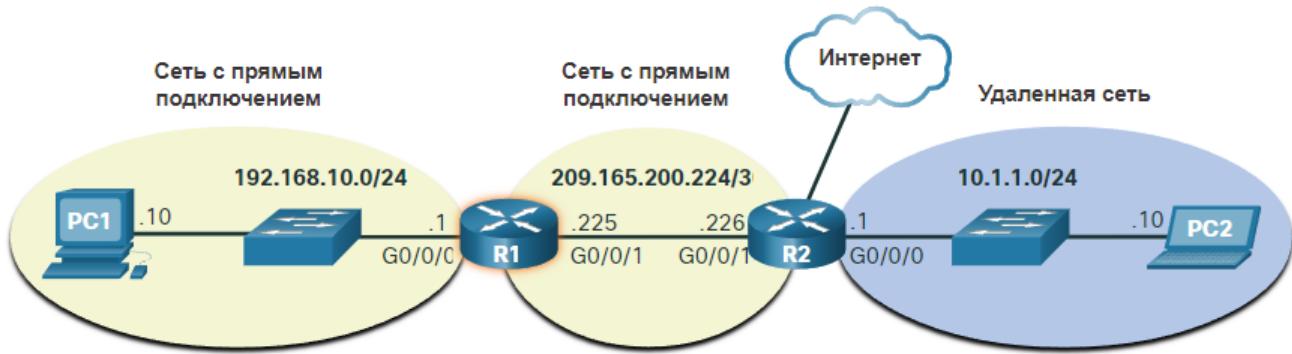


В таблице маршрутизации маршрутизатора есть три типа маршрутов:

- **Напрямую подключенные (Directly Connected)** — эти маршруты автоматически добавляются маршрутизатором при условии, что интерфейс активен и имеет адресацию
- **Удаленные маршруты (Remote Networks)** — эти маршруты не имеет прямого подключения, они могут быть изучены:
 - Вручную — со статическим маршрутом
 - Динамически — используя протокол маршрутизации, чтобы маршрутизаторы делились информацией друг с другом
- **Маршрут по умолчанию (Default Route)** — перенаправляет весь трафик в определенный интерфейс, если в таблице маршрутизации нет совпадения

В таблице маршрутизации хранятся три типа записей маршрута:

- **Сети с прямым подключением (Directly-connected networks)**- эти записи сетевого маршрута являются активными интерфейсами маршрутизатора. Маршрутизаторы добавляют маршрут с прямым подключением, когда интерфейс настроен с IP-адресом и активирован. Каждый из интерфейсов маршрутизатора подключен к разному сегменту сети. На рисунке сети с прямым подключением в таблице маршрутизации R1 IPv4 будут иметь значения 192.168.10.0/24 и 209.165.200.224/30.
- **Удаленные сети** — это сети, подключенные к другим маршрутизаторам. Маршрутизаторы узнают о удаленных сетях либо путем явной настройки администратором, либо путем обмена информацией о маршрутах с помощью протокола динамической маршрутизации. На рисунке удаленная сеть в таблице маршрутизации R1 IPv4 будет иметь значение 10.1.1.0/24.
- **Маршрут по умолчанию** — как и узел, большинство маршрутизаторов также включают запись маршрута по умолчанию, в качестве последнего средства, если иного маршрута до нужной сети в таблице маршрутизации нет. Маршрут по умолчанию используется, если в таблице IP-маршрутизации нет лучшего (наибольшего) соответствия. На рисунке таблица маршрутизации R1 IPv4, скорее всего, будет включать маршрут по умолчанию для пересылки всех пакетов маршрутизатору R2.



R1 имеет две сети прямого подключения:

- 192.168.10.0/24
- 209.165.200.224/30

R1 также имеет удаленные сети (например, 10.1.1.0/24 и Интернет), о которых он может узнать.

Маршрутизатор может узнать о удаленных сетях одним из двух способов:

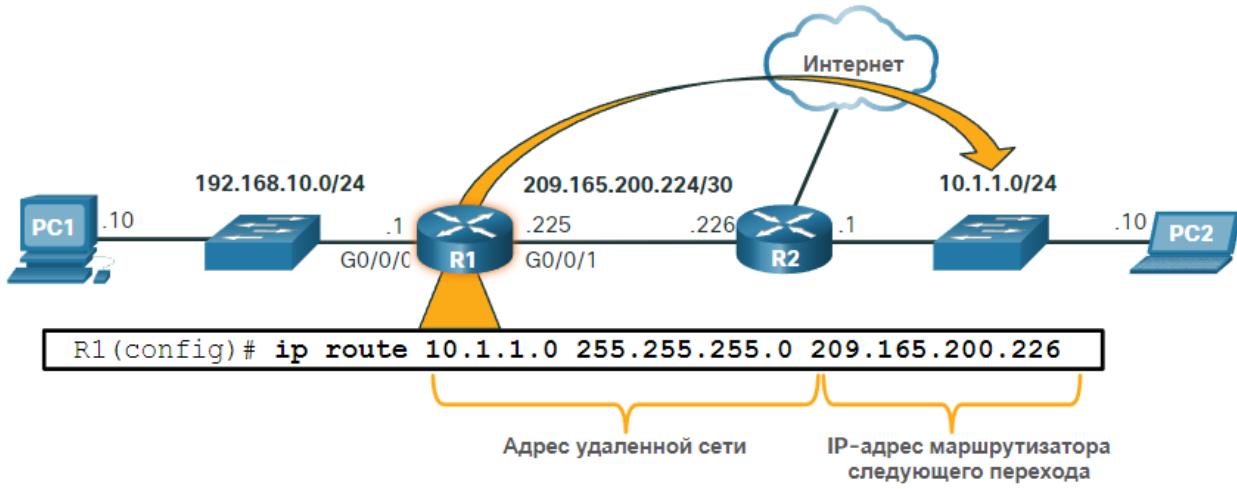
- **Вручную.** Данные об удаленных сетях вручную вводятся в таблицу маршрутов с использованием статических маршрутов.
- **Динамически.** Удаленные маршруты автоматически добавляются с использованием протокола динамической маршрутизации.

d. Статическая и динамическая маршрутизации.

Статическая vs Динамическая

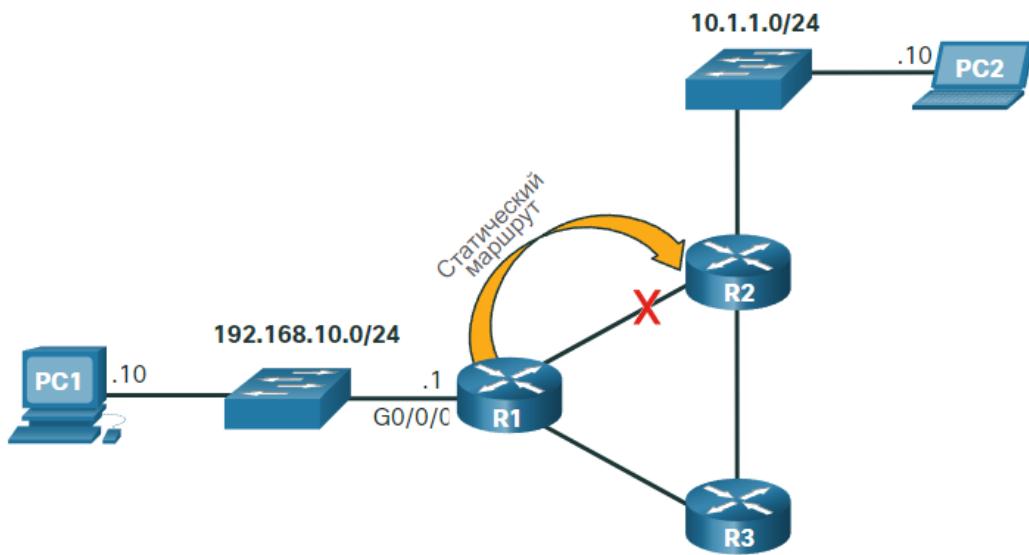
Функция	Динамическая маршрутизация	Статическая маршрутизация
Сложность конфигурирования	Не зависит от размера сети	Повышается с увеличением размера сети
Изменения топологии	Изменяется автоматически в соответствии с изменениями топологии	Требуется участие администратора
Масштабируемость	Подходит для простых и сложных топологий	Подходит для простых топологий
Информационная безопасность	Безопасность должна быть настроена	Безопасность является неотъемлемым элементом
Потребление ресурсов	Использует центральный процессор, память, пропускную способность канала	Никаких дополнительных ресурсов не требуется
Предсказуемость пути	Маршрут зависит от используемой топологии и протокола маршрутизации	Явно определяется администратором

Статические маршруты - это записи маршрутов, которые настраиваются вручную. На рисунке показан пример статического маршрута, настроенного вручную на маршрутизаторе R1. Статический маршрут включает в себя адрес удаленной сети и IP-адрес маршрутизатора следующего перехода.



R1 настраивается вручную со статическим маршрутом для достижения сети 10.1.1.0/24. Если этот путь изменится, R1 потребуется новый статический маршрут.

В случае изменения топологии сети статический маршрут не обновляется автоматически и должен быть перенастроен вручную. Например, на рисунке R1 имеет статический маршрут для достижения сети 10.1.1.0/24 через R2. Если этот путь больше не доступен, потребуется перенастроить R1 на новый статический маршрут к сети 10.1.1.0/24 через R3. Поэтому маршрутизатор R3 должен иметь запись маршрута в таблице маршрутизации для отправки пакетов, предназначенных для 10.1.1.0/24, на R2.



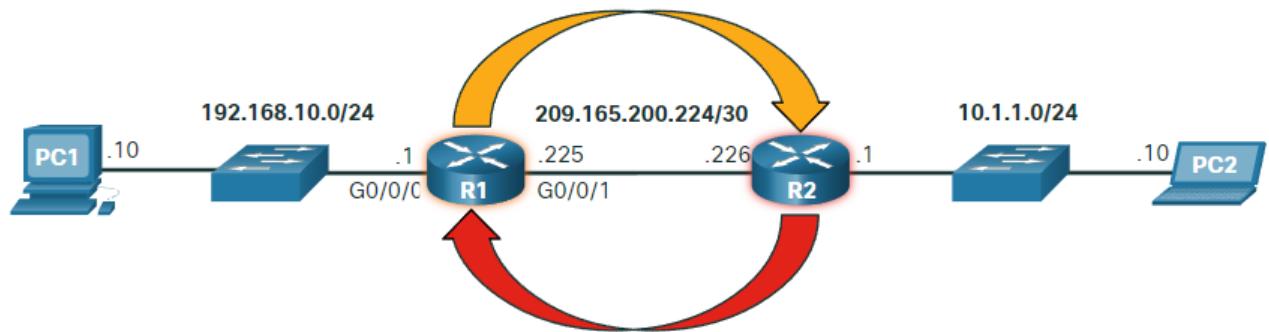
Если маршрут от R1 через R2 больше недоступен, необходимо настроить новый статический маршрут через R3. Статический маршрут не корректируется автоматически для изменения топологии.

Статическая маршрутизация имеет следующие характеристики:

- Статический маршрут должен быть настроен вручную.
- Администратору необходимо перенастроить статический маршрут, если есть изменения в топологии и статический маршрут больше не является жизнеспособным.
- Статический маршрут подходит для небольшой сети и когда избыточных каналов мало или нет.
- Статический маршрут обычно используется с протоколом динамической маршрутизации для настройки маршрута по умолчанию.

Протокол динамической маршрутизации позволяет маршрутизаторам автоматически получать информацию о удаленных сетях, включая маршрут по умолчанию, от других маршрутизаторов. Маршрутизаторы, использующие протоколы динамической маршрутизации, автоматически обмениваются информацией о маршрутизации с другими маршрутизаторами и выполняют обновления в случае каких-либо изменений в топологии без участия сетевого администратора. При изменении топологии сети маршрутизаторы совместно используют эту информацию с помощью протокола динамической маршрутизации и автоматически обновляют свои таблицы маршрутизации.

Протоколы динамической маршрутизации включают OSPF и расширенный протокол маршрутизации внутреннего шлюза (EIGRP). На рисунке показан пример того, как маршрутизаторы R1 и R2 автоматически используют сетевую информацию с помощью протокола маршрутизации OSPF

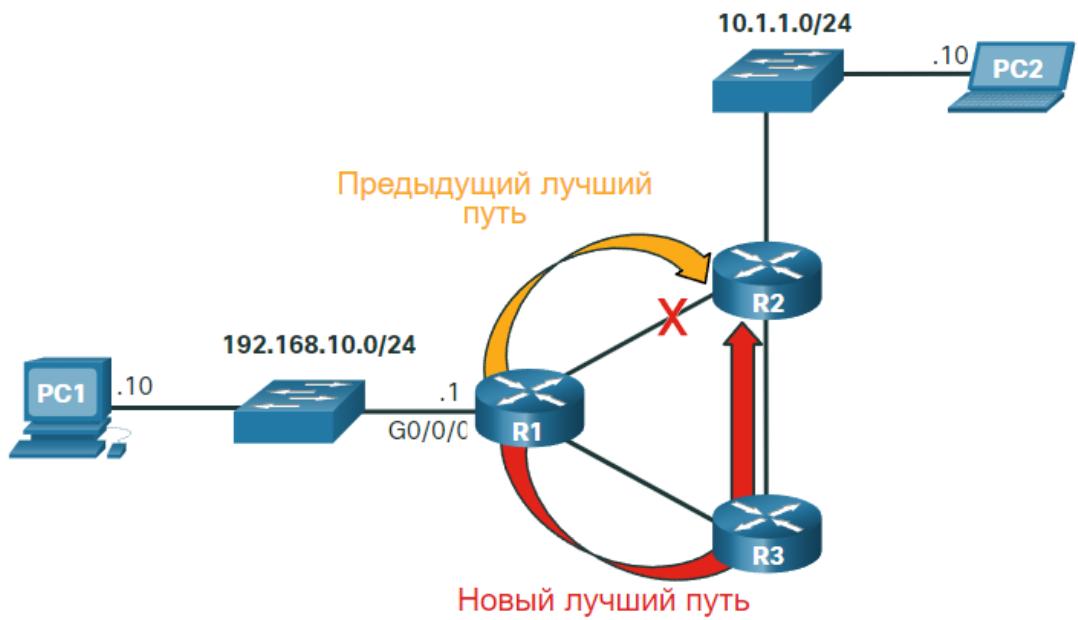


- R1 использует протокол маршрутизации OSPF, чтобы сообщить R2 о сети 192.168.10.0/24.
- R2 использует протокол маршрутизации OSPF, чтобы сообщить R1 о сети 10.1.1.0/24.

Базовая конфигурация требует, чтобы администратор сети включил непосредственно подключенные сети в рамках протокола динамической маршрутизации. Протокол динамической маршрутизации будет автоматически выполнять следующие действия:

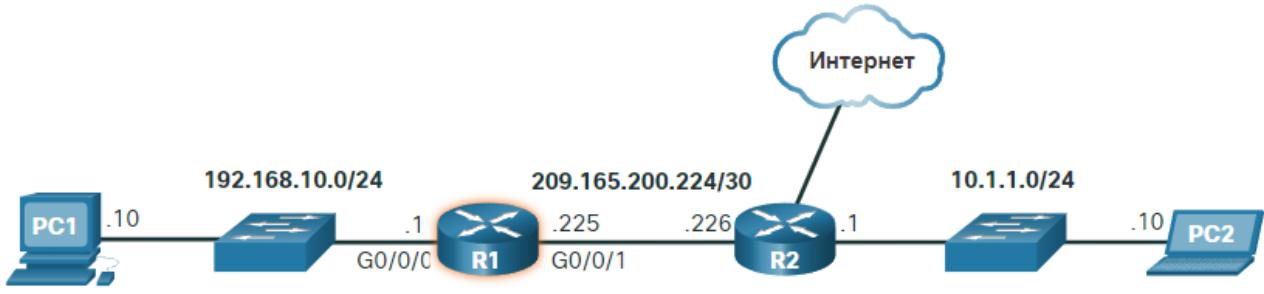
- Обнаружение удаленных сетей;
- Обновление данных маршрутизации;
- Выбор оптимального пути к сетям назначения;
- Поиск нового оптимального пути в случае, если текущий путь недоступен.

Если маршрутизатор вручную настроен со статическим маршрутом или узнает о удаленной сети динамически с помощью протокола динамической маршрутизации, адрес удаленной сети и адрес следующего перехода вводятся в таблицу IP-маршрутизации. Как показано на рисунке, при изменении топологии сети маршрутизаторы будут автоматически корректироваться и пытаться найти новый оптимальный путь.



R1, R2 и R3 используют протокол динамической маршрутизации OSPF. При изменении топологии сети они могут автоматически корректироваться, чтобы найти новый оптимальный путь.

Обратите внимание на рисунке, что R2 подключен к Интернету. Поэтому администратор настраивает R1 со статическим маршрутом, отправляющим пакеты R2, если в таблице маршрутизации нет конкретной записи, соответствующей IP-адресу назначения. R1 и R2 также используют маршрутизацию OSPF для объявления напрямую подключенных сетей.



```
R1# show ip route
Codes: L - local, C - connected, S - static, 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
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      a - application route
      + - replicated route, % - next hop override, p - overrides from PfR
Gateway of last resort is 209.165.200.226 to network 0.0.0.0
S*  0.0.0.0/0 [1/0] via 209.165.200.226, GigabitEthernet0/0/1
    10.0.0.0/24 is subnetted, 1 subnets
O     10.1.1.0 [110/2] via 209.165.200.226, 00:02:45, GigabitEthernet0/0/1
    192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.10.0/24 is directly connected, GigabitEthernet0/0/0
L     192.168.10.1/32 is directly connected, GigabitEthernet0/0/0
    209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
C     209.165.200.224/30 is directly connected, GigabitEthernet0/0/1
L     209.165.200.225/32 is directly connected, GigabitEthernet0/0/1
R1#
```

Команда **show ip route** привилегированного режима EXEC используется для просмотра таблицы маршрутизации IPv4 на маршрутизаторе Cisco IOS. В примере показана таблица маршрутизации IPv4 маршрутизатора R1. В начале каждой записи таблицы маршрутизации находится код, который используется для идентификации типа маршрута или способа его изучения. К общим источникам маршрутов (кодам) относятся следующие:

- **L** - IP-адрес локального интерфейса с прямым подключением
- **C** – Присоединенная напрямую сеть
- **S** — Статический маршрут был вручную настроен администратором
- **O** - OSPF
- **D** - EIGRP

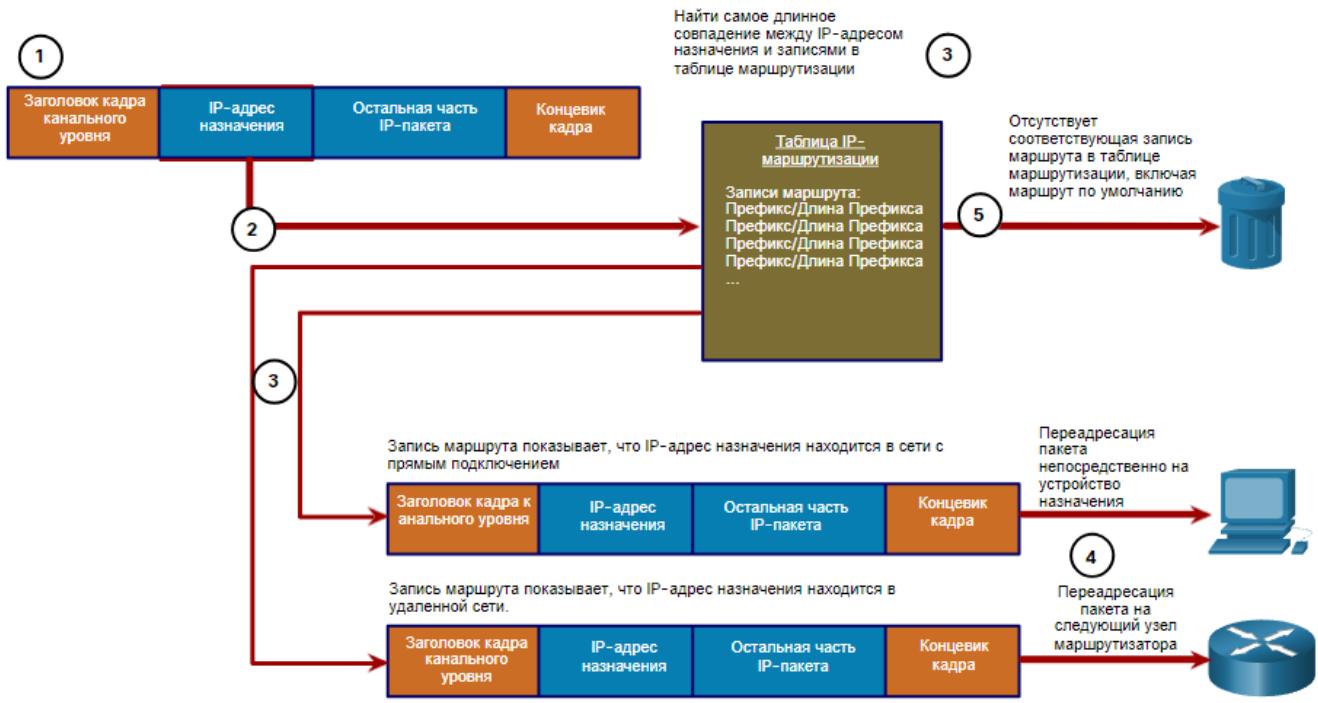
В таблице маршрутизации отображаются все известные маршруты назначения IPv4 для R1. Маршрут по умолчанию (default route) имеет сетевой адрес всех нулей. Например, сетевой адрес IPv4 — 0.0.0.0. Статическая запись маршрута в таблице маршрутизации начинается с кода **S***, как показано в примере.

e. Механизмы пересылки пакетов.

Процесс принятия решения о переадресации пакетов

Теперь, когда маршрутизатор определил наилучший путь для пакета, основанный на самом длинном совпадении, он должен определить, как инкапсулировать пакет и переслать его из правильного исходящего интерфейса.

На рисунке показано, как маршрутизатор сначала определяет оптимальный путь, а затем пересыпает пакет.



Следующие шаги описывают процесс пересылки пакетов, показанный на рисунке:

1. Кадр канального уровня с инкапсулированным IP-пакетом поступает на входной интерфейс.
 2. Маршрутизатор проверяет IP-адрес назначения в заголовке пакета и обращается к своей таблице IP-маршрутизации.
 3. Маршрутизатор находит самый длинный совпадающий префикс в таблице маршрутизации.
 4. Маршрутизатор инкапсулирует пакет во кадр канального уровня выходного интерфейса и пересыпает его из него.
- Назначением может быть устройство, подключенное к сети, или маршрутизатор следующего перехода.
5. Однако если нет соответствующей записи маршрута, пакет отбрасывается.

Переадресации пакета на устройство в сети с прямым подключением

Если запись маршрута указывает, что выходным интерфейсом является напрямую подключенная сеть, это означает, что конечный IP-адрес пакета принадлежит устройству непосредственно подключенной сети. Таким образом, пакет может быть перенаправлен непосредственно на устройство назначения. Конечное устройство обычно является конечным устройством в локальной сети Ethernet, что означает, что пакет должен быть инкапсулирован в кадр Ethernet.

Чтобы инкапсулировать пакет в кадр Ethernet, маршрутизатор должен определить MAC-адрес назначения, связанный с IP-адресом назначения пакета. Процесс зависит от того, является ли пакет пакетом IPv4 или IPv6.

- **Пакет IPv4** — маршрутизатор проверяет свою таблицу ARP на наличие адреса IPv4 назначения и связанного MAC-адреса Ethernet. Если совпадение отсутствует, маршрутизатор отправляет запрос ARP. Устройство назначения возвращает ответ ARP с MAC-адресом. Теперь маршрутизатор может пересыпать пакет IPv4 в кадр Ethernet с правильным MAC-адресом назначения.
- **Пакет IPv6** — маршрутизатор проверяет свой кэш соседей на наличие адреса IPv6 назначения и связанного MAC-адреса Ethernet. Если совпадение отсутствует, маршрутизатор отправляет сообщение ICMPv6 Neighbor Solicitation (NS). Устройство назначения возвращает сообщение ICMPv6 Neighbor Advertisement (NA) с MAC-адресом. Теперь маршрутизатор может пересыпать пакет IPv6 в кадр Ethernet с правильным MAC-адресом назначения.

Пересылка пакета на маршрутизатор следующего перехода.

Если запись маршрута указывает, что IP-адрес назначения находится в удаленной сети, это означает, что IP-адрес назначения пакета принадлежит устройству в сети, которое не подключено напрямую. Поэтому пакет должен быть перенаправлен другому маршрутизатору, в частности маршрутизатору следующего перехода. Адрес следующего перехода указывается в записи маршрута.

Если маршрутизатор пересылки и маршрутизатор следующего перехода находятся в сети Ethernet, аналогичный процесс (ARP и ICMPv6 Neighbor Discovery) будет происходить для определения MAC-адреса назначения пакета, как описано выше. Разница заключается в том, что маршрутизатор будет искать IP-адрес маршрутизатора следующего прыжка в таблице ARP или в соседнем кэше вместо IP-адреса назначения пакета.

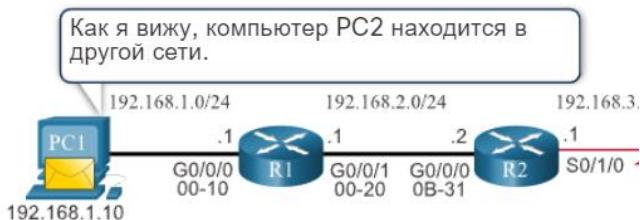
Примечание: Этот процесс будет отличаться для других типов сетей уровня 2.

Отбрасывает пакет - нет совпадения в таблице маршрутизации

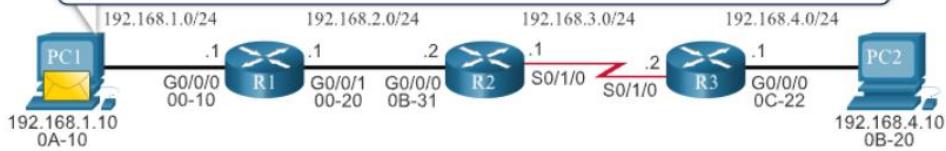
Если в таблице маршрутизации нет соответствия между IP-адресом назначения и префиксом, и если маршрут по умолчанию отсутствует, пакет будет отброшен.

PC1 отправляет пакет PC2

В первой анимации PC1 отправляет пакет PC2. Обратите внимание, что если запись ARP не существует в таблице ARP для шлюза по умолчанию 192.168.1.1, PC1 отправляет запрос ARP. Маршрутизатор R1 затем возвращает ответ ARP.



Поскольку компьютер PC2 находится в другой сети, я инкапсулирую пакет и отправлю его на маршрутизатор в МОЕЙ сети. Позвольте мне найти нужный MAC-адрес...



Кадр канала передачи данных

2-го уровня

Адрес MAC 00-10	MAC-адрес источника 0A-10	Введите 0x800	IP-адрес источника 192.168.1.10	Адрес IP 192.168.4.10	Поля IP-адреса	Данные	Концевик
192.168.1.10 0A-10	00-10	0x800	192.168.1.10	192.168.4.10			

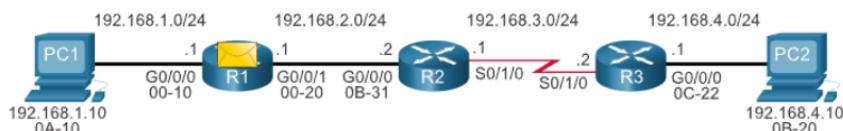
Данные пакета 3-го уровня

ARP-кэш компьютера PC1 для маршрутизатора R1

IP-адрес	MAC-адрес
192.168.1.1	00-10

Маршрутизатор R1 пересыпает пакет на компьютер PC2

R1 теперь пересыпает пакет на PC2. Поскольку выходной интерфейс находится в сети Ethernet, маршрутизатор R1 должен преобразовать IPv4-адрес следующего перехода в MAC-адрес места назначения с помощью протокола ARP: Если запись ARP не существует в таблице ARP для интерфейса следующего перехода 192.168.2.2, R1 отправляет запрос ARP. R2 возвращает ответ ARP.

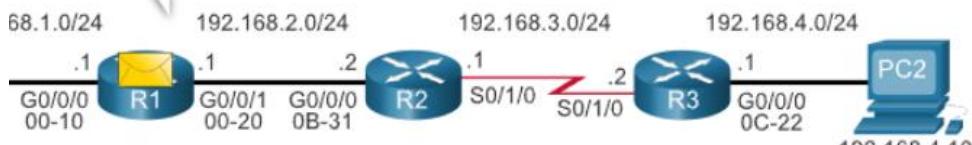


Кадр канала передачи данных 2-го уровня

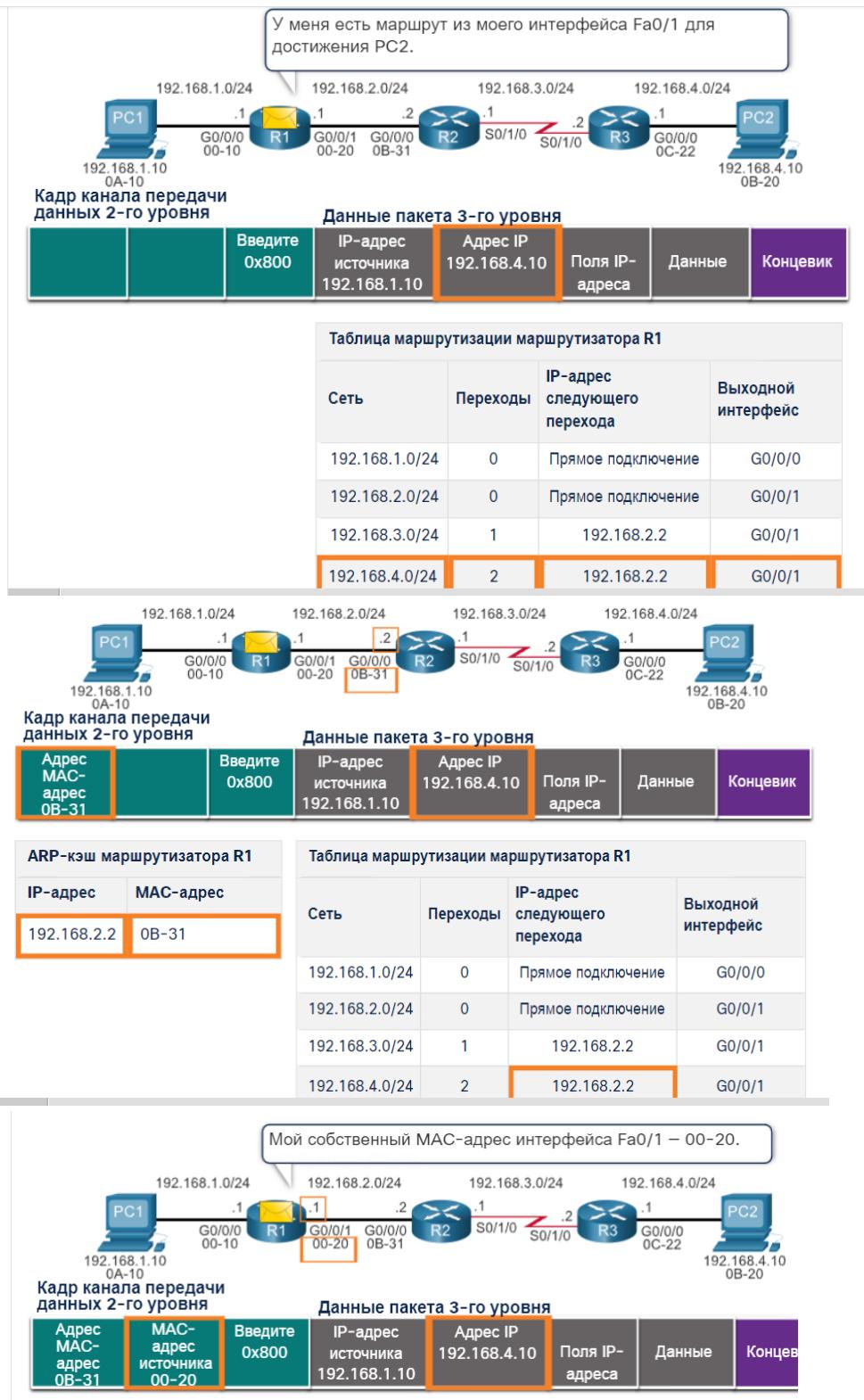
Данные пакета 3-го уровня

Адрес MAC 00-10	MAC-адрес источника 0A-10	Введите 0x800	IP-адрес источника 192.168.1.10	Адрес IP 192.168.4.10	Поля IP-адреса	Данные	Концевик
192.168.1.10 0A-10	00-10	0x800	192.168.1.10	192.168.4.10			

Кадр был отправлен мне от MAC-адреса 0A-10. Разрешите мне исследовать это подробнее.



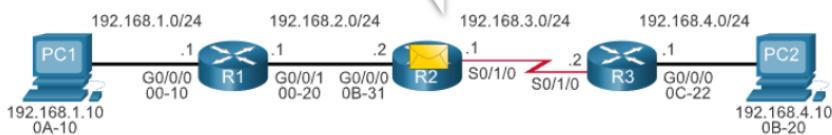
Судя по типу и назначению IP-адреса, пакет необходимо перенаправить.



Маршрутизатор R2 пересыпает пакет маршрутизатору R3

R2 теперь пересыпает пакет в R3. Поскольку выходной интерфейс не находится в сети Ethernet, маршрутизатор R2 не должен преобразовывать IPv4-адрес следующего перехода в MAC-адрес места назначения. Если интерфейс представляет собой последовательное одноранговое соединение (p2p), маршрутизатор инкапсулирует пакет IPv4 в соответствующий формат кадра канального уровня, используемый выходным интерфейсом (HDLC, PPP и т. д.). Поскольку на последовательных интерфейсах нет MAC-адресов, маршрутизатор R2 устанавливает канальный адрес назначения равным широковещательному адресу.

Пакет отправлен через последовательное соединение;
поэтому нужно использовать широковещательный адрес
назначения 2-го уровня.

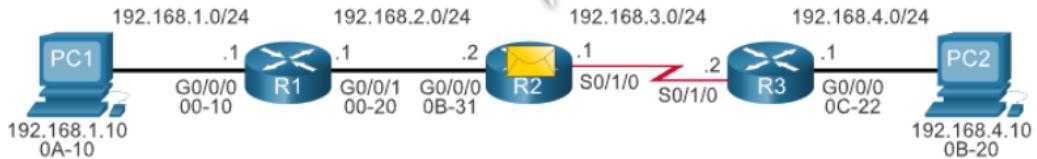


Адрес 0x8F			IP-адрес источника 192.168.1.10	Адрес IP 192.168.4.10	Поля IP-адреса	Данные	Концевик
------------	--	--	---------------------------------	-----------------------	----------------	--------	----------

Таблица маршрутизации коммутатора R3

Сеть	Переходы	IP-адрес следующего перехода	Выходной интерфейс
192.168.1.0/24	1	192.168.3.1	G0/0/0
192.168.2.0/24	0	Прямое подключение	G0/0/0
192.168.3.0/24	0	Прямое подключение	S0/0/0
192.168.4.0/24	1	192.168.3.2	S0/0/0

Это одноранговое последовательное соединение, поэтому адрес источника не нужен.



Адрес 0x8F	Управление 0x00	Введите 0x800	IP-адрес источника 192.168.1.10	Адрес IP 192.168.4.10	Поля IP-адреса	Данные	Конце
------------	-----------------	---------------	---------------------------------	-----------------------	----------------	--------	-------

Маршрутизатор R3 пересыпает пакет на компьютер PC2

R3 теперь пересыпает пакет на PC2. Поскольку IPv4-адрес назначения находится в сети Ethernet с прямым подключением, R3 должен разрешить IPv4-адрес назначения пакета с соответствующим MAC-адресом. Если в ARP-кэше нет записи, маршрутизатор R3 посылает ARP-запрос через свой интерфейс FastEthernet 0/0. PC2 отправляет ARP-ответ со своим MAC-адресом.

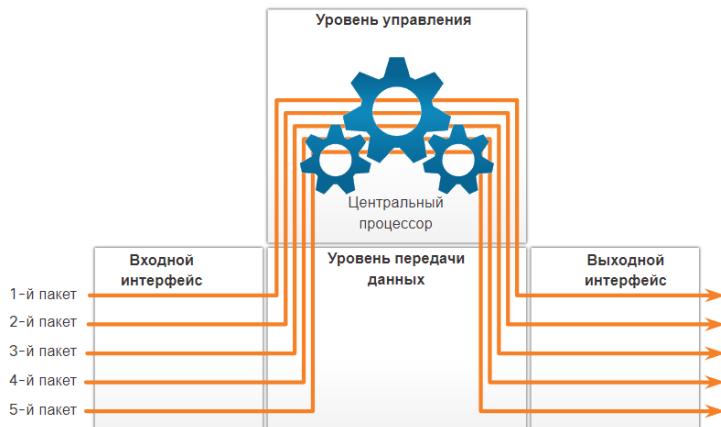
Как упоминалось ранее, основная ответственность функции пересылки пакетов заключается в инкапсуляции пакетов в соответствующий тип кадра канала передачи данных для исходящего интерфейса. Чем эффективнее маршрутизатор может выполнять эту задачу, тем быстрее пакеты будут пересыпаться маршрутизатором. Маршрутизаторы поддерживают три механизма пересылки пакетов.

- Процессорная коммутация (Process switching)
- Быстрая коммутация (Fast switching)
- Cisco Express Forwarding (CEF)

Процессорная коммутация

Устаревший механизм пересылки пакетов, все еще доступный на маршрутизаторах Cisco. Когда пакет прибывает на интерфейс, он пересыпается на уровень управления, где ЦП сопоставляет адрес назначения с записью в таблице маршрутизации, а затем определяет выходной интерфейс и пересыпает пакет. Важно понимать, что маршрутизатор совершают это с каждым пакетом, даже если целый поток пакетов предназначен для одного адреса назначения. Механизм процессорной коммутации работает очень медленно и редко

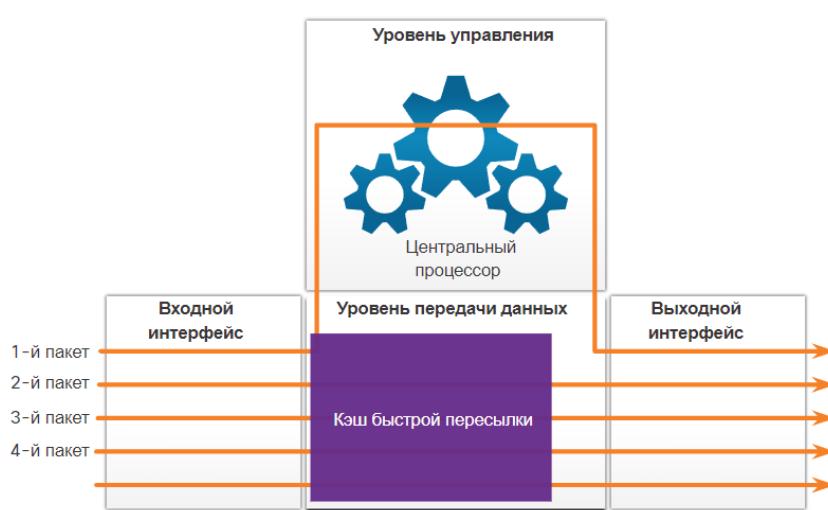
реализуется в современных сетях. Сравните данный механизм с механизмом быстрой коммутации.



Быстрая коммутация

Быстрая коммутация — это еще один, старый механизм переадресации пакетов, который был преемником процессорной коммутации. Быстрое переключение использует кэш быстрой коммутации для хранения информации следующего перехода. Когда пакет прибывает на интерфейс, он пересыпается на уровень управления, где ЦП ищет совпадение в кэше быстрой коммутации. Если совпадение не найдено, пакет проходит программную коммутацию и пересыпается на выходной интерфейс. Информация о трафике для пакетов также хранится в кэше быстрой коммутации. Если на интерфейс прибывает другой пакет, адресованный тому же назначению, то из кэш-памяти повторно используется информация о следующем переходе без вмешательства ЦП.

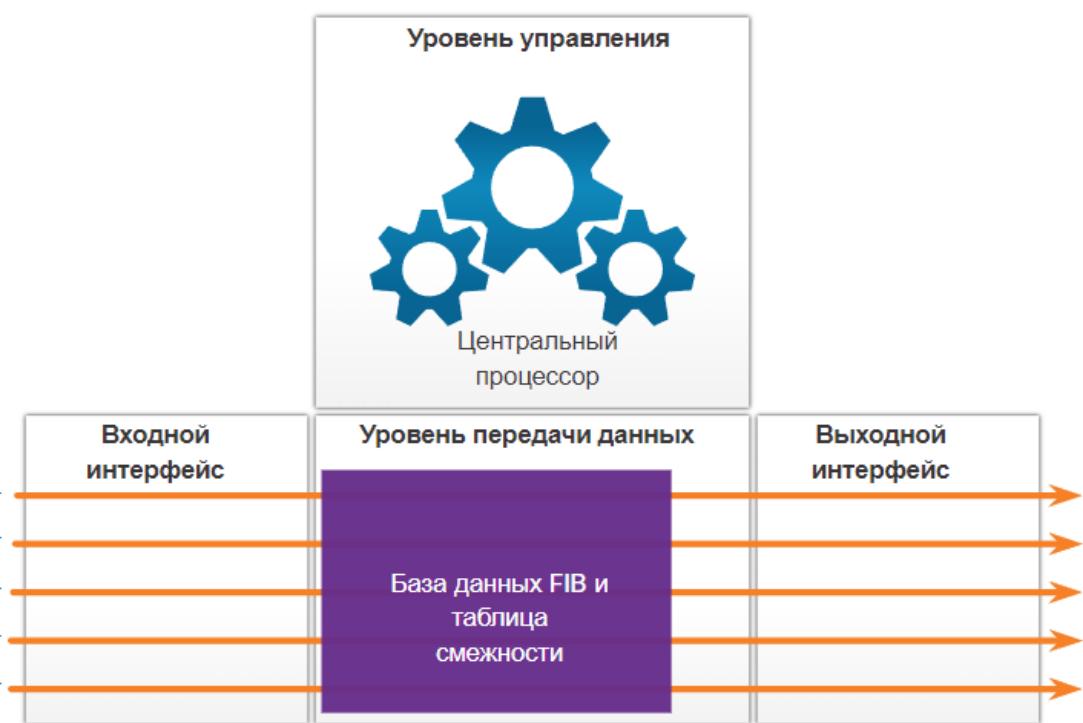
При быстрой коммутации только первый пакет потока проходит программную коммутацию, после чего он добавляется в кэш быстрой коммутации. Следующие четыре пакета быстро обрабатываются, исходя из информации в кэш-памяти.



Cisco Express Forwarding (CEF)

CEF является самым новым и используемым по умолчанию механизмом пересылки пакетов Cisco IOS. Как и быстрая коммутация, CEF создает 24-портовую базу данных переадресации (FIB) и таблицу смежности (adjacency table). Однако записи таблицы инициированы не пакетами, как при быстрой коммутации, а изменениями — например изменениями в сетевой топологии. Таким образом, по завершении сходимости сети в базе данных FIB и таблице смежности содержится вся необходимая информация, необходимая маршрутизатору при пересылке пакета. Коммутация CEF — это самый быстрый механизм пересылки, наиболее предпочтительный для использования на маршрутизаторах Cisco.

CEF формирует базу данных FIB и таблицу смежности после завершения сходимости сети. Все пять пакетов быстро обрабатываются на уровне данных.



Три механизма пересылки пакетов можно описать, проведя следующую аналогию:

- Программная коммутация делает все расчеты каждый раз, даже в случае решения идентичных проблем.
- Быстрая коммутация делает расчеты один раз, запоминая ответ для последующих идентичных случаев.
- Механизм CEF решает каждую из возможных проблем заранее, внося ее в электронную таблицу.

4. Маршрутизация:

a. Статическая маршрутизация. Типы статических маршрутов.

Статическая vs Динамическая

Функция	Динамическая маршрутизация	Статическая маршрутизация
Сложность конфигурирования	Не зависимость от размера сети	Повышается с увеличением размера сети
Изменения топологии	Изменяется автоматически в соответствии с изменениями топологии	Требуется участие администратора
Масштабируемость	Подходит для простых и сложных топологий	Подходит для простых топологий
Информационная безопасность	Безопасность должна быть настроена	Безопасность является неотъемлемым элементом
Потребление ресурсов	Использует центральный процессор, память, пропускную способность канала	Никаких дополнительных ресурсов не требуется
Предсказуемость пути	Маршрут зависит от используемой топологии и протокола маршрутизации	Явно определяется администратором

Статические маршруты можно настроить для IPv4 и IPv6. Оба протокола поддерживают следующие типы статических маршрутов:

- Стандартный статический маршрут – обычный маршрут, который сами записываем.
- Статический маршрут по умолчанию
- Плавающий статический маршрут
- суммарный статический маршрут – для объединения сетей

Статические маршруты настраиваются с помощью команд глобальной конфигурации **ip route** и **ipv6 route**.

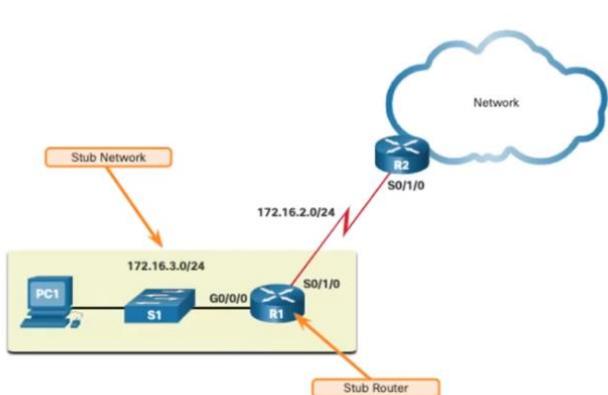
Статический маршрут по умолчанию

```
R(config)# ip route 0.0.0.0 0.0.0.0 {ip-address  
| exit-intf}
```

Плавающий статический маршрут

```
R(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.10 5
```

Статическая маршрутизация



Когда:

- Маленькие сети, которые не изменяются. Легко поддерживать таблицу маршрутизации
- Stub network, один маршрут
- Единственный маршрут по умолчанию (доступ ко всем сетям)

При настройке статического маршрута следующий переход может быть идентифицирован по IP-адресу, интерфейсу выхода или использовать оба варианта. В зависимости от того, как указан адрес назначения, создается один из трех следующих типов маршрута:

- **Маршрут следующего перехода** - Указывается только IP-адрес следующего перехода.

Настройка статических маршрутов

```
R1(config)# ip route 172.16.1.0 255.255.255.0 172.16.2.2  
R1(config)# ip route 192.168.1.0 255.255.255.0 172.16.2.2  
R1(config)# ip route 192.168.2.0 255.255.255.0 172.16.2.2
```

```
R1# show ip route | begin Gateway  
Gateway of last resort is not set  
    172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks  
S      172.16.1.0/24 [1/0] via 172.16.2.2  
C      172.16.2.0/24 is directly connected, Serial0/1/0  
L      172.16.2.1/32 is directly connected, Serial0/1/0  
C      172.16.3.0/24 is directly connected, GigabitEthernet0/0/0  
L      172.16.3.1/32 is directly connected, GigabitEthernet0/0/0  
S      192.168.1.0/24 [1/0] via 172.16.2.2  
S      192.168.2.0/24 [1/0] via 172.16.2.2
```

- **Напрямую подключенный статический маршрут.** Указывается только интерфейс выхода маршрутизатора.

```
R1(config)# ip route 172.16.1.0 255.255.255.0 s0/1/0
R1(config)# ip route 192.168.1.0 255.255.255.0 s0/1/0
R1(config)# ip route 192.168.2.0 255.255.255.0 s0/1/0
```

```
R1# show ip route | begin Gateway
Gateway of last resort is not set
    172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
S      172.16.1.0/24 is directly connected, Serial0/1/0
C      172.16.2.0/24 is directly connected, Serial0/1/0
L      172.16.2.1/32 is directly connected, Serial0/1/0
C      172.16.3.0/24 is directly connected, GigabitEthernet0/0/0
L      172.16.3.1/32 is directly connected, GigabitEthernet0/0/0
S      192.168.1.0/24 is directly connected, Serial0/1/0
S      192.168.2.0/24 is directly connected, Serial0/1/0
```

!!!Только для serial!!!

- **Полностью заданный статический маршрут.** Определены IP-адрес и интерфейс выхода следующего перехода.

```
R1(config)# ip route 172.16.1.0 255.255.255.0 GigabitEthernet 0/0/1 172.16.2.2
R1(config)# ip route 192.168.1.0 255.255.255.0 GigabitEthernet 0/0/1 172.16.2.2
R1(config)# ip route 192.168.2.0 255.255.255.0 GigabitEthernet 0/0/1 172.16.2.2
```

```
R1# show ip route | begin Gateway
Gateway of last resort is not set
    172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
S      172.16.1.0/24 [1/0] via 172.16.2.2, GigabitEthernet0/0/1
C      172.16.2.0/24 is directly connected, GigabitEthernet0/0/1
L      172.16.2.1/32 is directly connected, GigabitEthernet0/0/1
C      172.16.3.0/24 is directly connected, GigabitEthernet0/0/0
L      172.16.3.1/32 is directly connected, GigabitEthernet0/0/0
S      192.168.1.0/24 [1/0] via 172.16.2.2, GigabitEthernet0/0/1
S      192.168.2.0/24 [1/0] via 172.16.2.2, GigabitEthernet0/0/1
```

Настройка статических маршрутов IPv4 выполняется с помощью следующей команды:

```
Router(config)# ip route network-address subnet-mask { ip-address | exit-intf [ip-address]} [distance]
```

Примечание: Необходимо настроить параметры *ip-address*, *exit-intf*, или *ip-address* и *exit-intf*.

Параметр	Описание
<i>network-address</i>	Определяет сетевой адрес назначения IPv4 удаленной сети, чтобы добавить в таблицу маршрутизации.
<i>subnet-mask</i>	<ul style="list-style-type: none"> Определяет маску подсети удаленной сети. Маска подсети может быть изменена при объединении групп сетей и создает суммарный статический маршрут.
<i>ip-address</i>	<ul style="list-style-type: none"> Определяет адрес IPv4 маршрутизатора следующего прыжка. Обычно используется с широковещательными сетями (например, Ethernet). Может создать рекурсивный статический маршрут, где маршрутизатор выполняет дополнительный поиск, чтобы найти интерфейс выхода.
<i>exit-intf</i>	<ul style="list-style-type: none"> Определяет интерфейс выхода для пересылки пакетов. Статический маршрут с прямым подключением. Обычно используется при подключении к сети в конфигурации «точка-точка».
<i>exit-intf ip-address</i>	Создает полностью определенный статический маршрут, поскольку он указывает выход и адрес IPv4 следующего перехода.
<i>distance</i>	<ul style="list-style-type: none"> Опциональная команда, которая может использоваться для назначения административного расстояния значение от 1 до 255. Обычно используется для настройки плавающего статического маршрута, административное расстояние, которое выше, чем динамически изученный маршрут.

Команда статического маршрута IPv6



Настройка статических маршрутов IPv6 выполняется с помощью следующей команды:

```
Router(config)# ipv6 route ipv6-prefix/prefix-Length {ipv6-address | exit-intf [ipv6-address]} [distance]
```

Большинство параметров идентичны IPv4-версии этой команды.

Параметр	Описание
<i>ipv6-prefix</i>	Определяет сетевой адрес назначения IPv6 удаленной сети чтобы добавить в таблицу маршрутизации.
<i>/prefix-Length</i>	Определяет длину префикса удаленной сети.
<i>ipv6-address</i>	<ul style="list-style-type: none"> • Определяет IPv6-адрес маршрутизатора следующего перехода. • Обычно используется с широковещательными сетями (например, Ethernet) • Может создать рекурсивный статический маршрут, где маршрутизатор выполняет дополнительный поиск, чтобы найти интерфейс выхода.
<i>exit-intf</i>	<ul style="list-style-type: none"> • Определяет интерфейс выхода для пересылки пакетов. • Статический маршрут с прямым подключением. • Обычно используется при подключении к сети в конфигурации «точка-точка».
<i>exit-intf ipv6-address</i>	Создает полностью определенный статический маршрут, поскольку он указывает выходной интерфейс и адрес IPv6 следующего перехода.
<i>distance</i>	<ul style="list-style-type: none"> • Опциональная команда, которая может использоваться для назначения административного расстояния в значении от 1 до 255. • Обычно используется для настройки плавающего статического маршрута, административное расстояние, которое выше, чем динамически изученный маршрут.

Примечание: Для того чтобы маршрутизатор мог осуществлять пересылку пакетов для IPv6, необходимо настроить команду глобальной конфигурации **ipv6 unicast-routing**.

Статический маршрут IPv4 по умолчанию

Как показано на рисунке, синтаксис команды для статического маршрута по умолчанию аналогичен синтаксису команды для любого другого статического маршрута за исключением того, что адрес сети указывается как **0.0.0.0** а маска подсети – **0.0.0.0**. 0.0.0.0 0.0.0.0 в маршруте будет соответствовать любому сетевому адресу.

Примечание: Статический маршрут IPv4 по умолчанию обычно называют маршрутом с четырьмя нулями (quad-zero).

Синтаксис основной команды статического маршрутизатора по умолчанию следующий:

```
Router(config)# ip route 0.0.0.0 0.0.0.0 {ip-address | exit-intf}
```

Статический маршрут IPv6 по умолчанию

Синтаксис команд для статического маршрута по умолчанию похож на синтаксис команд для любого другого маршрута, за исключением того, что для параметра *ipv6-prefix/prefix-length* задано значение **::/0**, который совпадает со всеми маршрутами.

Основной синтаксис команды для статического маршрута по умолчанию для IPv6 выглядит следующим образом:

```
Router(config)# ipv6 route ::/0 {ipv6-address | exit-intf}
```

Плавающие статические маршруты — это статические маршруты, используемые для предоставления резервного пути основному статическому маршруту или динамическому маршруту на случай сбоя в работе канала. Плавающий статический маршрут используется только тогда, когда основной маршрут недоступен.

Для этой цели плавающий статический маршрут настраивается с более высоким значением административного расстояния, чем основной маршрут. Административное расстояние определяет надежность маршрута. При наличии нескольких путей к адресу назначения маршрутизатор выбирает путь с самым низким значением административного расстояния.

По умолчанию статические маршруты имеют значение административного расстояния, равное 1, поэтому они имеют приоритет перед маршрутами, полученными от протоколов динамической

маршрутизации. Например, для некоторых распространенных протоколов динамической маршрутизации используются следующие административные расстояния:

- EIGRP = 90
- OSPF = 110
- IS-IS = 115

Административную дистанцию статического маршрута можно увеличить и, таким образом, сделать этот маршрут менее приоритетным, чем другой статический маршрут или маршрут, полученный через протокол динамической маршрутизации. Таким образом, статический маршрут «плавает» и не используется в то время, когда маршрут с более коротким административным расстоянием работает. Однако, если предпочтительный маршрут потерян, плавающий статический маршрут может быть использован, и трафик будет направлен по этому альтернативному маршруту. Плавающие статические маршруты IP настраиваются с помощью **distance** аргумента для указания административного расстояния. Если значение административного расстояния не задано, используется значение по умолчанию (1).

Плавающий статический маршрут

```
R1# show ip route static | begin Gateway
Gateway of last resort is 172.16.2.2 to network 0.0.0.0
S*   0.0.0.0/0 [1/0] via 172.16.2.2
R1# show ipv6 route static | begin S :
S    ::/0 [1/0]
      via 2001:DB8:ACAD:2::2
R1#
```

```
R1(config)# ip route 0.0.0.0 0.0.0.0 172.16.2.2
R1(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.2 5
R1(config)# ipv6 route ::/0 2001:db8:acad:2::2
R1(config)# ipv6 route ::/0 2001:db8:feed:10::2 5
```

```
R1# show ip route static | begin Gateway
Gateway of last resort is 10.10.10.2 to network 0.0.0.0
S*   0.0.0.0/0 [5/0] via 10.10.10.2
R1# show ipv6 route static | begin :::
S    ::/0 [5/0]
      via 2001:DB8:FEED:10::2
R1#
```

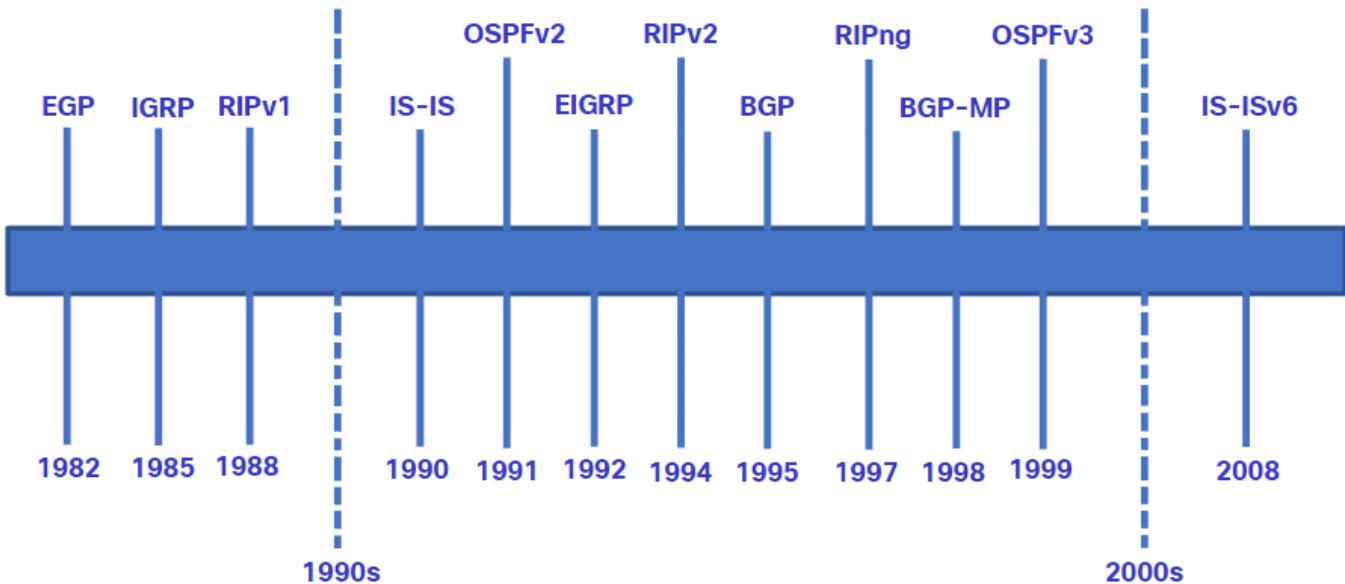
Поиск и устранение неисправностей

ping
traceroute
show ip route
show ip interface brief

б. Динамическая маршрутизация. Классификация протоколов маршрутизации.

Протокол RIP стал одним из первых протоколов маршрутизации. Первая версия (RIPv1) появилась в 1988 г

Наряду с развитием и усложнением сетей возникла необходимость в новых протоколах маршрутизации. Позднее протокол RIP был обновлен до версии RIPv2, которая лучше соответствовала потребностям новых крупных сетей того времени. Однако версия RIPv2 все же не отвечает масштабам современных сетевых решений. В соответствии с требованиями сетей большего размера были разработаны два усовершенствованных протокола маршрутизации: протокол маршрутизации «алгоритм кратчайшего пути» (OSPF) и протокол маршрутизации IS-IS. Компания Cisco разработала внутренний протокол маршрутизации шлюзов (IGRP) и усовершенствованный протокол IGRP (EIGRP), которые также обеспечивают хорошую масштабируемость при реализации сетей большего размера.



Для поддержки связи IPv6 были разработаны более новые версии протоколов IP-маршрутизации, как показано в строке IPv6 в таблице.

Таблица классифицирует текущие протоколы маршрутизации. Протоколы внутреннего шлюза (IGP) — это протоколы маршрутизации, используемые для обмена информацией о маршрутизации в домене маршрутизации, управляемом одной организацией. Существует только один EGP и это BGP. BGP используется для обмена информацией о маршрутизации между различными организациями, известными как автономные системы (AS). BGP используется провайдерами для маршрутизации пакетов через Интернет. Протоколы маршрутизации вектора расстояния, состояния канала и векторного пути относятся к типу алгоритма маршрутизации, используемого для определения наилучшего пути.

	Протоколы внутренней маршрутизации			Протоколы внешнего шлюза	
	Вектор расстояния	Состояние канала		Вектор пути	
IPv4	RIPv2 EIGRP	OSPFv2 IS-IS		BGP-4	
IPv6	RIPng EIGRP для IPv6	OSPFv3 IS-IS для IPv6		BGP-MP	

с. Задачи протоколов маршрутизации. Компоненты.

Протокол маршрутизации представляет собой набор процессов, алгоритмов и сообщений, используемых для обмена данными маршрутизации и наполнения таблицы маршрутизации оптимальными путями. Назначение протоколов динамической маршрутизации состоит в следующем:

- обнаружение удаленных сетей;
- обновление данных маршрутизации;
- выбор оптимального пути к сетям назначения;
- поиск нового оптимального пути в случае, если текущий путь недоступен.

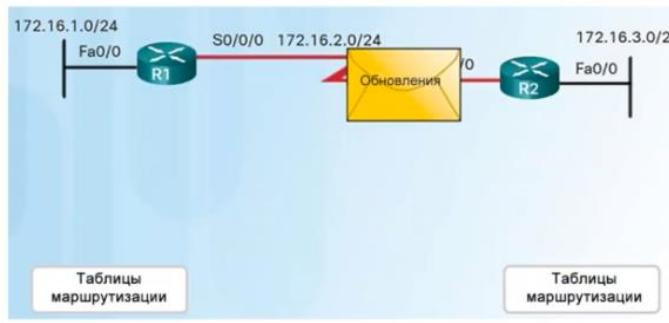
Протоколы динамической маршрутизации включают в себя следующие компоненты:

- **Структуры данных** — Протоколы маршрутизации обычно используют для своих операций таблицы или базы данных. Данная информация хранится в ОЗУ.

- **Сообщения протокола маршрутизации** — Протоколы маршрутизации используют различные типы сообщений для обнаружения соседних маршрутизаторов, обмена информацией о маршрутах и выполнения других задач, связанных с получением точной информации о сети.
- **Алгоритм** — алгоритм представляет собой определенный список действий, используемых для выполнения задачи. Протоколы маршрутизации используют алгоритмы, упрощающие обмен данных маршрутизации и определение оптимального пути.

При помощи протоколов маршрутизации маршрутизаторы динамически обмениваются информацией об удаленных сетях и автоматически сверяют эту информацию с собственными таблицами маршрутизации. Нажмите Воспроизвести, чтобы увидеть анимацию этого процесса. Протоколы маршрутизации определяют оптимальный путь или маршрут к каждой сети. Затем маршрут сверяется с таблицей маршрутизации. Этот маршрут будет добавлен в таблицу маршрутизации, если в таблице нет другого источника маршрутизации с меньшим административным расстоянием. Основным преимуществом протоколов динамической маршрутизации является то, что они обеспечивают обмен маршрутизирующей информацией между маршрутизаторами в случаях изменений в топологии. Подобный обмен данными позволяет маршрутизаторам автоматически получать информацию о новых сетях, а также находить альтернативные пути в случае сбоя канала к текущей сети.

d. Алгоритм работы протокола маршрутизации.



Работа протокола динамической маршрутизации :

- Маршрутизатор отправляет и принимает сообщения маршрутизации на свои интерфейсы
- Маршрутизатор предоставляет общий доступ к сообщениям маршрутизации и данным о маршрутах для других маршрутизаторов, использующих тот же протокол маршрутизации
- Маршрутизаторы осуществляют обмен данными маршрутизации для получения информации об удаленных сетях
- Когда маршрутизатор обнаруживает изменение топологии, протокол маршрутизации может объявить это изменение другим маршрутизаторам

Запуск после включения питания

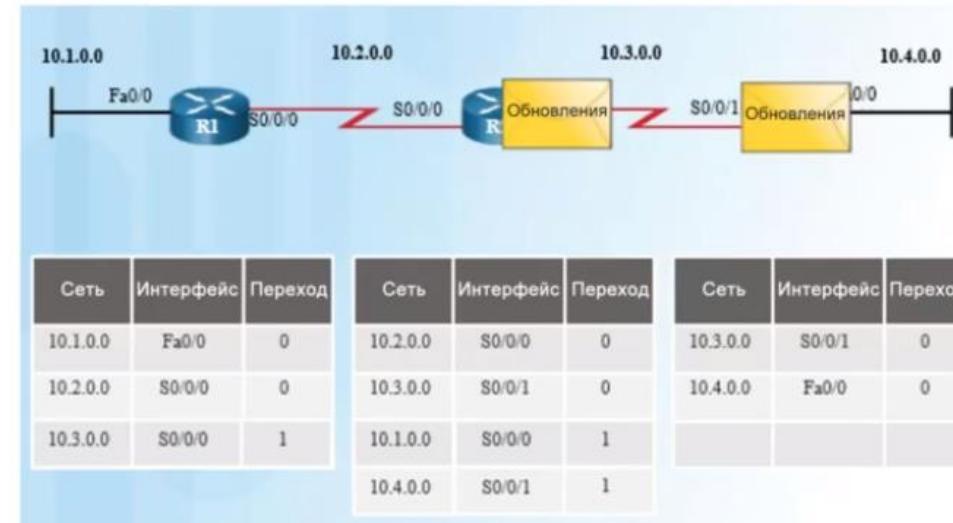
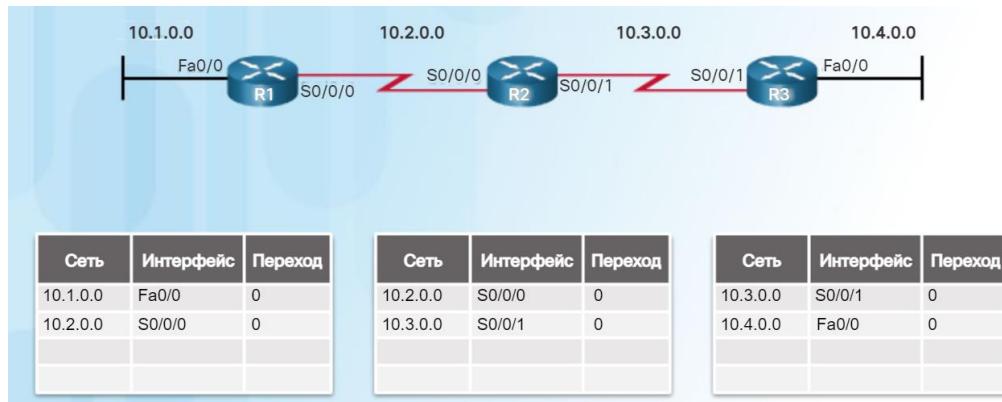
Все протоколы маршрутизации работают по одной схеме. В качестве примера рассмотрим следующий сценарий, в котором все три маршрутизатора работают по протоколу RIPv2.

При включении питания у маршрутизатора нет данных о топологии сети. Кроме того, у него нет данных о наличии устройств на другом конце каналов. Маршрутизатору доступна лишь информация из его собственного файла конфигурации, сохраненного в энергонезависимом ОЗУ (NVRAM). После успешной загрузки маршрутизатор применяет сохраненную конфигурацию. Если IP-адресация настроена верно, первоначально маршрутизатор выполняет обнаружение напрямую подключенных сетей.

Обратите внимание, как маршрутизаторы проходят процесс загрузки, а затем обнаруживают любые напрямую подключенные сети и маски подсети. Информация добавляется в таблицы маршрутизации следующим образом:

- R1 добавляет сеть 10.1.0.0, доступную через интерфейс FastEthernet 0/0, и сеть 10.2.0.0 становится доступной через интерфейс Serial 0/0/0.
- R2 добавляет сеть 10.2.0.0, доступную через интерфейс Serial 0/0/0, и сеть 10.3.0.0 становится доступной через интерфейс Serial 0/0/1.
- R3 добавляет сеть 10.3.0.0, доступную через интерфейс Serial 0/0/1, и сеть 10.4.0.0 становится доступной через интерфейс FastEthernet 0/0.

Имея эту исходную информацию, маршрутизаторы выполняют поиск дополнительных источников маршрутов для заполнения таблиц маршрутизации.



e. RIP: определение, описание, версии, принцип работы.

Протокол маршрутной информации

Протокол RIP — протокол маршрутизации первого поколения для среды IPv4, изначально указанный в RFC 1058. Настройка этого протокола достаточно проста, что делает его оптимальным протоколом для реализации в небольших сетях.

Протокол RIPv1 обладает следующими ключевыми характеристиками:

- Широковещательная рассылка обновлений маршрутизации (255.255.255.255) выполняется каждые 30 секунд.
- В качестве метрики для выбора пути служит число переходов.
- Число переходов, превышающее 15, считается бесконечным (т. е. слишком удаленным). Маршрутизатор 15-го перехода не передает обновление маршрутизации на следующий маршрутизатор.

В 1993 г. RIPv1 был обновлен до бесклассового протокола маршрутизации, известного как RIP версия 2 (RIPv2). RIPv2 включал в себя следующие улучшения:

- **Бесклассовый протокол маршрутизации.** Протокол поддерживает использование VLSM и CIDR, поскольку включает маску подсети в обновления маршрутизации.
- **Повышенная эффективность.** Протокол пересыпает обновления на групповой адрес 224.0.0.9, а не на адрес широковещательной рассылки 255.255.255.255.
- **Меньшее число записей маршрутизации.** Протокол поддерживает ручное объединение маршрутов на любом интерфейсе.
- **Безопасность.** Протокол поддерживает механизм аутентификации, что обеспечивает безопасность обновлений таблиц маршрутизации между соседними устройствами.

В таблице на рисунке представлено краткое описание различий протоколов RIPv1 и RIPv2.

Обновления протокола RIP инкапсулируются в сегмент протокола UDP, при этом номера портов источника и назначения настроены на порт UDP 520.

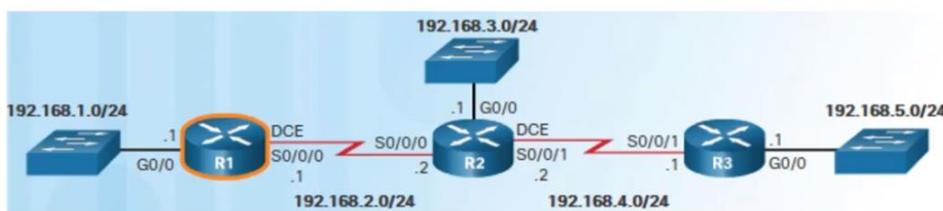
В 1997 году была представлена версия протокола RIP с поддержкой IPv6. В основе протокола RIPng лежит протокол RIPv2. В протоколе до сих пор действует ограничение в 15 переходов, а административная дистанция равна 120.

Сравнение протоколов RIPv1 и RIPv2

Характеристики и свойства	RIPv1	RIPv2
Метрика	Оба протокола используют в качестве простой метрики число переходов. Максимальное число переходов составляет 15.	
Обновления, направленные на адрес	255.255.255.255	224.0.0.9
Поддержка VLSM	✗	✓
Поддержка CIDR	✗	✓
Поддержка суммирования	✗	✓
Поддержка аутентификации	✗	✓

Команда router

```
Router (config)# router rip  
Router (config-router)# network ip_network
```



Пример

```
R1(config)# router rip
R1(config-router)# network 192.168.1.0
R1(config-router)# network 192.168.2.0
R1(config-router)#

```

Проверка show ip protocols

```
R1# show ip protocols
*** IP Routing is NSF aware ***
Routing Protocol is "rip"

```

Проверка show ip route

```
R1# show ip route | begin Gateway
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, Serial0/0/0
L        192.168.2.1/32 is directly connected, Serial0/0/0
R        192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:24, Serial0/0/0
R        192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:24, Serial0/0/0
R        192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:24, Serial0/0/0
R1#

```

Версии RIP

```
R(config)# router rip
```

```
R(config-router)# version 2
```

Версия 1 может получить сообщение версии 2, а версия 2 от версии 1 не может. Они не совместимы.

Пример

```
R1(config)# router rip
R1(config-router)# version 2
R1(config-router)# ^Z
R1#
R1# show ip protocols | section Default
  Default version control: send version 2, receive version 2
    Interface      Send   Recv   Triggered   RIP Key-chain
    GigabitEthernet0/0    2       2
    Serial0/0/0          2       2
R1#
```

Классы сетей

класс А	0	адрес сети (7 бит)	адрес хоста (24 бита)
класс В	10	адрес сети (14 бит)	адрес хоста (16 бит)
класс С	110	адрес сети (21 бит)	адрес хоста (8 бит)
класс D	1110	адрес многоадресной рассылки	
класс Е	1111 ^[1]	зарезервировано	

Класс определяется по первым битам.

Сеть – это классовая сеть. Все сети, которые не относятся к классовому разделению, являются подсетями.

Auto Summarization

```
Router (config)# router rip
Router (config-router)# no auto-summary
```

Чтобы RIPv2 распространяла информацию с маской подсети, надо выключить auto-summary(объединяет подсети по классовой маске).

Пример

```
R1(config)# router rip
R1(config-router)# no auto-summary
R1(config-router)# end
R1#
*Mar 10 14:11:49.659: %SYS-5-CONFIG_I: Configured from
console by console
R1# show ip protocols | section Automatic
  Automatic network summarization is not in effect
R1#
```

Passive Interfaces

```
R(config)# router rip  
R(config-router)# passive-interface interface-name
```

Или

```
R(config)# passive-interface default
```

Пример

```
R(config-if)#no passive-interface
```

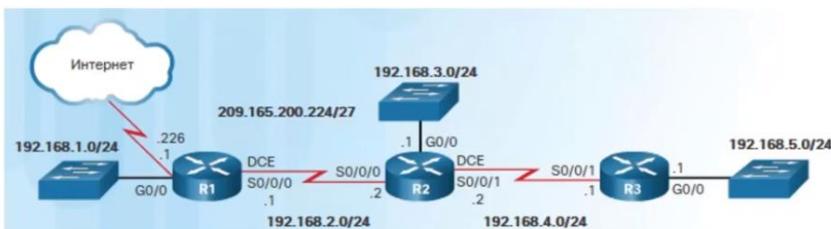
Интерфейсы в которые мы не хотим распространять служебные сообщения протоколов маршрутизации, но хотим рассказывать об этих сетях. Не хотим посылать в сеть где клиенты подключены служебные сообщения.

Пример

```
R1# show ip protocols | begin Default  
Default version control: send version 2, receive version 2  
Interface Send Recv Triggered RIP Key- chain  
Serial0/0/0 2 2  
Automatic network summarization is not in effect  
Maximum path: 4  
Routing for Networks:  
 192.168.1.0  
 192.168.2.0  
Passive Interface(s):  
 GigabitEthernet0/0  
Routing Information Sources:  
   Gateway          Distance      Last Update  
   192.168.2.2        120          00:00:06  
Distance: (default is 120)
```

Маршрут по умолчанию

```
R(config)# ip route 0.0.0.0 0.0.0.0  
R(config)# router rip  
R(config-router)# default-information originate
```



R1 граничный маршрутизатор, где запущен протокол маршрутизации. Можно настроить статический маршрут по умолчанию. И распространить всем маршрутизаторам информацию, что выход в сеть происходит через этот маршрутизатор. Команда default-information dhvte.

Пример

```
R1(config)# ip route 0.0.0.0 0.0.0.0 s0/0/1 209.165.200.226
R1(config)# router rip
R1(config-router)# default-information originate
R1(config-router)# ^Z
R1#
*Mar 10 23:33:51.801: %SYS-5-CONFIG_I: Configured from console by console
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.226 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 209.165.200.226, Serial0/0/1
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.1.0/24 is directly connected, GigabitEthernet0/0
L     192.168.1.1/32 is directly connected, GigabitEthernet0/0
    192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.2.0/24 is directly connected, Serial0/0/0
L     192.168.2.1/32 is directly connected, Serial0/0/0
R     192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:08, Serial0/0/0
R     192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:08, Serial0/0/0
R     192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:08, Serial0/0/0
    209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
C     209.165.200.0/24 is directly connected, Serial0/0/1
L     209.165.200.225/27 is directly connected, Serial0/0/1
R1#
```

f. Записи таблицы маршрутизации. Вид и устройство таблицы маршрутизации.

Записи таблицы маршрутизации

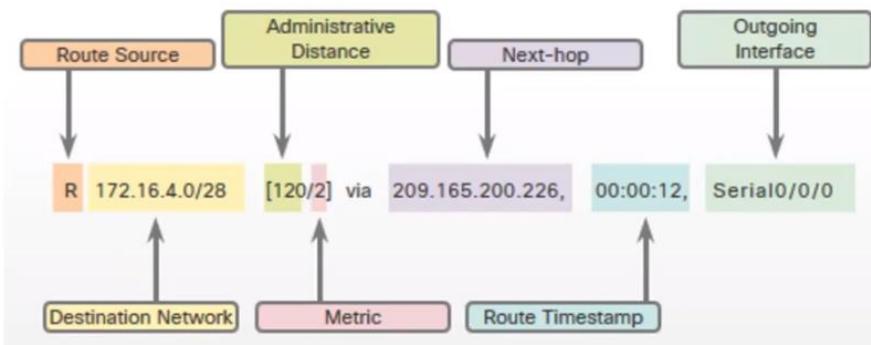
```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 209.165.200.234
    172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
C     172.16.1.0/24 is directly connected, GigabitEthernet0/0
L     172.16.1.1/32 is directly connected, GigabitEthernet0/0
R     172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:21, Serial0/0/0
R     172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:21, Serial0/0/0
R     172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:21, Serial0/0/0
R     192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:16, Serial0/0/0
    209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C     209.165.200.224/30 is directly connected, Serial0/0/0
L     209.165.200.225/32 is directly connected, Serial0/0/0
R     209.165.200.228/30 [120/1] via 209.165.200.226, 00:00:21, Serial0/0/0
C     209.165.200.232/30 is directly connected, Serial0/0/1
L     209.165.200.233/32 is directly connected, Serial0/0/1
R1#
```

Смотрим в cisco.

Записи таблицы маршрутизации

Route Source	Destination Network	Outgoing Interface
C	172.16.1.0/24 is directly connected,	GigabitEthernet0/0
L	172.16.1.1/32 is directly connected,	GigabitEthernet0/0



1. Источник маршрута(удаленная(static, rip) или напрямую подключенная (local, connected))
2. Сеть назначения
3. Степень доверия источнику маршрута. Чем меньше число, тем больше доверяют

4. Метрика (расстояние до сети). Чем меньше метрика, тем оптимальнее маршрут.
5. Адрес следующего маршрутизатора
6. Исходящий интерфейс

Administrative Distance

Route Source	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
Internal BGP	200

Directly connected попадают сразу, как только включился маршрутизатор. Статические маршруты тоже попадают, если интерфейс через который он ведет активен. Далее попадает наилучший динамический маршрут. Сначала смотрится administrative distance. Если для одной и той же сети, есть несколько путей изученных одним и тем же способом, тогда смотрим маршрут с наименьшей метрикой. Попадает только один маршрут, наилучший!!! Если совпала и метрика, тогда попадут обе записи и трафик будет балансируться.

Термины

```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 209.165.200.234
    172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
    C     172.16.1.0/24 is directly connected, GigabitEthernet0/0
    L     172.16.1.1/32 is directly connected, GigabitEthernet0/0
    R     172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:21, Serial0/0/0
    R     172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:21, Serial0/0/0
    R     172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:21, Serial0/0/0
    R     192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:16, Serial0/0/0
    209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
    C     209.165.200.224/30 is directly connected, Serial0/0/0
    L     209.165.200.225/32 is directly connected, Serial0/0/0
    R     209.165.200.228/30 [120/1] via 209.165.200.226, 00:00:21, Serial0/0/0
    C     209.165.200.232/30 is directly connected, Serial0/0/1
    L     209.165.200.233/32 is directly connected, Serial0/0/1
R1#
```

Маршруты:

- Ultimate route
- Level 1 route
- Level 1 parent route
- Level 2 child routes

Вертикальная иерархия. Смещение слева (неровные столбики)

1. Окончательные маршруты
2. Маршруты уровня 1
3. Родительский маршрут уровня 1
4. Дочерние маршруты второго уровня

Ultimate route (окончательный маршрут)

```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 209.165.200.234, Serial0/0/1
      is directly connected, Serial0/0/1
  172.16.0.0/16 is variably subnetted, 5 subnets, 3 masks
C   172.16.1.0/24 is directly connected, GigabitEthernet0/0
L   172.16.1.1/32 is directly connected, GigabitEthernet0/0
R   172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
R   172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R   172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R   192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:03, Serial0/0/0
      209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C   209.165.200.224/30 is directly connected, Serial0/0/0
L   209.165.200.225/32 is directly connected, Serial0/0/0
R   209.165.200.228/30 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
C   209.165.200.232/30 is directly connected, Serial0/0/1
L   209.165.200.233/32 is directly connected, Serial0/0/1
```

Строчка которой можно воспользоваться для отправки трафика. По ней сразу известно куда отправить трафик. Оранжевым цветом. Указан либо интерфейс, либо адрес следующего маршрутизатора, либо и то и другое.

Level 1 route

```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 209.165.200.234
      172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
C     172.16.1.0/24 is directly connected, GigabitEthernet0/0
L     172.16.1.1/32 is directly connected, GigabitEthernet0/0
R     172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:21, Serial0/0/0
R     172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:21, Serial0/0/0
R     172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:21, Serial0/0/0
R     192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:16, Serial0/0/0
      209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C     209.165.200.224/30 is directly connected, Serial0/0/0
L     209.165.200.225/32 is directly connected, Serial0/0/0
R     209.165.200.228/30 [120/1] via 209.165.200.226, 00:00:21, Serial0/0/0
C     209.165.200.232/30 is directly connected, Serial0/0/1
L     209.165.200.233/32 is directly connected, Serial0/0/1
```

Либо классовая сеть, либо объединение сетей, либо маршрут по умолчанию.

Level 1 parent route

```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 209.165.200.234
      172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
C     172.16.1.0/24 is directly connected, GigabitEthernet0/0
L     172.16.1.1/32 is directly connected, GigabitEthernet0/0
R     172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:21, Serial0/0/0
R     172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:21, Serial0/0/0
R     172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:21, Serial0/0/0
R     192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:16, Serial0/0/0
      209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C     209.165.200.224/30 is directly connected, Serial0/0/0
L     209.165.200.225/32 is directly connected, Serial0/0/0
R     209.165.200.228/30 [120/1] via 209.165.200.226, 00:00:21, Serial0/0/0
C     209.165.200.232/30 is directly connected, Serial0/0/1
L     209.165.200.233/32 is directly connected, Serial0/0/1
R1#
```

Классовая сеть, которая не указывает куда дальше отправиться трафику. Ей нельзя воспользоваться чтобы отправить трафик.

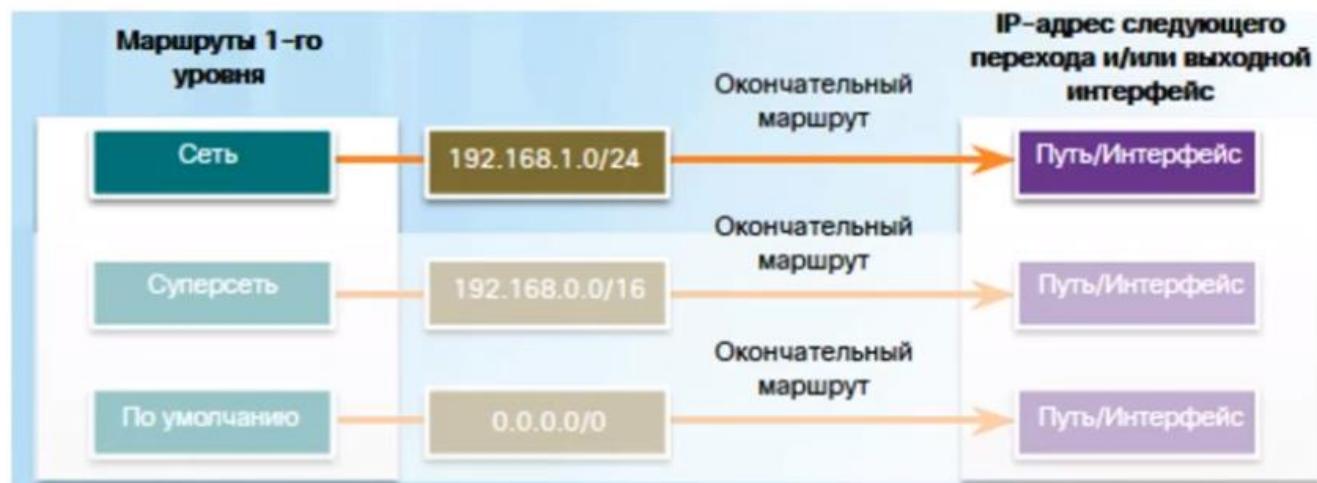
Level 2 child routes

```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

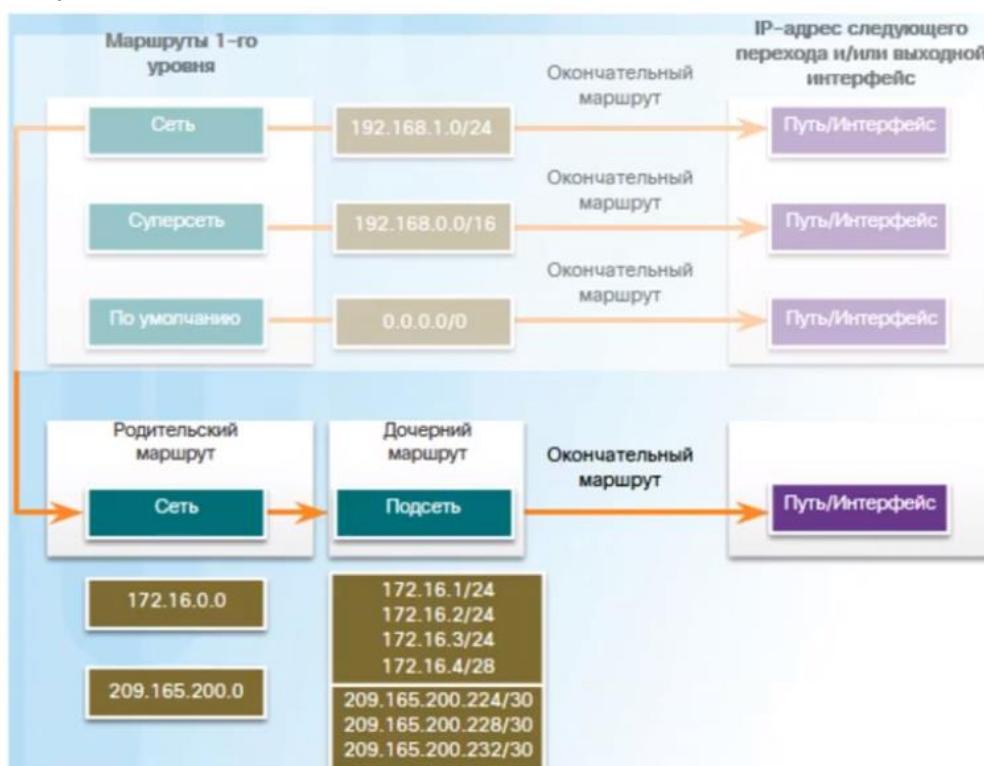
S*   0.0.0.0/0 [1/0] via 209.165.200.234
    172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
C     172.16.1.0/24 is directly connected, GigabitEthernet0/0
L     172.16.1.1/32 is directly connected, GigabitEthernet0/0
R     172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:21, Serial0/0/0
R     172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:21, Serial0/0/0
R     172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:21, Serial0/0/0
R     192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:16, Serial0/0/0
209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C     209.165.200.224/30 is directly connected, Serial0/0/0
L     209.165.200.225/32 is directly connected, Serial0/0/0
R     209.165.200.228/30 [120/1] via 209.165.200.226, 00:00:21, Serial0/0/0
C     209.165.200.232/30 is directly connected, Serial0/0/1
L     209.165.200.233/32 is directly connected, Serial0/0/1
R1#
```

Подсети родительских сетей. Являются окончательными.

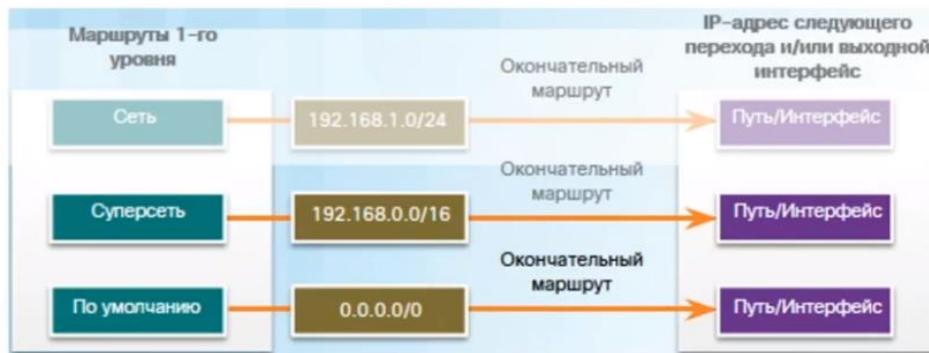
г. Процесс поиска маршрута по таблице маршрутизации.



Пришел пакет. Сперва просматриваются маршруты первого уровня, которые окончательные. С наилучшим совпадением по маске.



В первый уровень также попадают родительские маршруты. Если совпали с записью родительского маршрута (который не является окончательным), тогда смотрим по дочерним маршрутам.



Дальше переходим к следующей записи. Супerset – объединение сетей. Мaska меньше и совпадение требуется тое меньше. Если не совпали и нет маршрута по умолчанию, то отбрасываем пакет.

Плюс в том, что если нет совпадений по родительским маршрутам, то и дочерние смотреть не нужно. А это большой блок данных.

Статический маршрут лучше указывать полностью, иначе придется делать поиск по таблице маршрутизации 2 раза. Второй чтобы найти выходящий интерфейс.

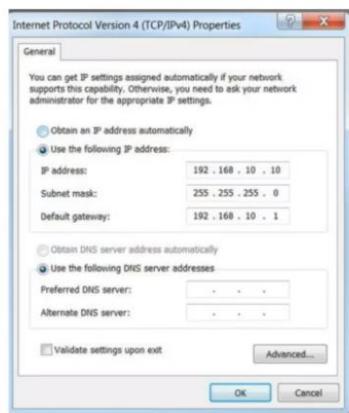
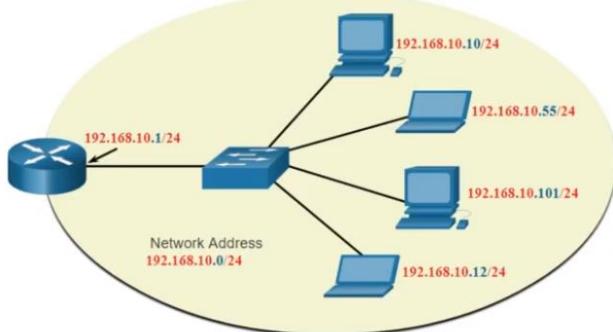
5. IP-адресация:

a. Определение IP-протокола. Его задачи.

IP-адресация

IP является одним из основных протоколов связи сетевого уровня

IP-адрес — уникальный идентификатор устройства



- **IPv4 адрес** - это уникальный IPv4 адрес хоста.
- **Маска подсети** - используется для определения сетевой части и части хоста адреса IPv4.

IP – основной протокол сетевого уровня.

Сетевой уровень, или третий уровень модели OSI, предоставляет сервисы, позволяющие оконечным устройствам обмениваться данными по сети. Как показано на рисунке, IP версии 4 (IPv4) и IP версии 6 (IPv6) являются протоколами связи основного сетевого уровня. Другие протоколы сетевого уровня включают протоколы маршрутизации, такие как Open Shortest Path First (OSPF), и протоколы обмена сообщениями, такие как Internet Control Message Protocol (ICMP).

Для выполнения сквозных коммуникаций через границы сети протоколы сетевого уровня выполняют четыре основные операции:

- **Адресация окончательных устройств** - Окончательным устройствам необходимо назначить уникальный IP-адрес для возможности их идентификации в сети.
- **Инкапсуляция** - Сетевой уровень получает единицу данных протокола (PDU) от транспортного уровня. Во время выполнения процесса, который называется инкапсуляцией, сетевой уровень добавляет информацию заголовка IP, например IP-адрес узла источника (отправляющего) и узла назначения (получающего). Процесс инкапсуляции выполняется источником IP-пакета.
- **Маршрутизация** - Сетевой уровень предоставляет сервисы, с помощью которых пакеты направляются к узлу назначения в другой сети. Для перемещения к другим сетям пакет должен быть обработан маршрутизатором. Роль маршрутизатора заключается в том, чтобы выбрать пути для пакетов и направить их к узлу назначения. Такой процесс называется маршрутизацией. До того как достигнуть узла назначения, пакет может пройти через несколько маршрутизаторов. Каждый маршрут на пути пакета к узлу назначения называется переходом.
- **Деинкапсуляция** - По прибытии пакета на сетевой уровень узла назначения этот узел проверяет IP-заголовок пакета. Если IP-адрес назначения в заголовке совпадает с его собственным IP-адресом, заголовок IP удаляется из пакета. После деинкапсуляции пакета, выполняемой сетевым узлом, полученная единица данных протокола (PDU) уровня 4 пересыпается соответствующей службе на транспортном уровне. Процесс деинкапсуляции выполняется конечным узлом IP-пакета.

В отличие от транспортного уровня (уровень 4 модели OSI), который управляет передачей данных между процессами, запущенными на каждом узле, протоколы сетевого уровня (IPv4 и IPv6) указывают структуру пакета и тип обработки, которые используются для перемещения данных от одного узла к другому. Функционирование без учета данных, передаваемых в каждом пакете, позволяет сетевому уровню передавать пакеты для нескольких типов коммуникации между несколькими узлами.

Протокол IP был разработан как протокол с низкой нагрузкой. Он обеспечивает только те функции, которые необходимы для доставки пакета от узла источника к узлу назначения по взаимосвязанной системе сетей. Этот протокол не предназначен для мониторинга и управления потоком пакетов. Эти функции, при необходимости, выполняются другими протоколами на других уровнях, в первую очередь — протоколом TCP на уровне 4.

Основные характеристики IP:

- **Без установления соединения** - означает, что перед отправкой пакетов данных соединение с хостом назначения не устанавливается.
- **Негарантированная доставка** - показывает, что IP-протокол по своей сути неустойчив, так как доставка пакетов не гарантируется.
- **Независимость от среды** - работа не зависит от средства подключения (médный, оптоволоконный кабель или беспроводная среда).

На протяжении многих лет разрабатывались дополнительные протоколы и процессы для решения новых задач. Тем не менее даже в результате изменений IPv4 по-прежнему имеет три основных недостатка.

- **Недостаток IP-адресов.** - IPv4 может предложить лишь ограниченное количество уникальных публичных IPv4-адресов. Несмотря на то что существует примерно 4 миллиарда IPv4-адресов, возросшее число новых устройств, в которых используется протокол IP, а также потенциальный рост менее развитых регионов привели к необходимости дополнительного увеличения количества адресов.
- **Недостаток сквозных соединений.** - Преобразование сетевых адресов (NAT) представляет собой технологию, которая обычно применяется в сетях IPv4. NAT позволяет различным устройствам совместно использовать один публичный IPv4-адрес. При этом, поскольку публичный IPv4-адрес используется совместно, IPv4-адрес узла внутренней сети скрыт. Это может представлять проблему при использовании технологий, для которых необходимы сквозные соединения.
- **Повышенная сложность сети** — несмотря на то, что NAT продлил срок службы IPv4, он был предназначен только как механизм перехода на IPv6. NAT в своей разнообразной реализации создает дополнительную сложность в сети, создавая задержку и затрудняя поиск и устранение неисправностей.

Зачем IPv6?



Необходимость IPv6:

- Адреса IPv4 заканчиваются: IPv6 является преемником IPv4 и имеет более крупное 128-битное адресное пространство
- Исправление ограничений IPv4 и другие улучшения
- Зависимость от NAT, что нарушает связность сети

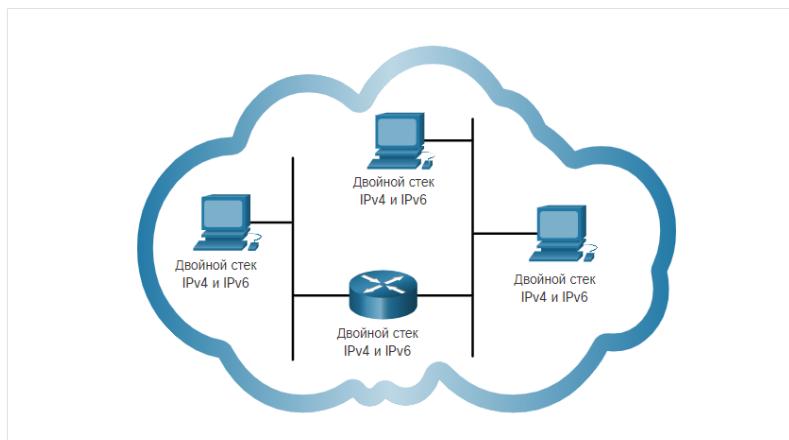
Со существование IPv4 и IPv6:

- **Двойной стек** (работа с IPv4 и IPv6 одновременно)
- **Туннелирование** (передача пакета IPv4 в заголовке IPv6)
- **Трансляция** (NAT64)

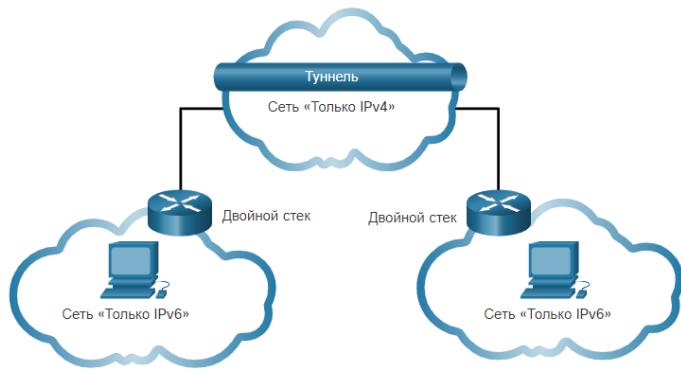
К улучшениям, которые предлагает протокол IPv6, относятся следующие.

- **Расширенное адресное пространство.** IPv6-адреса используют 128-битную иерархическую адресацию, в отличие от протокола IPv4, использующего 32 бита.
- **Улучшенная обработка пакетов.** Структура заголовка IPv6 была упрощена благодаря уменьшению количества полей.
- **Отсутствие необходимости в использовании NAT.** Благодаря большому количеству публичных IPv6-адресов нет необходимости в преобразовании сетевых адресов (NAT) между частными и публичными адресами IPv4. Это позволяет избежать некоторых проблем, связанных с NAT, с которыми сталкиваются приложения, требующие сквозного подключения.

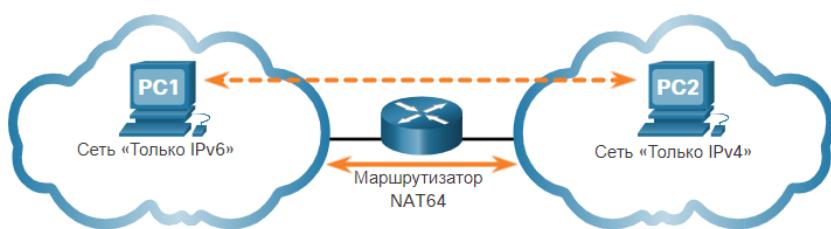
Протоколы IPv4 и IPv6 можно одновременно использовать в одной сети посредством двойного стека. Устройства с двойным стеком одновременно работают с протокольными стеками IPv4 и IPv6. Известный как собственный IPv6, это означает, что сеть клиента имеет подключение IPv6 к своему Интернет-провайдеру и может получать доступ к контенту, найденному в Интернете, через IPv6.

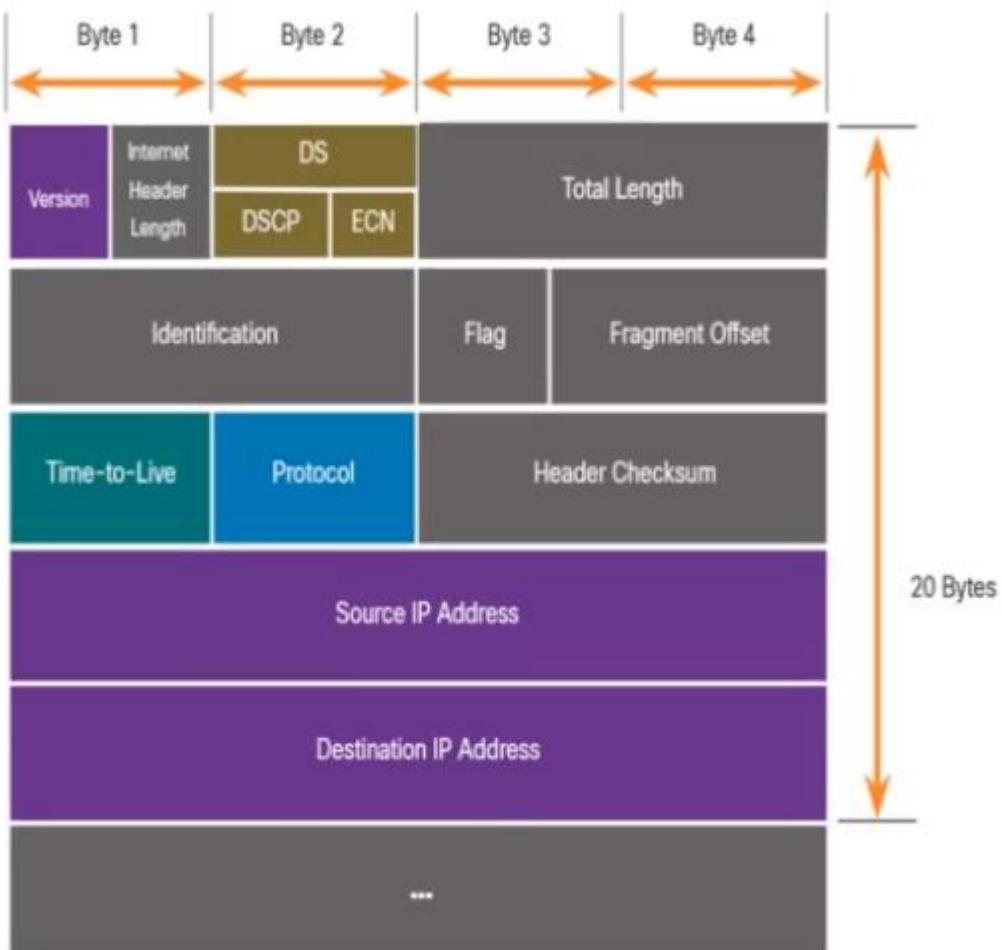


Туннелирование – это способ передачи IPv6-пакета по IPv4-сети. IPv6-пакет инкапсулируется внутри IPv4-пакета, как и другие типы данных.



Преобразование сетевых адресов 64 (NAT64) позволяет устройствам под управлением IPv6 обмениваться данными с устройствами под управлением IPv4 посредством способа преобразования, аналогичного NAT для IPv4. Пакет IPv6 преобразуется в пакет IPv4, а пакет IPv4 преобразуется в пакет IPv6.



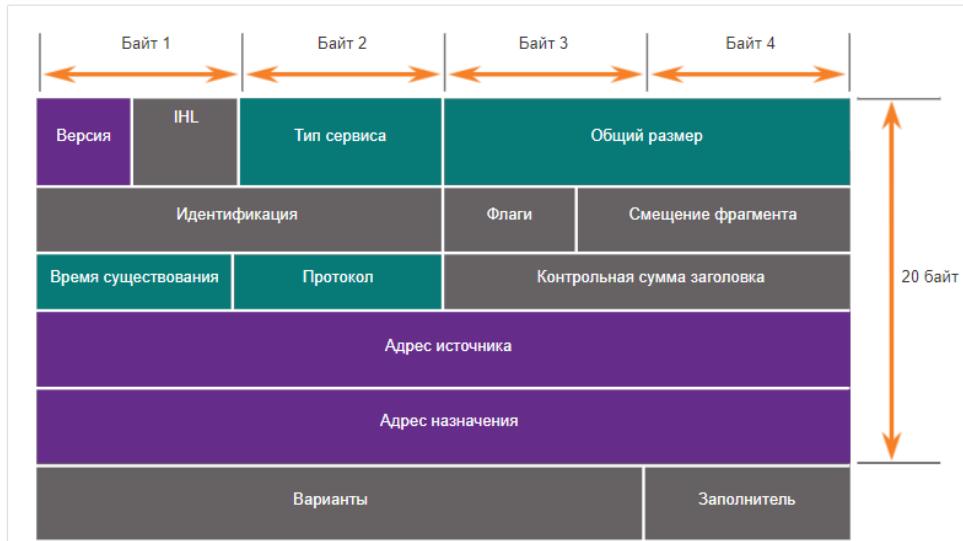


IPv4 Packet

Среди наиболее важных полей в заголовке IPv4 можно выделить следующие:

- **Версия.** – Содержит 4-битное двоичное значение, определяющее версию IP-пакета. Для пакетов IPv4 в этом поле всегда указано значение 0100.
- **Дифференцированные сервисы (Differentiated Services, DS).** – Поле, которое ранее называлось «Тип сервиса» (Type of Service, ToS). DS – это 8-битное поле, используемое для определения приоритета каждого пакета. 6 наиболее важных битов поля дифференцированных услуг (DSCP) и последние 2 бита – это биты явного уведомления о заторах (ECN).
- **Контрольная сумма заголовка** – используется для обнаружения повреждения в заголовке IPv4.
- **Время существования (Time-to-Live, TTL).** Содержит 8-битное двоичное значение, используемое для ограничения времени существования пакета. Отправитель пакета устанавливает начальное значение времени существования (TTL), которое уменьшается на единицу каждый раз при обработке пакета маршрутизатором. Если значение в поле TTL уменьшается до нуля, маршрутизатор отбрасывает пакет и отправляет на IP-адрес источника сообщение о превышении времени протокола ICMP (управление сообщениями в сети). Поскольку маршрутизатор уменьшает TTL каждого пакета, маршрутизатор также должен пересчитать контрольную сумму заголовка.
- **Протокол.** – Это поле используется для определения протокола следующего уровня. Это 8-битное двоичное значение, указывающее тип полезной нагрузки данных, которые переносит пакет, что позволяет сетевому уровню пересыпать данные на соответствующий протокол более высокого уровня. Обычно используются значения ICMP (1), TCP (6) и UDP (17).
- **IPv4-адрес источника.** – Содержит 32-битное двоичное значение, которое представляет IPv4-адрес источника пакета. IPv4-адрес источника – это всегда индивидуальный адрес.
- **IPv4-адрес назначения.** – Содержит 32-битное двоичное значение, которое представляет IPv4-адрес назначения пакета. IPv4-адрес назначения – одноадресная рассылка, многоадресная рассылка, или широковещательный адрес.

Заголовок пакета IPv4



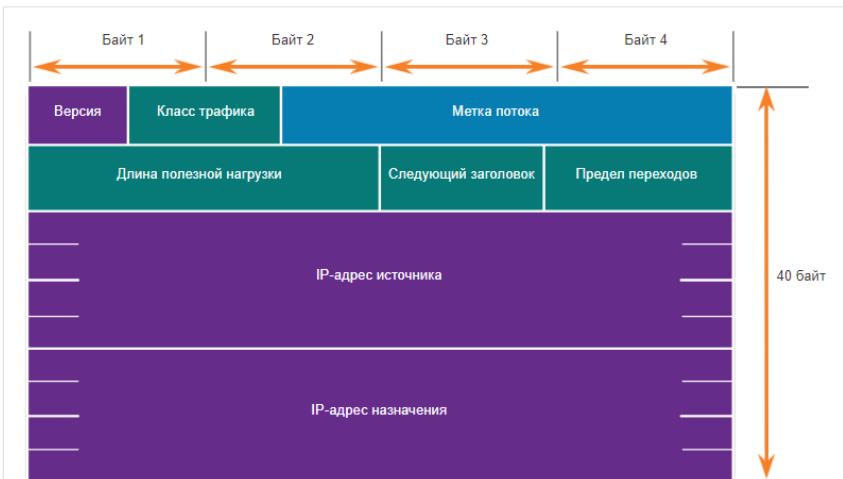
Условные обозначения

- Имя полей перешедшее от IPv4 к IPv6
- Имя и позиция, измененные в IPv6
- Новое поле в IPv6

Время жизни сообщения сокращается на каждом маршрутизаторе. Поэтому на каждом шаге надо пересчитывать чек сумму. Пропали флаги, смещения фрагмента. Использовались для фрагментации. Теперь этого нет, это экономит ресурсы.

В IPv6 есть шифрование. Ссылается в поле next header.

Заголовок пакета IPv6



Условные обозначения

Поля в заголовке пакета IPv6:

- Версия.** - Это поле содержит 4-битное двоичное значение, которое определяет версию IP-пакета. Для пакетов IPv6 в этом поле всегда указано значение 0110.
- Класс трафика.** - Это 8-битное поле, соответствующее полю «Дифференцированные услуги (DS)» в заголовке IPv4.
- Метка потока.** - Это 20-битное поле указывает на то, что всем пакетам с одинаковыми метками потока назначается одинаковый тип обработки маршрутизаторами.
- Длина полезной нагрузки.** - Это 16-битное поле указывает длину блока данных или полезной нагрузки пакета IPv6. Это не включает длину заголовка IPv6, который является фиксированным 40-байтным заголовком.
- Следующий заголовок.** - Это 8-битное поле, соответствующее полю «Протокол» в заголовке IPv4. Оно указывает тип полезной нагрузки данных, которые переносит пакет, что позволяет сетевому уровню пересыпал данные на соответствующий протокол более высокого уровня.
- Предел перехода.** - Это 8-битное поле, заменяющее поле «Время существования» (TTL) в IPv4. Это значение уменьшается на единицу каждым маршрутизатором, пересылающим пакет. Когда счетчик достигает значения 0, пакет отбрасывается и на отправляющий узел пересыпается сообщение ICMPv6, которое означает, что пакет не достиг своего назначения, так как был превышен предел переходов. В отличие от IPv4, IPv6 не включает контрольную сумму заголовка IPv6, так как эта функция выполняется как на нижнем, так и на верхнем уровнях. Это означает, что контрольную сумму не нужно пересчитывать каждым маршрутизатором при уменьшении поля Hop Limit, что также повышает производительность сети.
- IPv6-адрес источника.** - Это 128-битное поле, определяющее IPv6-адрес хоста-отправителя.
- IPv6-адрес назначения.** - Это 128-битное поле, определяющее IPv6-адрес хоста-получателя.

b. Версии IP. Формат адресов.

Адрес IPv4 является иерархическим и состоит из раздела сети и раздела хоста. Определяя ту или иную часть, необходимо обращать внимание не на десятичное значение, а на 32-битный поток, как показано на рисунке.

IPv4-адрес



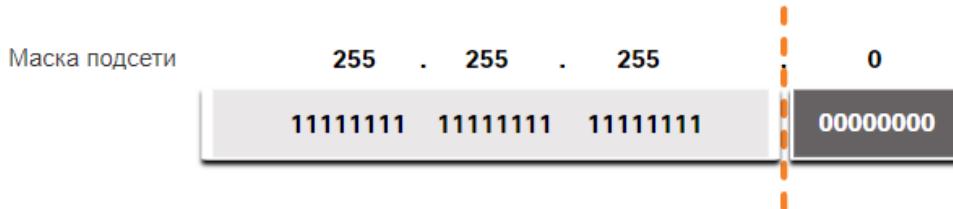
Биты в сетевой части адреса должны быть одинаковыми у всех устройств, находящихся в одной сети. Биты в хостовой части адреса должны быть уникальными для каждого хоста в сети. Если два узла имеют одну битовую комбинацию в определенной сетевой части 32-битного потока, то эти два узла находятся в одной и той же сети.

Как показано на рисунке, для назначения адреса IPv4 узлу требуется следующее:

- **IPv4 адрес** - это уникальный IPv4 адрес хоста.
- **Маска подсети** - используется для определения сетевой части и части хоста адреса IPv4.

Маска подсети IPv4 — это 32-битовое значение, которое отделяет сетевую часть адреса от хостовой части. При назначении устройству IPv4-адреса для определения адреса сети, к которому относится данное устройство, используется маска подсети. Сетевой адрес представляет все устройства в одной сети.

Маска подсети



Обратите внимание, что маска подсети представляет собой последовательную последовательность из единичных битов (1), за которой следует последовательная последовательность из нулевых битов (0).

Для идентификации сетевой и узловой части IPv4-адреса маска подсети побитово сравнивается с IPv4-адресом слева направо, как показано на рисунке.

Связь адреса IPv4 с маской подсети

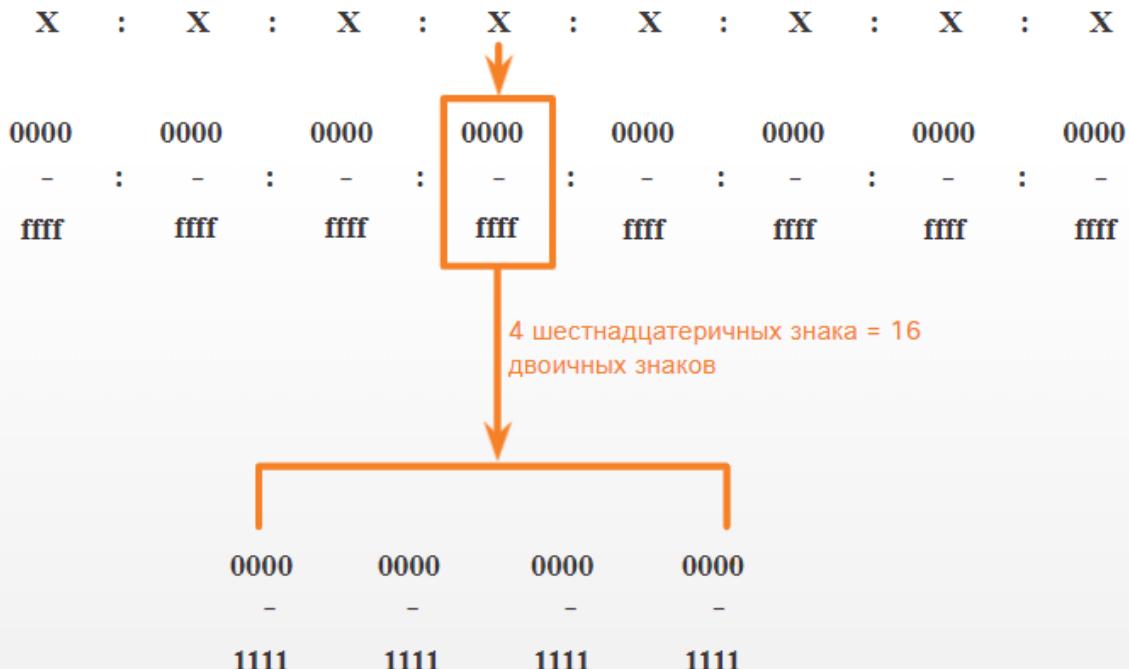


Обратите внимание, что маска подсети на самом деле не содержит сетевой или узловой части IPv4-адреса; она лишь указывает компьютеру, где искать эти части в конкретном IPv4-адресе.

Сам процесс, используемый для определения сетевой и узловой частей адреса, называется логической операцией И (AND).

Длина IPv6-адресов составляет 128 бит, написанных в виде строки шестнадцатеричных значений. Каждые 4 бита представлены одной шестнадцатеричной цифрой, причем общее количество шестнадцатеричных значений равно 32, как показано на рисунке. IPv6-адреса нечувствительны к регистру, их можно записывать как строчными, так и прописными буквами.

16-битные сегменты или гекстеты



Предпочтительный формат

Как показано на рисунке, формат записи адреса IPv6 — x:x:x:x:x:x:x, где каждый x состоит из четырех шестнадцатеричных значений. Термин октет относится к восьми битам адреса IPv4. В IPv6-адресах сегмент из 16 бит или четырех шестнадцатеричных цифр неофициально называют гекстетом. Каждый x — это 1 гекстет, 16 бит или 4 шестнадцатеричные цифры.

Предпочтительный формат означает, что IPv6-адрес записан с помощью 32 шестнадцатеричных цифр. Тем не менее, это не самый оптимальный способ представления IPv6-адреса. Существуют два правила, которые помогают уменьшить количество цифр, необходимых для представления адреса IPv6.

Первое правило для сокращения записи IPv6-адресов — пропуск всех начальных 0 (нулей) в шестнадцатеричной записи. Вот четыре примера способов опустить ведущие нули:

- 01AB можно представить как 1AB
- 09f0 может быть представлен как 9f0

Это правило применяется только к начальным нулям, а НЕ к конечным, иначе адрес будет непонятен. Например, гекстет «ABC» может быть представлен как «0ABC», либо как «ABC0» (а это разные значения).

Второе правило для сокращения записи адресов IPv6 заключается в том, что двойное двоеточие (::) может заменить любую единую, смежную строку одного или нескольких 16-битных сегментов (гекстетов), состоящих из нулей. Например, 2001:db8:cafe:1:0:0:0:1 (ведущие 0s опущены) можно представить как 2001:db8:cafe:1::1. Двойное двоеточие (::) используется вместо трех гекстетов все-0 (0:0:0).

2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d /64

- Префикс

2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d /64****

- Префикс

- Адрес интерфейса

2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d /64

- Префикс

- Адрес интерфейса

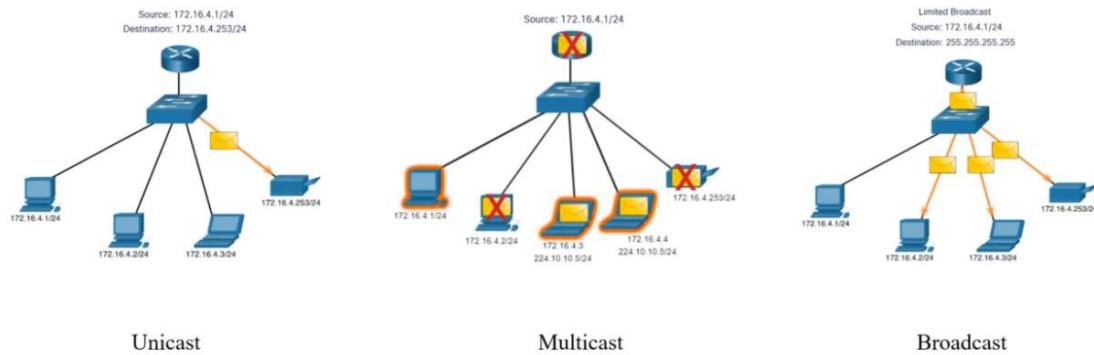
- Длина префикса

Адрес: 2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d /64

Префикс: **2001:0db8:11a3:09d7:0000:0000:0000:0000**

с. Типы рассылки IPv4 и IPv6.

Типы рассылки IPv4



Одноадресная передача относится к одному устройству, отправляющему сообщение другому устройству в режиме один-в-один.

Одноадресный пакет имеет IP-адрес назначения, который является одноадресный адрес, который передается одному получателю. Исходный IP-адрес может быть только одноадресный адрес, так как пакет может быть получен только из одного источника. Это независимо от того, является ли конечный IP-адрес одноадресным, широковещательным или многоадресным.

Широковещательная рассылка связана с устройством, отправляющее сообщение всем остальным устройствам в сети в режиме «один-ко всем».

Пакет широковещательной рассылки содержит IPv4-адрес назначения, в узловой части которого присутствуют только единицы (1).

Ограниченнная широковещательная рассылка

Источник: 172.16.4.1/24

Получатель 255.255.255.255

Направленная широковещательная рассылка

В дополнение к широковещательному адресу 255.255.255.255 для каждой сети имеется широковещательный IPv4 адрес. Пакет широковещательной рассылки содержит IPv4-адрес назначения, в узловой части которого присутствуют только единицы (1).. Например, направленный широковещательный адрес для 192.168.1.0/24 — 192.168.1.255. Этот адрес позволяет осуществлять связь со всеми узлами в этой сети. Чтобы отправить данные всем узлам в сети, узел может отправить один пакет, адресован широковещательному адресу сети.

Многоадресная рассылка уменьшает трафик, позволяя узлу отправлять один пакет выбранной группе узлов, которые подписаны на группу многоадресной рассылки.

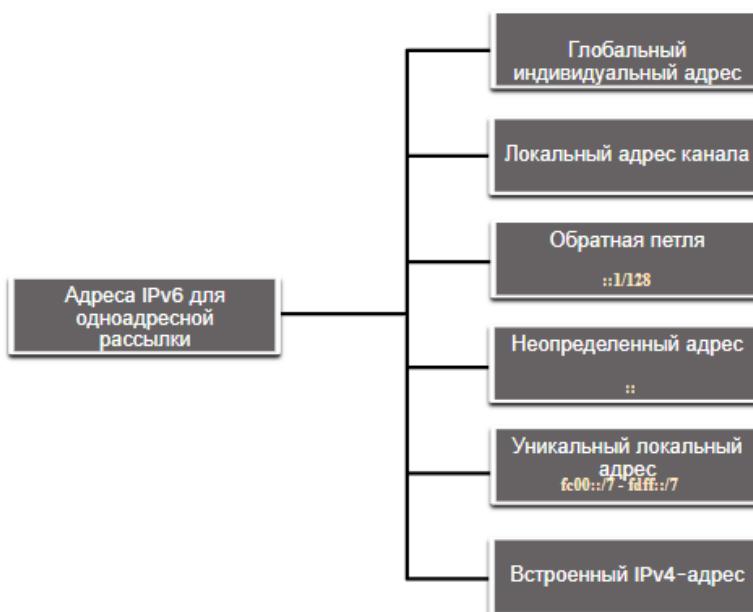
Многоадресный пакет — это пакет с IP-адресом назначения, который является адресом многоадресной рассылки. Для многоадресной рассылки в протоколе IPv4 зарезервированы адреса от 224.0.0.0 до 239.255.255.255.

Как и в случае IPv4, существуют различные типы адресов IPv6. На самом деле, существует три широкие категории адресов **IPv6**:

- **Индивидуальный (или одноадресной рассылки, unicast)** : служит для однозначного определения интерфейса на устройстве под управлением протокола IPv6.
- **Групповой (или адрес многоадресной рассылки)** : используется для отправки одного IPv6-пакета на несколько адресов назначения.
- **Произвольный (или адрес произвольной рассылки)** : любой индивидуальный IPv6-адрес, который может быть назначен нескольким устройствам. Пакет, отправляемый на адрес произвольной рассылки, направляется к ближайшему устройству с этим адресом. Произвольные адреса в данном курсе не рассматриваются.

В отличие от IPv4, IPv6 не использует широковещательный адрес. Однако есть групповой IPv6-адрес для всех узлов, который дает аналогичный результат.

Адреса IPv6 для одноадресной рассылки



d. Области действия IPv4 и IPv6.

IPv4 делится на частные и публичные.

Публичные IPv4-адреса представляют собой адреса, на глобальном уровне маршрутизируемые между маршрутизаторами интернет-провайдеров (Internet Service Provider, ISP). Однако, не все доступные IPv4-адреса можно использовать в Интернете. Имеются блоки адресов, называемые частными адресами, которые в большинстве компаний назначаются в качестве IPv4-адресов внутренних хостов.

- **Сетевой и широковещательный** – первый и последний адреса сети не могут быть назначены конечным узлам
- **Loopback** – 127.0.0.1 специальный адрес, которой направляет все сообщение на узел отправитель (127.0.0.0 – 127.255.255.255 зарезервированы)

The Private Address Blocks

Сетевой адрес и префикс	Диапазон частных адресов RFC 1918
10.0.0.0/8	10.0.0.0 - 10.255.255.255
172.16.0.0/12	172.16.0.0 - 172.31.255.255
192.168.0.0/16	192.168.0.0 - 192.168.255.255

Частный адреса

Некоторые адреса (например, сетевые и широковещательные) нельзя назначать узлам. Также есть особые адреса, которые можно назначать узлам, но с ограничениями способов взаимодействия этих узлов в сети.

адреса loopback

Адреса loopback (127.0.0.0 /8 или от 127.0.0.1 до 127.255.255.254): чаще определяются как только один адрес 127.0.0.1— это особые адреса, которые используют узлы, чтобы направлять трафик самим себе.

Например, они могут использоваться узлом, чтобы проверить работоспособность конфигурации TCP/IP, как показано на рисунке

Область действия IPv6-адресов

- Global unicast address
 - Аналог публичных IPv4-адресов
 - Распределяются IANA
 - Блок адресов: **2000::/3**
- Unique-local address
 - Аналог частных IPv4-адресов
 - Блоки **FC00::/7** и **FD00::/8**
- Link-local address
 - Не маршрутизируются
 - Назначаются автоматически
 - В пределах локального канала
 - Блок **FE80::/10**
- Site-local address
 - Не используются



В отличие от устройств IPv4, имеющих только один адрес, **адреса IPv6** обычно имеют два одноадресных адреса:

- **Глобальный индивидуальный адрес** аналогичен публичному IPv4-адресу. Эти адреса, к которым можно проложить маршрут по Интернету, являются уникальными по всему миру. Глобальные индивидуальные адреса могут быть настроены статически или присвоены динамически.
- **Локальный адрес канала (LLA)** — это необходимо для каждого устройства с поддержкой IPv6. Локальные адреса канала используются для обмена данными с другими устройствами по одному локальному каналу. В протоколе IPv6 термин «канал» означает подсеть. Локальные адреса каналов ограничены одним каналом. Они должны быть уникальны только в рамках этого канала, поскольку вне канала к ним нельзя проложить маршрут. Другими словами, маршрутизаторы не смогут пересыпать пакеты, имея локальный адрес канала источника или назначения.

Unique-local address - Уникальные локальные адреса (диапазон fc00::/7 до fdff::/7) пока не реализованы. Таким образом, этот модуль охватывает только конфигурацию GUA и LLA. Однако уникальные локальные адреса могут использоваться для адресов устройств, которые не должны быть доступны извне, таких как внутренние серверы и принтеры.

Уникальные локальные IPv6-адреса имеют некоторые общие особенности с частными адресами RFC 1918 для IPv4, но при этом между ними имеются и значительные различия.

- Уникальные локальные адреса используются для локальной адресации в пределах узла или между ограниченным количеством узлов.
- Уникальные локальные адреса могут использоваться для устройств, которым никогда не понадобится использование других сетей или получение из них данных.
- Уникальные локальные адреса не маршрутизируются глобально и не преобразуются в глобальный адрес IPv6.

Глобальные индивидуальные IPv6-адреса (GUA) уникальны по всему миру и доступны для маршрутизации через Интернет IPv6. Эти адреса эквивалентны публичным IPv4-адресам. Корпорация по управлению доменными именами и IP-адресами (Internet Committee for Assigned Names and Numbers, ICANN), оператор Администрации адресного пространства Интернет (IANA) выделяет блоки IPv6-адресов пяти региональным интернет-регистраторам (RIR). В настоящее время назначаются только глобальные индивидуальные адреса с первыми тремя битами 001 или 2000::/3.

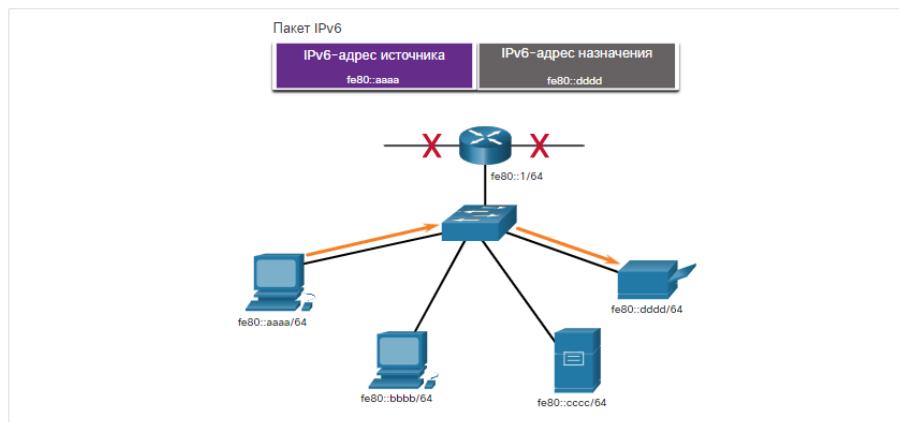
Локальный IPv6-адрес канала позволяет устройству обмениваться данными с другими устройствами с включенным протоколом IPv6 в том же канале (подсети) и только в нем. Пакеты с локальным адресом канала источника или назначения не могут быть направлены за пределы канала, в котором создается пакет.

ГUA не является обязательным требованием. Однако каждый сетевой интерфейс с поддержкой IPv6 должен иметь LLA.

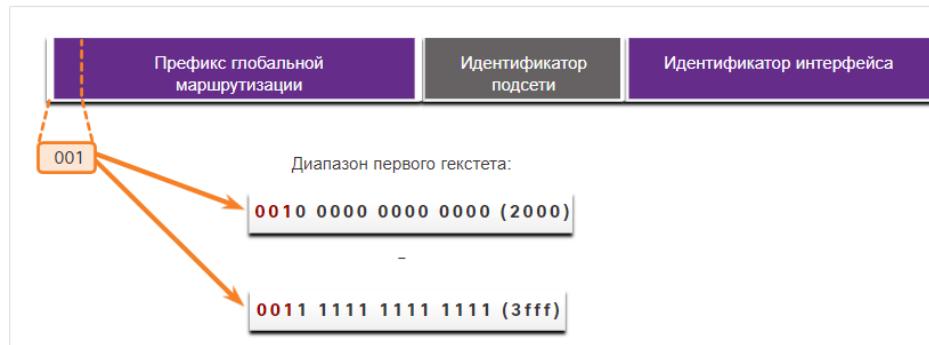
Если локальный адрес канала не настроен вручную на интерфейсе, устройство автоматически создает его самостоятельно, не обращаясь к DHCP-серверу. Узлы под управлением IPv6 создают локальный IPv6-адрес канала даже в том случае, если устройству не был назначен глобальный индивидуальный IPv6-адрес. Это позволяет устройствам под управлением IPv6 обмениваться данными с другими устройствами под управлением IPv6 в одной подсети, в том числе со шлюзом по умолчанию (маршрутизатором).

IPv6 LLA находятся в диапазоне fe80::/10. /10 указывает, что первые 10 битов — 1111 1110 10xx xxxx. Диапазон значений первого гекстета: от 1111 1110 1000 0000 (fe80) до 1111 1110 1011 1111 (febf).

Обмен данными между локальными IPv6-адресами канала



Глобальные индивидуальные IPv6-адреса (GUA)

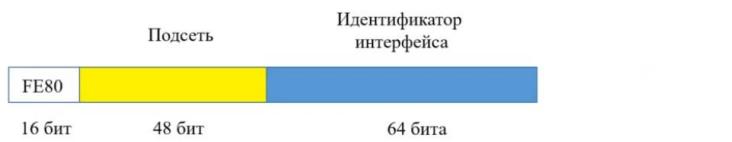


Префикс глобальной маршрутизации 48 бит. Выделяется организациям. 16 бит для создания подсетей.

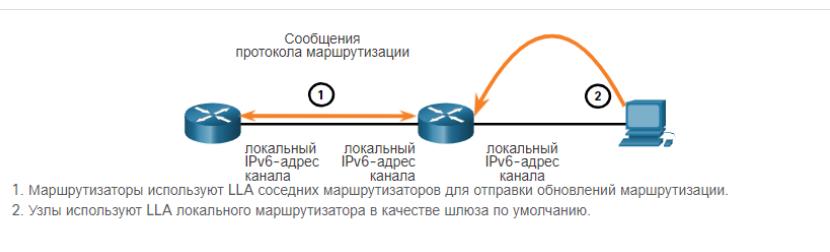
Адрес IPv6 с префиксом глобальной маршрутизации /48 и префиксом /64



Link-local address (LLA)



Диапазон значений первого гексетта: от 1111 1110 1000 0000 (fe80) до 1111 1110 1011 1111(fe8f)



Специальные IP-адреса

IPv4:

- 0.0.0.0/0 — маршрут по умолчанию
- 127.0.0.0 /8 — адреса обратной петли (loopback)
- 169.254.0.0 /16 — локальные адреса канала
- 192.0.2.0/24 — адреса TEST-NET, используются для обучения

IPv6:

- ::/128 — текущий хост
- ::/0 — маршрут по умолчанию
- ::1/128 — адреса обратной петли (loopback)
- FF02::1 — все узлы на канале связи
- FF02::2 — все маршрутизаторы на канале связи

Локальные адреса канала – назначаются автоматически, если не смогли назначить. Не маршрутизируются никуда.

f. Способы назначения адресов IPv4 и IPv6.

Назначение IPv6-адресов

Статически

Автоматически/динамически

Задать глобальный адрес на интерфейс

```
R(config-if)# ipv6 address ipv6-address/prefix-length
```

Задать link-local адрес на интерфейс

```
R(config-if)# ipv6 address ipv6-link-local-address link-local
```

Включение маршрутизации IPv6

```
R(config)# ipv6 unicast-routing
```

Просмотр таблицы маршрутизации

```
R# show ipv6 route
```

ipv6 address

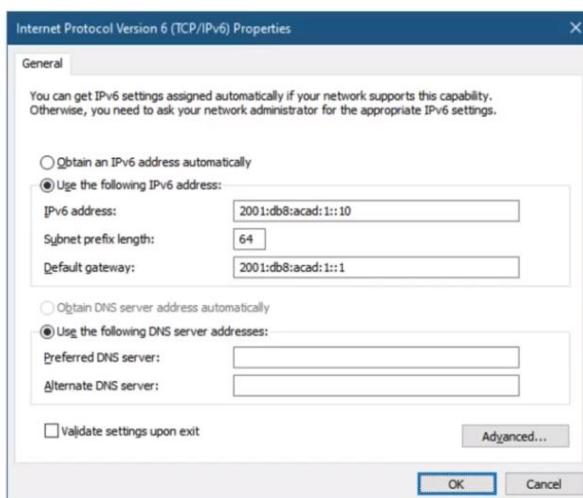
```
R1(config)# interface gigabitethernet 0/0/0
R1(config-if)# ipv6 address 2001:db8:acad:1::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface gigabitethernet 0/0/1
R1(config-if)# ipv6 address 2001:db8:acad:2::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface serial 0/1/0
R1(config-if)# ipv6 address 2001:db8:acad:3::1/64
R1(config-if)# no shutdown
```

```
R1(config)# interface gigabitethernet 0/0/0
R1(config-if)# ipv6 address fe80::1:1 link-local
R1(config-if)# exit
R1(config)# interface gigabitethernet 0/0/1
R1(config-if)# ipv6 address fe80::1:2 link-local
R1(config-if)# exit
R1(config)# interface serial 0/1/0
R1(config-if)# ipv6 address fe80::1:3 link-local
R1(config-if)# exit
```

```
show ipv6 route
```

```
R1#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
      U - Per-user Static route, M - MIPv6
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
      O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
      D - EIGRP, EX - EIGRP external
C 2001:DB8:ACAD:1::/64 [0/0]
  via GigabitEthernet0/0, directly connected
L 2001:DB8:ACAD:1::1/128 [0/0]
  via GigabitEthernet0/0, receive
C 2001:DB8:ACAD:2::/64 [0/0]
  via GigabitEthernet0/1, directly connected
L 2001:DB8:ACAD:2::1/128 [0/0]
```

Задание IPv6 на хосте



Рекомендуется в качестве default gateway указывать линк локал адрес.

Назначение IPv6-адресов

Статически

Автоматическая/динамическая настройка адресов:

Без сохранения состояния (SLAAC, SLAAC + DHCPv6)

С сохранением состояния (DHCPv6)

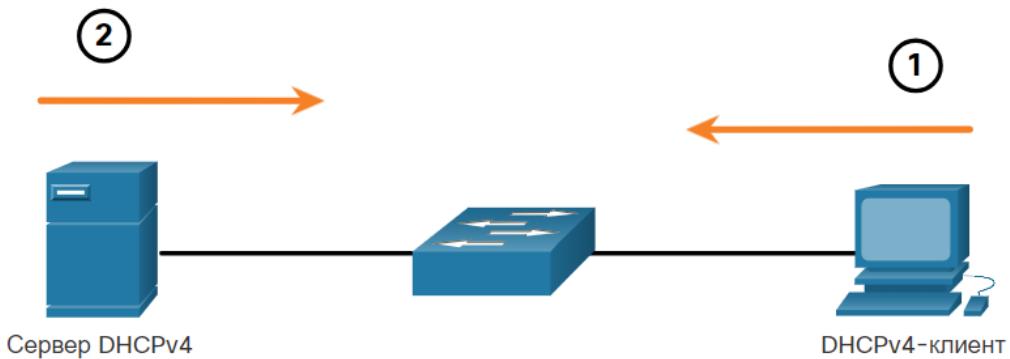
6. DHCP:

a. Определение. Алгоритм работы DHCP. Ретрансляция DHCP.

Протокол динамической конфигурации узла v4 (DHCPv4) динамически назначает адреса IPv4 и другую информацию о конфигурации сети. Поскольку стационарные ПК обычно составляют основную часть сетевых узлов, протокол DHCPv4 является крайне полезным инструментом, позволяющим сетевым администраторам значительно экономить время.

Сервер DHCPv4 динамически назначает или выдает в аренду IPv4-адрес из пула адресов на ограниченный период времени по выбору сервера или до тех пор, пока у клиента есть необходимость в адресе.

Клиенты арендуют данные у сервера на период, определенный администратором. Администраторы настраивают серверы DHCPv4 таким образом, чтобы срок аренды истекал в разное время. Срок аренды обычно составляет от 24 часов до недели или более. По истечении срока аренды клиент должен запросить другой адрес, хотя в большинстве случаев клиенту повторно назначается тот же адрес.



1. Процесс аренды DHCPv4 начинается с отправки клиентом сообщения с запросом служб DHCP-сервера.
2. Если есть DHCPv4 сервер, который получает сообщение, он будет отвечать IPv4 адрес и возможные другие сведения о конфигурации сети.

Принципы работы DHCPv4

DHCPv4 работает по модели «клиент-сервер». Когда клиент подключается к серверу DHCPv4, сервер присваивает или сдает ему в аренду IPv4-адрес. Клиент с арендованным IP-адресом подключается к сети до истечения срока аренды. Периодически клиент должен связываться с DHCP-сервером для продления срока аренды. Благодаря подобному механизму «переехавшие» или отключившиеся клиенты не занимают адреса, в которых они больше не нуждаются. По истечении срока аренды сервер DHCP возвращает адрес в пул, из которого адрес может быть повторно получен при необходимости.

При начальной загрузке клиента (или ином способе подключения к сети) начинается 4-шаговый процесс получения адреса в аренду.

1. Обнаружение DHCP (DHCPDISCOVER)
2. Предложение DHCP (DHCPOFFER)
3. Запрос DHCP (DHCPREQUEST)
4. Подтверждение DHCP (DHCPACK)



(использование)

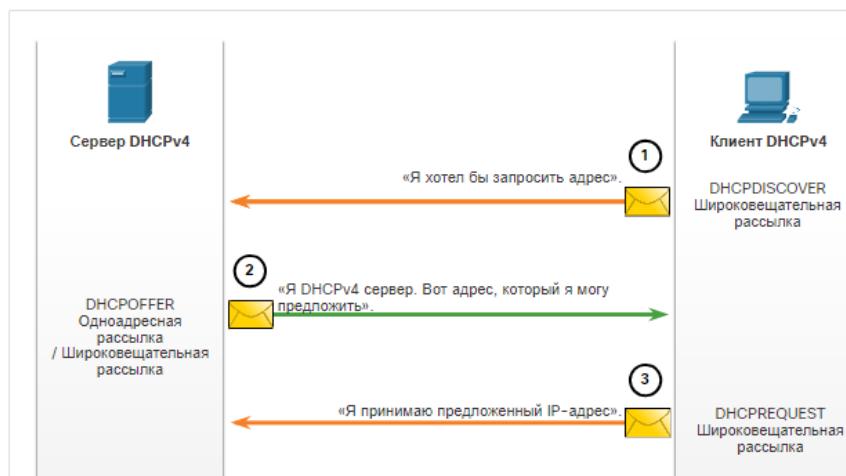
Шаг 2. DHCP Offer (предложение)

Когда клиент получает от сервера сообщение DHCPOFFER, он отправляет в ответ сообщение DHCPREQUEST. Это сообщение используется как для первоначальной аренды адреса, так и для ее продления. Когда сообщение используется при первоначальной аренде, DHCPCREQUEST служит уведомлением о принятии предложения привязки к предложенным сервером параметрам и косвенным отклонением для всех других серверов, которые могли предоставить клиенту предложение привязки.

Шаг 3. DHCP Request (запрос)

Шаг 4. Подтверждение DHCP (DHCPACK)

В корпоративных сетях часто используется несколько DHCPv4-серверов. Сообщение DHCPCREQUEST отправляется в форме широковещательной рассылки с целью информирования данного DHCPv4-сервера и других DHCPv4-серверов о том, что предложение было принято.



(обнаружение)

Шаг 2. DHCP Offer (предложение)

Когда сервер DHCPv4 получает сообщение DHCPDISCOVER, он резервирует доступные IPv4-адреса для выдачи в аренду клиенту. Сервер также создает запись ARP, состоящую из MAC-адреса запрашивающего клиента и выданного клиенту IPv4-адреса. DHCPv4-сервер посыпает сообщение привязки DHCPOFFER запрашивающему клиенту.

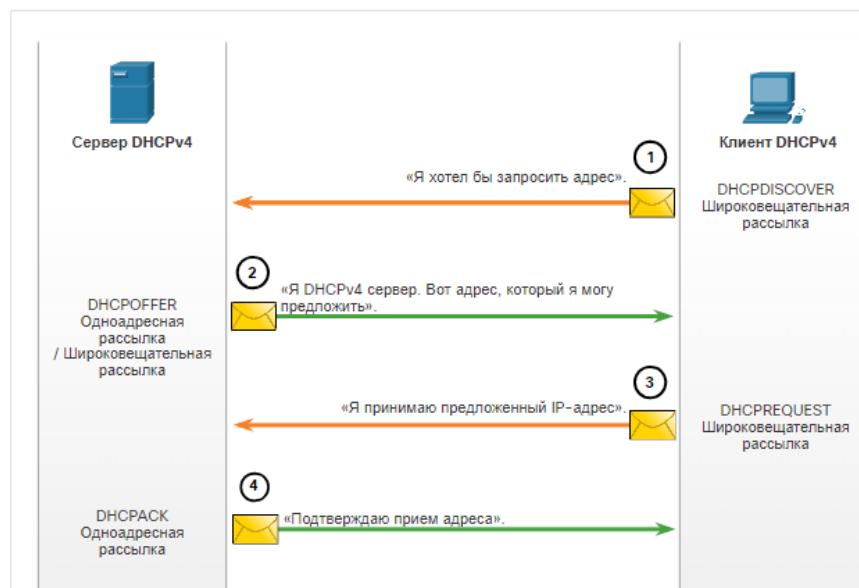
Шаг 3. DHCP Request (запрос)

Шаг 4. Подтверждение DHCP (DHCPACK)



(обнаружение)
Шаг 2. DHCP Offer (предложение)
Шаг 3. DHCP Request (запрос)

Шаг 4. Подтверждение DHCP (DHCPACK)
--



Шаги, чтобы возобновить аренду

До истечения срока аренды клиент начинает двухэтапный процесс продления аренды с сервером DHCPv4, как показано на рисунке:

1. DHCP Request (DHCPREQUEST)

Перед окончанием аренды клиент отправляет сообщение DHCPREQUEST непосредственно DHCPv4-серверу, который первоначально предложил IPv4-адрес. Если сообщение DHCPACK не получено за определенный период времени, клиент отправляет другое сообщение DHCPREQUEST широковещательной рассылкой, чтобы другой DHCPv4-сервер мог продлить срок аренды.

2. DHCP Acknowledgment (DHCPACK)

При получении сообщения DHCPREQUEST сервер подтверждает информацию об аренде ответным сообщением DHCPACK.

Примечание: Эти сообщения (в первую очередь DHCPOFFER и DHCPACK) могут отправляться в виде одноадресной рассылки или широковещательной рассылки в соответствии с IETF RFC 2131.



Для настройки сервера DHCPv4 Cisco IOS выполните следующие действия:

Шаг 1. Исключение IPv4-адресов

Шаг 2. Определение имени пула DHCPv4.

Шаг 3. Создание пула DHCPv4

Шаг 1. Исключение IPv4-адресов

Маршрутизатор, выполняющий функцию DHCPv4-сервера, присваивает все IPv4-адреса из пула DHCPv4-адресов, если конфигурацией не предусмотрено исключение отдельных адресов. Как правило, некоторые IPv4-адреса из пула присваиваются сетевым устройствам для постоянного использования. Следовательно, эти IPv4-адреса не должны присваиваться другим устройствам. Для исключения адресов IPv4 используется следующий синтаксис команды:

```
Router(config)# ip dhcp excluded-address low-address [high-address]
```

Можно исключить один адрес или диапазон адресов, задав адреса нижнего и верхнего пределов диапазона. В число исключенных адресов должны входить адреса, присвоенные маршрутизаторам, серверам, принтерам и другим устройствам, которые были или будут настроены вручную. Можно также ввести команду несколько раз.

Шаг 2. Определение имени пула DHCPv4.

При выполнении настройки DHCPv4-сервера задается пул адресов, предназначенных для распределения.

Как показано в примере, команда **ip dhcp pool *pool-name*** создает пул с указанным именем и переводит маршрутизатор в режим конфигурации DHCPv4, который определяется приглашением Router (dhcp-config) #.

Для определения пула используется следующий синтаксис команды

```
Router(config)# ip dhcp pool pool-name
Router(dhcp-config)#
```

Шаг 3. Создание пула DHCPv4

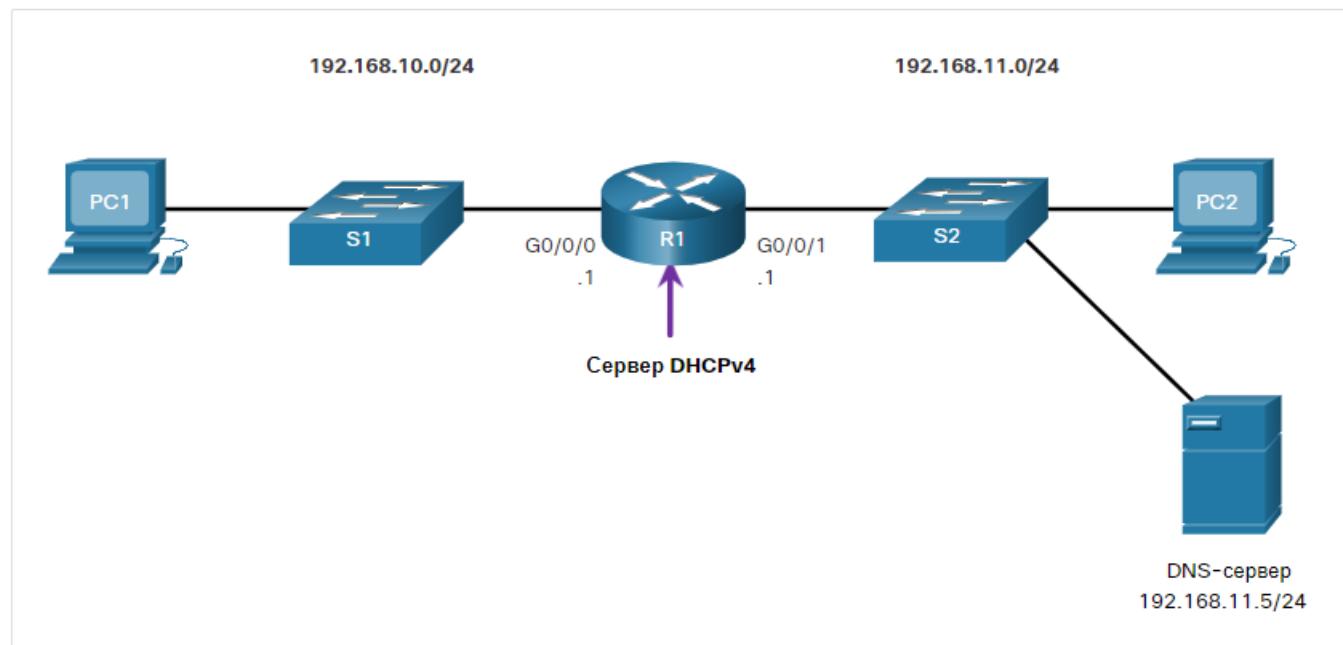
В таблице перечислены задачи для завершения настройки пула DHCPv4.

Пул адресов и основной шлюз маршрутизатора должны быть настроены. Используйте команду **network** для определения диапазона доступных адресов. Используйте команду **default-router**, чтобы задать основной шлюз маршрутизатора. Шлюзом обычно выступает интерфейс LAN маршрутизатора, ближайшего к клиентским устройствам. Требуется только один шлюз, но при наличии нескольких шлюзов можно перечислить вплоть до восьми адресов.

Остальные команды DHCPv4-пула являются дополнительными. Например, IPv4-адрес DNS-сервера, доступный DHCPv4-клиенту, настраивается с помощью команды **dns-server**. Команда **domain-name** используется, чтобы задать доменное имя. Продолжительность аренды протокола DHCPv4 изменяется командой **lease**. По умолчанию продолжительность аренды равна одному дню. Чтобы задать сервер NetBIOS WINS, используется команда **netbios-name-server**.

Задача	Команда IOS
Определение пула адресов.	<code>network network-number [mask /prefix-length]</code>
Определение маршрутизатора или шлюза по умолчанию.	<code>default-router address [address2...address8]</code>
Назначение DNS-сервера.	<code>dns-server address [address2...address8]</code>
Назначение доменного имени.	<code>domain-name domain</code>
Определение срока DHCP-аренды.	<code>lease {days [hours [minutes]] infinite}</code>
Определение сервера NetBIOS WINS.	<code>netbios-name-server address [address2...address8]</code>

Для примера настройки используется топология, показанная на рисунке.



В примере показано, как сделать R1 сервером DHCPv4 для локальной сети 192.168.10.0/24.

```

R1(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9
R1(config)# ip dhcp excluded-address 192.168.10.254
R1(config)# ip dhcp pool LAN-POOL-1
R1(dhcp-config)# network 192.168.10.0 255.255.255.0
R1(dhcp-config)# default-router 192.168.10.1
R1(dhcp-config)# dns-server 192.168.11.5
R1(dhcp-config)# domain-name example.com
R1(dhcp-config)# end
R1#
  
```

Команды настройки

Исключить IPv4-адреса

```
R(config)# ip dhcp excluded-address low_address [high_address]
```

Создать пул DHCP

```
R(config)# ip dhcp pool pool_name
```

Задать пул адресов

```
R(dhcp-config)# network network-number [mask | /prefix-length]
```

Задать маршрутизатор или шлюз по умолчанию

```
R(dhcp-config)# default-router address [address2...address8]
```

Задать маршрутизатор или шлюз по умолчанию

```
R(dhcp-config)# lease {days [hours [minutes]] | infinite}
```

```
show running-config | section  
dhcp
```

```
R1# show running-config | section dhcp
ip dhcp excluded-address 192.168.10.1 192.168.10.9
ip dhcp excluded-address 192.168.10.254
ip dhcp pool LAN-POOL-1
  network 192.168.10.0 255.255.255.0
  default-router 192.168.10.1
  dns-server 192.168.11.5
  domain-name example.com
```

```
show ip dhcp binding
```

```
R1# show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address     Client-ID/          Lease expiration      Type      State       Interface
               Hardware address/
               User name
192.168.10.10  0100.5056.b3ed.08    Sep 15 2019 8:42 AM  Automatic  Active
GigabitEthernet0/0/0
```

```
show ip dhcp server statistics
```

```
R1# show ip dhcp server statistics
Memory usage           19465
Address pools          1
Database agents         0
Automatic bindings      2
Manual bindings         0
Expired bindings        0
Malformed messages      0
Secure arp entries      0
Renew messages          0
Workspace timeouts      0
Static routes           0
Relay bindings          0
Relay bindings active    0
Relay bindings terminated 0
Relay bindings selecting 0
Message                Received
  BOOTREQUEST            0
  DHCPDISCOVER           4
  DHCPREQUEST            2
  DHCPRECLINE             0
  DHCPRELEASE             0
  DHCPINFORM              0
```

Отключение DHCP-сервера

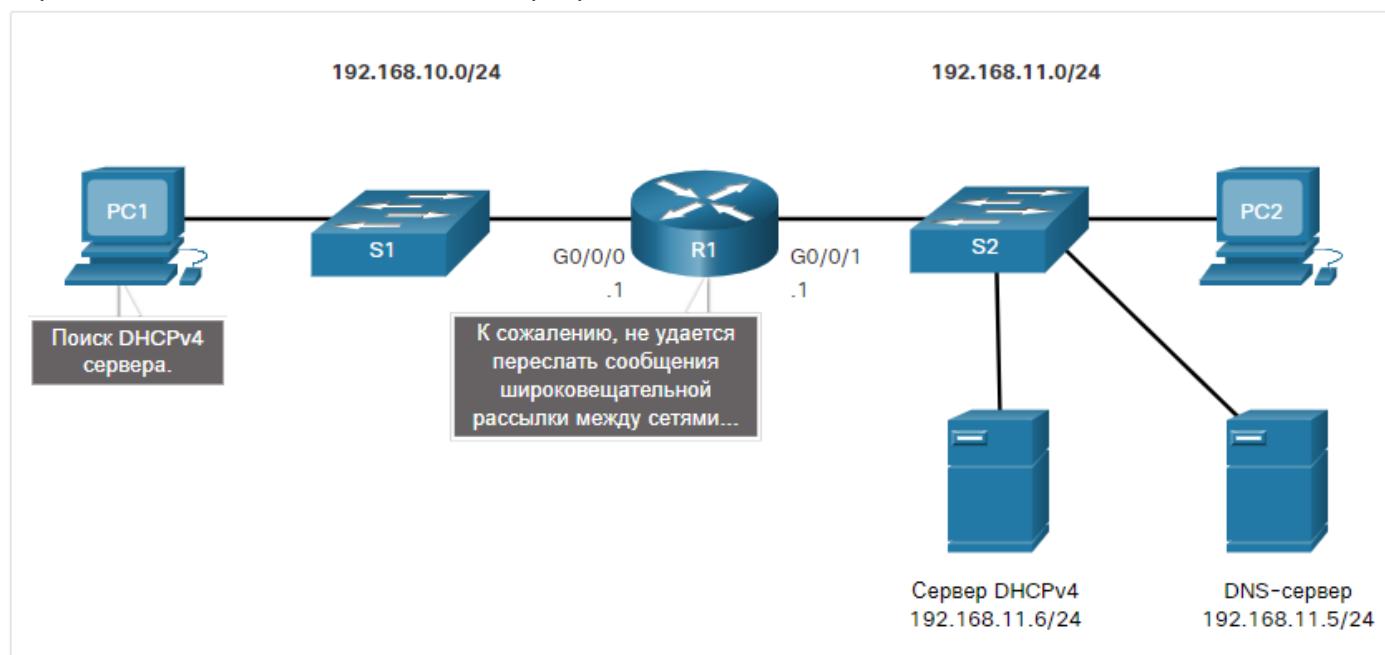
```
R1(config)# no service dhcp  
R1(config)# service dhcp  
R1(config)#{
```

Включен по умолчанию!

DHCPv4-ретрансляция

В сложной иерархической сети корпоративные серверы обычно располагаются в серверной ферме. Данные серверы могут предоставлять службы DHCP, DNS, TFTP и FTP. Клиенты сети и серверы, как правило, находятся в разных подсетях. Для определения местоположения серверов и получения услуг клиенты часто используют сообщения широковещательной рассылки.

На рисунке показана попытка PC1 получить IPv4-адрес от DHCP-сервера при помощи сообщения широковещательной рассылки. В этом сценарии маршрутизатор R1 не настроен в качестве DHCPv4-сервера и не отправляет сообщения широковещательной рассылки. Поскольку DHCPv4-сервер расположен в другой сети, PC1 не может получить IP-адрес через DHCP. R1 должен быть настроен на ретрансляцию сообщений DHCPv4 на сервер DHCPv4.



`ipconfig /release`

PC1 – это компьютер с ОС Windows. Администратор сети сбрасывает всю текущую информацию об IPv4-адресации с помощью команды `ipconfig /release`. Обратите внимание, что IPv4-адрес освобождается. Отображаемый адрес должен выглядеть как 0.0.0.0.

```
C:\Users\Student> ipconfig /release  
Windows IP Configuration  
Ethernet adapter Ethernet0:  
  Connection-specific DNS Suffix . :  
  Default Gateway . . . . . :
```

ipconfig /renew

Затем администратор сети пытается обновить сведения об адресации IPv4 с помощью команды **ipconfig /renew**. Команда инициирует отправку сообщения DHCPDISCOVER широковещательной рассылки устройством PC1. Выходные данные указывают, что PC1 не смог найти DHCPv4-сервер. Запрос не выполнен, поскольку маршрутизаторы не пересыпают сообщения широковещательной рассылки.

Администратор сети может добавить DHCPv4 серверы на R1 для всех подсетей. Однако это повлечет за собой дополнительные расходы и административные накладные расходы.

```
C:\Users\Student> ipconfig /renew
Windows IP Configuration
An error occurred while renewing interface Ethernet0 : unable to connect to your DHCP server. Request
has timed out.
```

ip helper-address

Лучше всего настроить R1 с помощью команды конфигурации интерфейса **ip helper-address address**. Это приведет к тому, что R1 будет ретранслировать широковещательные рассылки DHCPv4 на сервер DHCPv4. Как показано в примере, интерфейс R1, принимающий широковещательную рассылку от PC1, настроен на ретрансляцию DHCPv4 на сервер DHCPv4 по адресу 192.168.11.6.

```
R1(config)# interface g0/0/0
R1(config-if)# ip helper-address 192.168.11.6
R1(config-if)# end
R1#
```

show ip interface

Когда маршрутизатор R1 сконфигурирован как агент DHCPv4-ретрансляции, он принимает широковещательные запросы, а затем отправляет эти запросы как одноадресную рассылку на IPv4-адрес 192.168.11.6. Администратор сети может использовать эту команду **show ip interface** для проверки конфигурации.

```
R1# show ip interface g0/0/0
GigabitEthernet0/0/0 is up, line protocol is up
  Internet address is 192.168.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is 192.168.11.6
(дальше выходные данные опущены)
```

ipconfig /all

Как показано в выходных данных, PC1 теперь может получить адрес IPv4 с сервера DHCPv4, как это было подтверждено с помощью команды **ipconfig /all**.

```
C:\Users\Student> ipconfig /all
Настройка IP для Windows

Ethernet adapter Ethernet0:
  Connection-specific DNS Suffix . : example.com
  IPv4 Address. . . . . : 192.168.10.10
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.10.1
```

Настройка DHCP-клиента

Настройка маршрутизатора в качестве DHCP-клиента

```
R(config-if)# ip address dhcp
```



```
SOHO(config)# interface G0/0/1
SOHO(config-if)# ip address dhcp
SOHO(config-if)# no shutdown
Sep 12 10:01:25.773: %ONCP-6-ADDRESS_ASSIGN: Interface GigabitEthernet0/0/1 assigned DHCP address
209.165.201.12, mask 255.255.255.224, hostname SOHO
```

show ip interface

```
SOHO# show ip interface g0/0/1
GigabitEthernet0/0/1 is up, line protocol is up
  Internet address is 209.165.201.12/27
  Broadcast address is 255.255.255.255
  Address determined by DHCP
  (output omitted)
```

b. Методы назначения IPv6. Алгоритмы работы.

Назначение IPv6-адресов

Автоматическая/динамическая настройка адресов:

Без сохранения состояния (SLAAC, SLAAC + DHCPv6)

С сохранением состояния (DHCPv6)

Обнаружение маршрутизаторов (сообщения ICMPv6)

Router Advertisement (RA)

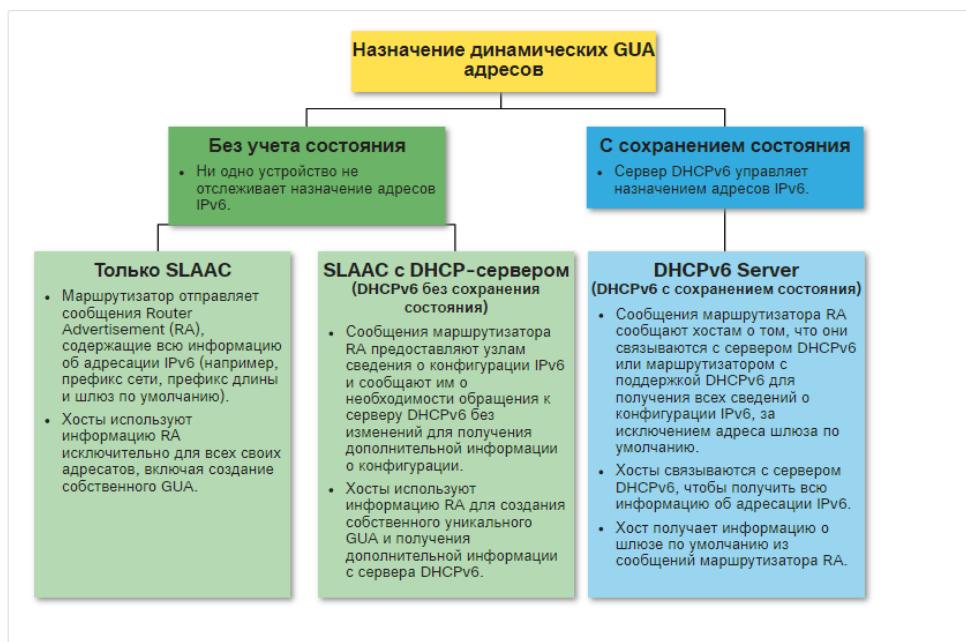
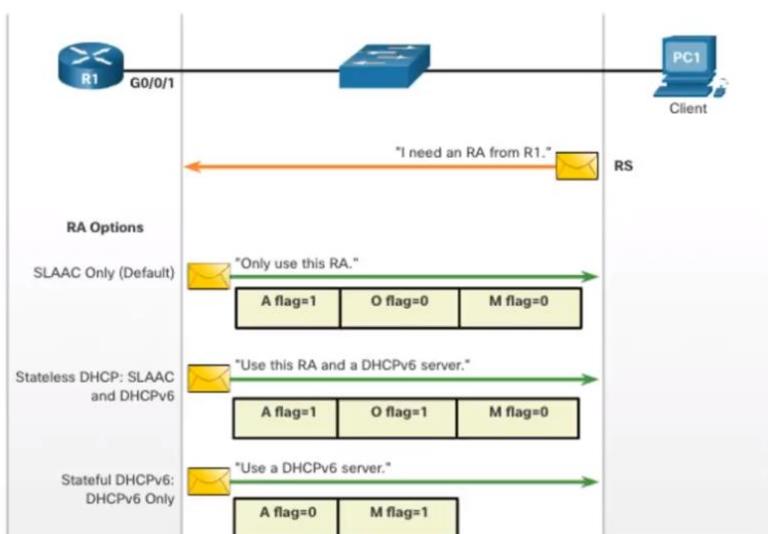
Router Solicitation (RS)

Флаги

Managed Configuration (M)

Other Configuration (O)

Назначение IPv6-адреса



На маршрутизаторе глобальный одноадресный адрес IPv6 настраивается вручную с помощью команды конфигурации интерфейса **ipv6 address ipv6-address/prefix-length**.

Сообщения ICMPv6

Информационные сообщения и сообщения об ошибках, возникающие в протоколе ICMPv6, очень похожи на сообщения о контроле и ошибках, используемые протоколом ICMPv4. Однако протокол ICMPv6 отличается расширенной функциональностью и новыми возможностями, которых нет в ICMPv4. Сообщения ICMPv6 инкапсулируются в IPv6-пакеты.

ICMPv6 включает четыре новых протокола в составе протокола обнаружения соседних узлов (Neighbor Discovery Protocol, ND или NDP).

Обмен сообщениями между маршрутизатором IPv6 и устройством IPv6, включая динамическое распределение адресов, осуществляется следующим образом:

- Сообщение «Запрос к маршрутизатору» (Router Solicitation, RS)
- Сообщение «Ответ маршрутизатора» (Router Advertisement, RA)

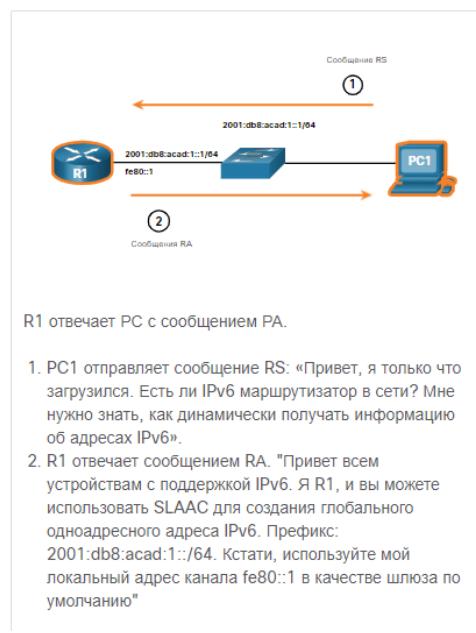
Обмен сообщениями между устройствами IPv6, включая обнаружение повторяющихся адресов и разрешение адресов, осуществляется следующим образом:

- Сообщение с запросом поиска соседей (NS)
- Сообщение об объявлении соседних узлов (NA)

Сообщения RA отправляются маршрутизаторами с поддержкой IPv6 каждые 200 секунд для предоставления информации об адресации узлам с поддержкой IPv6. Сообщение RA может включать такие данные об адресах для хостов, как префикс, длина префикса, DNS-адрес и доменное имя. Узел, использующий SLAAC, установит в качестве своего шлюза по умолчанию локальный адрес канала маршрутизатора, отправившего RA.



Маршрутизатор с поддержкой IPv6 также отправит сообщение RA в ответ на сообщение RS. На рисунке PC1 отправляет сообщение RS, чтобы определить, как получать информацию об адресах IPv6 динамически.



Обзор SLAAC

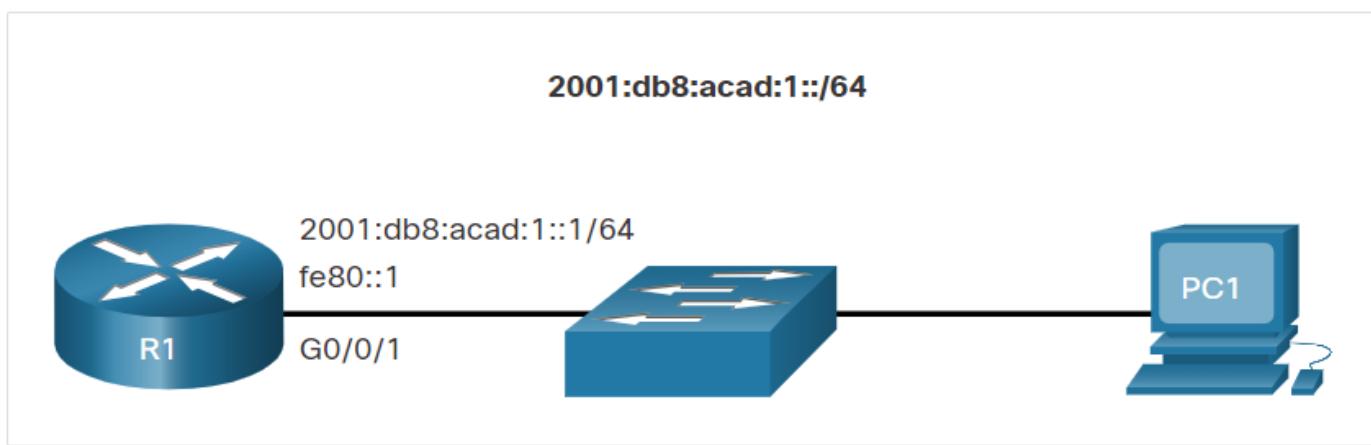
Не каждая сеть имеет доступ к серверу DHCPv6. Но каждое устройство в сети IPv6 нуждается в GUA. Метод SLAAC позволяет хостам создавать свой собственный уникальный глобальный одноадресный адрес IPv6 без использования служб DHCPv6 сервера.

SLAAC - это служба без сохранения состояния. Это означает, что нет сервера, который хранит информацию о сетевых адресах, чтобы узнать, какие IPv6-адреса используются и какие из них доступны.

SLAAC использует ICMPv6-сообщения запроса маршрутизатора и объявления маршрутизатора, чтобы предоставить информацию об адресации и другую информацию о конфигурации, обычно предоставляемую DHCP-сервером. Хост настраивает свой IPv6-адрес на основе информации, отправляемой в RA. Сообщения RA отправляются маршрутизатором IPv6 каждые 200 секунд.

Хост также может отправить сообщение Router Solicitation (RS) с запросом о том, чтобы маршрутизатор с поддержкой IPv6 передал хосту RA.

SLAAC может быть развернут только как SLAAC, или SLAAC с DHCPv6.



Предположим, что R1 GigabitEthernet 0/0/1 настроен с указанными IPv6 GUA и локальными адресами канала. Нажмите каждую кнопку, чтобы объяснить, как R1 включен для SLAAC.

Выходные данные команды **show ipv6 interface** отображают текущие настройки интерфейса G0/0/1.

Как было подчеркнуто, R1 были назначены следующие адреса IPv6:

- Локальный адрес канала IPv6 - fe80::1
- GUA и подсеть адрес канала IPv6 - 2001:db8:acad:1::1 и 2001:db8:acad:1::/64
- группа всех узлов IPv6 - ff02::1

```
R1# show ipv6 interface g0/0/1
GigabitEthernet0/0/1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::1
  No Virtual link-local address(es):
  Description: Link to LAN
  Global unicast address(es):
    2001:DB8:ACAD:1::1, subnet is 2001:DB8:ACAD:1::/64
  Joined group address(es):
    FF02::1
    FF02::1:FF00:1
  ( дальне выходные данные опущены)
R1#
```

Проверка адресов IPv6

Активация IPv6-маршрутизации

Убедитесь, что SLAAC включен

Несмотря на то, что интерфейс маршрутизатора имеет конфигурацию IPv6, он все еще не включен для отправки RAs, содержащих сведения о конфигурации адресов, на узлы, использующие SLAAC.

Чтобы включить отправку сообщений RA, маршрутизатор должен присоединиться к группе IPv6 all-routers с помощью команды **ipv6 unicast-routing** в режиме глобальной конфигурации, как показано в выходных данных.

```
R1(config)# ipv6 unicast-routing
R1(config)# exit
R1#
```

Сообщение RS отправляется на IPv6-адрес многоадресной рассылки FF02::2, который поддерживают все маршрутизаторы. Вы можете использовать эту команду **show ipv6 interface**, чтобы проверить, включен ли маршрутизатор, как показано на выходных данных.

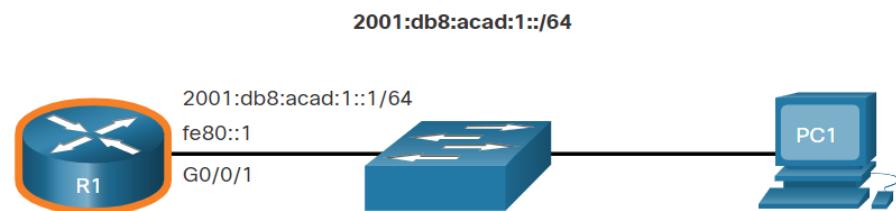
Маршрутизатор Cisco с поддержкой IPv6 отправляет сообщения RA на адрес многоадресной рассылки всех узлов IPv6 ff02::1 каждые 200 секунд.

```
R1# show ipv6 interface G0/0/1 | section Joined
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF00:1
R1#
```

Метод SLAAC только включен по умолчанию при настройке команды **ipv6 unicast-routing**. Также настроить глобальный ipv6 на маршрутизаторе. Все включенные интерфейсы Ethernet с настроенным GUA IPv6 начнут отправлять сообщения RA с флагом A равным 1, а флаги O и M — 0, как показано на рисунке.

Флаг **A = 1** предлагает клиенту создать свой собственный IPv6 GUA, используя префикс, объявленный в RA. Клиент может создать свой собственный идентификатор интерфейса, используя метод Extended Unique Identifier (EUI-64) или случайно сгенерированный.

Флаги **O =0** и **M=0** указывают клиенту использовать информацию исключительно в сообщении RA. Сюда входит информация о префиксе, длине префикса, DNS-сервере, MTU и информация о шлюзе по умолчанию. Далее клиент не получает никакой информации от сервера DHCPv6.



Сообщение RA

Флаг	значение
A	1
O	0
M	0



В примере PC1 включен для автоматического получения информации об адресации IPv6 . Из-за настроек флагов A, O и M PC1 выполняет только SLAAC, используя информацию, содержащуюся в сообщении RA, отправленном R1.

Адрес шлюза по умолчанию — это адрес источника IPv6 сообщения RA, который является LLA для R1. Адрес шлюза по умолчанию может быть получен только динамически из сообщения RA. Сервер DHCPv6 не предоставляет эту информацию

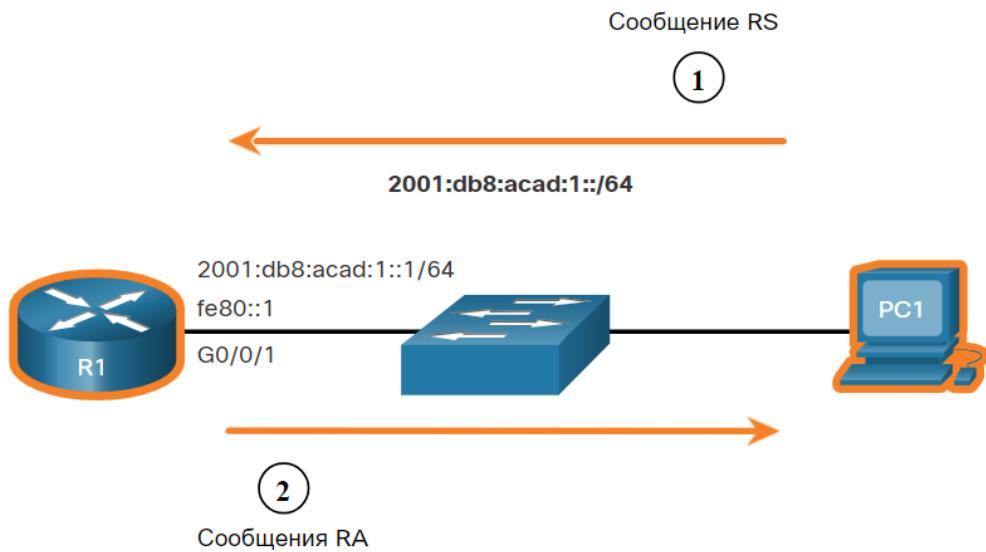
```
C:\PC1> ipconfig
Windows IP Configuration
Ethernet adapter Ethernet0:
  Connection-specific DNS Suffix  . :
  IPv6 Address . . . . . : 2001:db8:acad:1:1de9:c69:73ee:ca8c
  Link-local IPv6 Address . . . . : fe80::fb:1d54:839f:f595%21
  IPv4 Address . . . . . : 169.254.202.140
  Subnet Mask . . . . . : 255.255.0.0
  Default Gateway . . . . . : fe80::1%6
C:\PC1>
```

Сообщения RS ICMPv6

Маршрутизатор отправляет RA-сообщения каждые 200 секунд. Тем не менее, он также отправит сообщение RA, если он получает сообщение RS от хоста.

Когда клиент настроен на получение информации об адресации автоматически, он отправляет сообщение RS на адрес многоадресной рассылки всех маршрутизаторов IPv6 ff02::2.

На рисунке показано, как хост инициирует метод SLAAC.



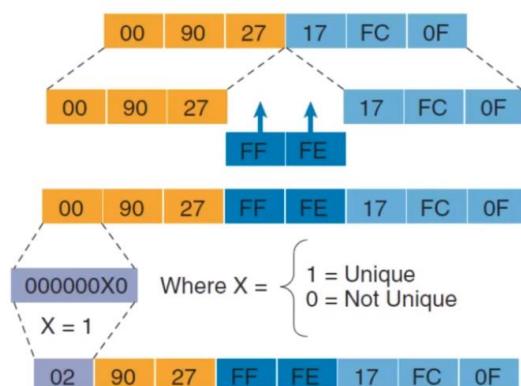
1. PC1 только что загрузился и еще не получил сообщение RA. Таким образом, он отправляет сообщение RS на адрес многоадресной рассылки IPv6 всех маршрутизаторов ff02::2 с запросом RA.
2. R1 входит в группу всех маршрутизаторов IPv6 и получает сообщение RS. Он генерирует RA, содержащий префикс локальной сети и длину префикса (например, 2001:db8:acad:1::/64). Затем он отправляет сообщение RA на адрес многоадресной рассылки всех узлов IPv6 ff02::1. PC1 использует эту информацию для создания уникального GUA IPv6.

Используя SLAAC, хост обычно получает информацию о 64-битной подсети IPv6 от маршрутизатора RA. Однако он должен генерировать оставшийся 64-битный идентификатор интерфейса (ID) одним из двух методов:

- **Генерация случайным образом** — 64-битный IID может быть случайным числом, сгенерированным операционной системой клиента. Этот метод теперь используется хостами Windows 10.
- **EUI-64** - хост создает идентификатор интерфейса, используя свой 48-битный MAC-адрес и вставляя шестнадцатеричное значение fffe в середине адреса. Некоторые операционные системы по умолчанию используют случайно сгенерированный идентификатор интерфейса вместо метода EUI-64, из-за проблем конфиденциальности. Это связано с тем, что MAC-адрес узла Ethernet используется EUI-64 для создания идентификатора интерфейса.

Примечание: Windows, Linux и Mac OS позволяют пользователю изменять генерирование идентификатора интерфейса либо случайным образом, либо использовать EUI-64.

Процесс (Extended Unique Identifier) EUI-64



Принцип работы SLAAC



Обнаружение дублирующихся адресов (DAD)

Этот процесс позволяет хосту создавать адрес IPv6. Однако нет гарантии, что адрес уникален в сети.

Поскольку SLAAC — это процесс без отслеживания состояния, компьютер PC1 должен проверить, что новый созданный IPv6-адрес является уникальным, прежде чем его использовать. Процесс обнаружения дубликатов адресов (DAD) используется хостом для обеспечения уникальности GUA IPv6.

DAD реализован с использованием ICMPv6. Чтобы выполнить DAD, хост отправляет сообщение ICMPv6 Neighbor Solicitation (NS) со специально созданным адресом многоадресной рассылки, который называется адресом многоадресной рассылки запрашиваемого узла. Этот адрес дублирует последние 24 бита IPv6-адреса узла.

Если другие устройства не отвечают сообщением с объявлением соседей, значит, практически гарантировано, что адрес является уникальным и может быть использован PC1. Если сообщение запроса поиска соседей получено PC1, значит, адрес не уникален и операционная система должна установить новый идентификатор интерфейса для использования.

Шаги работы DHCPv6

В этом разделе описывается DHCPv6 без сохранения состояния и с сохранением состояния. DHCPv6 без сохранения состояния использует части SLAAC для обеспечения того, чтобы вся необходимая информация была передана хосту. DHCPv6 с поддержкой состояния состояния не требует SLAAC.

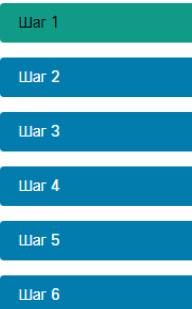
Несмотря на то что протокол DHCPv6 аналогичен DHCPv4 в своих функциональных возможностях, два протокола независимы друг от друга.

Если вариант работы DHCPv6 указан в сообщении RA, устройство начинает обмен данными по схеме клиент-сервер с использованием DHCPv6.

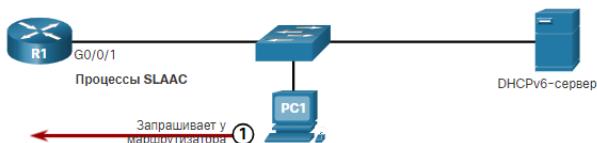
Сообщения DHCPv6 от сервера к клиенту используют UDP-порт назначения 546, а сообщения DHCPv6 от клиента к серверу используют UDP-порт назначения 547.

Шаги работы DHCPv6 заключаются в следующем:

1. Хост отправляет сообщение RS.
2. Маршрутизатор IPv6 отвечает сообщением RA.
3. Хост отправляет сообщение SOLICIT DHCPv6.
4. Сервер DHCPv6 отвечает сообщением DHCPv6 ADVERTISE.
5. Хост отвечает серверу DHCPv6.
6. Сервер DHCPv6 отправляет сообщение REPLY.

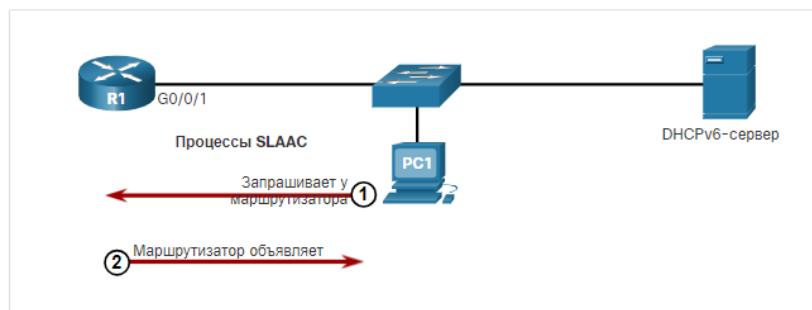


Шаг 1. Хост отправляет сообщение RS.
PC1 отправляет сообщение RS всем маршрутизаторам с поддержкой IPv6.



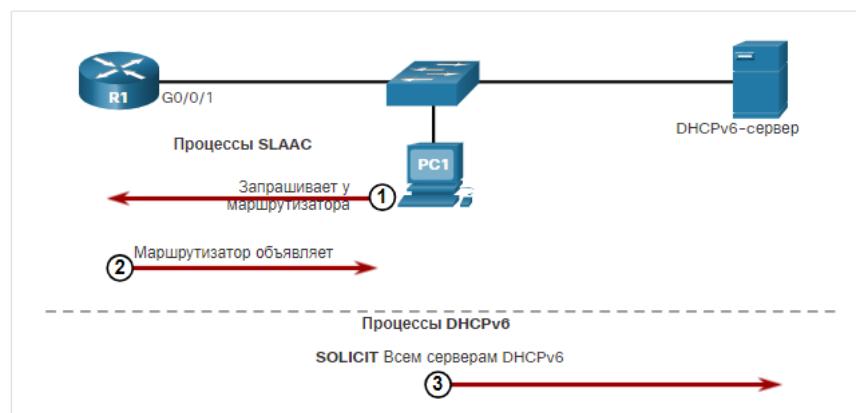
Шаг 2. Маршрутизатор IPv6 отвечает сообщением RA.

R1 получает RS и отвечает RA, указывая, что клиент должен инициировать связь с сервером DHCPv6.



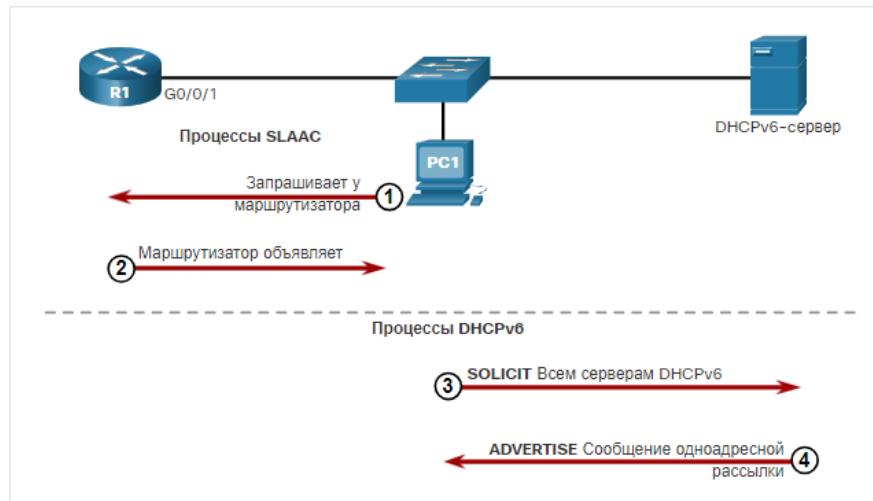
Шаг 3. Хост отправляет сообщение SOLICIT DHCPv6.

Клиент передает сообщение DHCPv6 SOLICIT на зарезервированный IPv6-адрес многоадресной рассылки FF02::1:2, используемый всеми DHCPv6 серверами. Этот адрес многоадресной рассылки действует в рамках канала link-local. Это означает, что маршрутизаторы не направляют сообщения в другие сети.



Шаг 4. Сервер DHCPv6 отвечает сообщением DHCPv6 ADVERTISE.

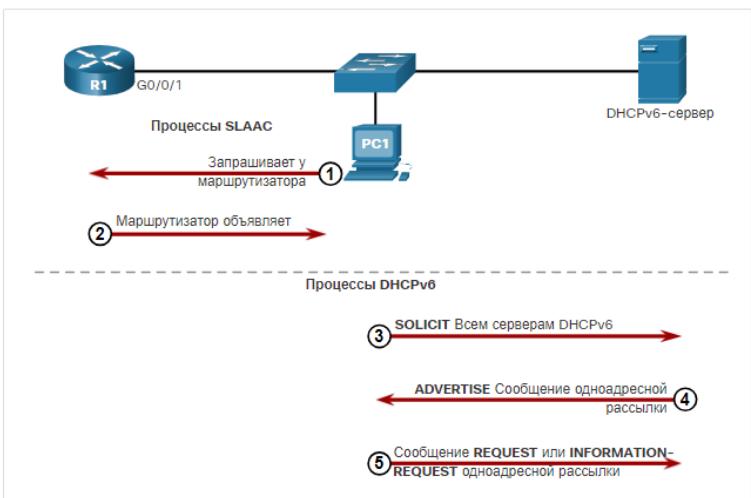
Один или несколько серверов DHCPv6 отвечают одноадресным DHCPv6-сообщением ADVERTISE. Сообщение ADVERTISE сообщает DHCPv6-клиенту, что сервер доступен для предоставления службы DHCPv6.



Шаг 5. Хост отвечает серверу DHCPv6.

Ответ PC1 зависит от того, использует ли он DHCPv6 с сохранением состояния или без состояния:

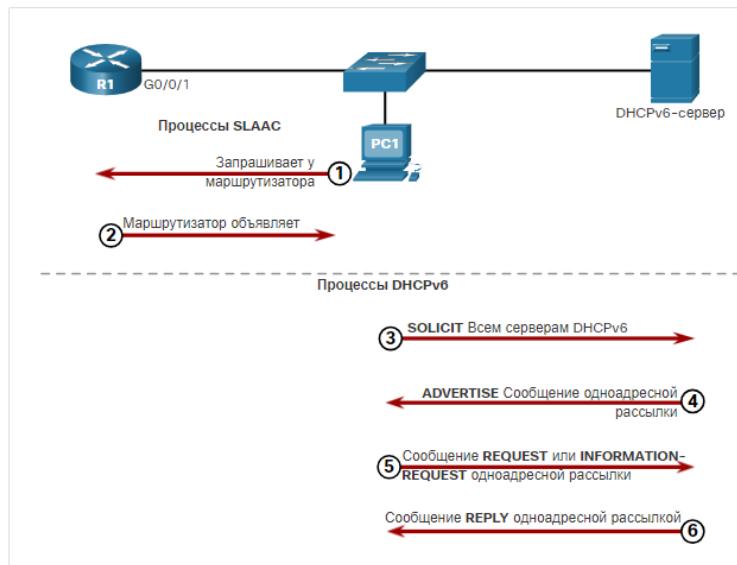
- **DHCPv6-клиент без сохранения состояния** – Клиент создает собственный IPv6-адрес при помощи префикса из сообщения RA и самогенерируемого идентификатора интерфейса. Клиент отправляет DHCPv6 сообщение INFORMATION-REQUEST серверу DHCPv6, запрашивая только параметры конфигурации, например, адрес DNS-сервера.
- **DHCPv6-клиент с отслеживанием состояния** – клиент отправляет DHCPv6 сообщение REQUEST серверу для получения IPv6-адреса и всех остальных параметров конфигурации от сервера.



Шаг 6. Сервер DHCPv6 отправляет сообщение REPLY.

Сервер отправляет клиенту одноадресные сообщения DHCPv6 REPLY. Содержимое сообщения зависит от того, отвечает ли оно на сообщение REQUEST или INFORMATION-REQUEST.

Примечание: Клиент будет использовать исходный IPv6 Link -локальный адрес RA в качестве адреса шлюза по умолчанию . Сервер DHCPv6 не предоставляет эту информацию.

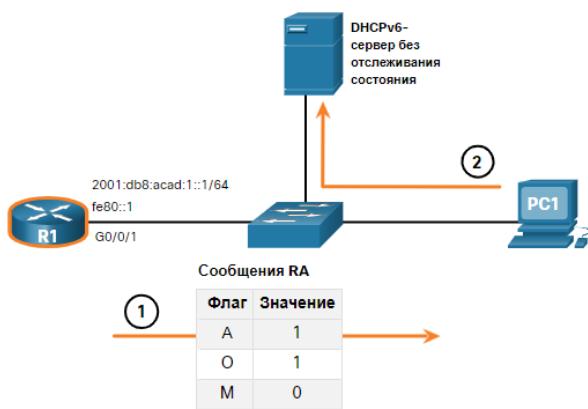


Операция DHCPv6 без сохранения состояния

DHCPv6 без отслеживания состояния сообщает клиенту об использовании информации в сообщении RA для адресации, при этом дополнительные параметры конфигурации доступны с сервера DHCPv6.

Этот процесс известен как протокол DHCPv6 без отслеживания состояния, поскольку сервер не поддерживает никакую информацию о состоянии клиента, то есть список доступных и распределенных IPv6-адресов. DHCPv6-серверы без отслеживания состояния предоставляют только параметры конфигурации для клиента, но не выделяют IPv6-адреса.

На рисунке показана операция DHCPv6 без учета состояния.



1. PC1 получает сообщение DHCP RA без состояния. Сообщение RA содержит префикс сети и длину префикса. Флаг M для DHCP с сохранением состояния имеет значение по умолчанию 0. Флаг A=1 указывает клиенту использовать SLAAC. Значение флага O, равное 1, используется для информирования клиента о том, что на DHCPv6-сервере без отслеживания состояния доступна дополнительная информация о конфигурации.

2. Клиент отправляет сообщение DHCPv6 SOLICIT в поисках сервера DHCPv6 без состояния для получения дополнительной информации (например, адреса DNS-серверов).

Настройка использования Stateless DHCP

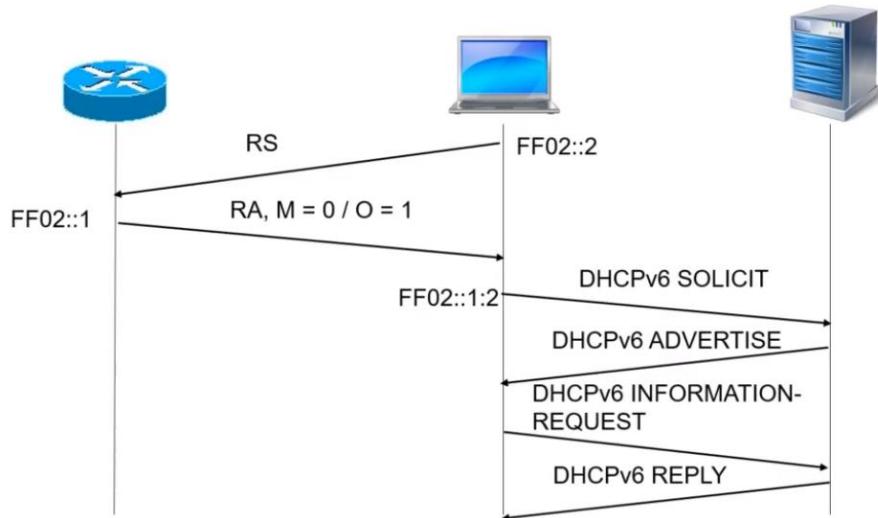
Установка флага O

```
R(config-if)# ipv6 nd other-config-flag
```

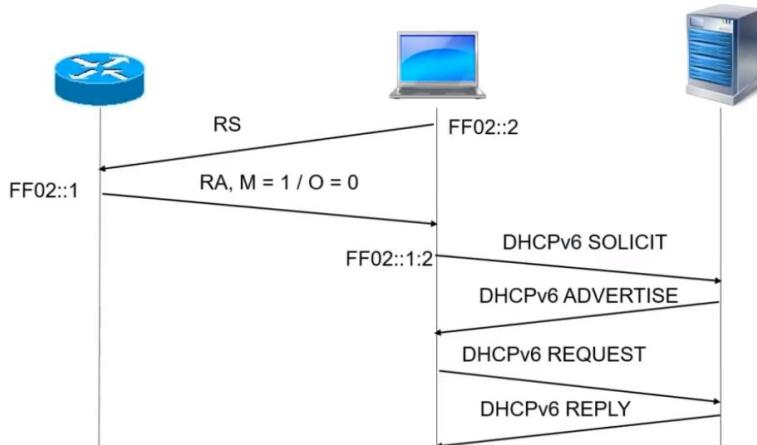
```
R1(config-if)# ipv6 nd other-config-flag
R1(config-if)# end
R1#
R1# show ipv6 interface g0/0/1 | begin ND
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds (using 30000)
ND advertised reachable time is 0 (unspecified)
ND advertised retransmit interval is 0 (unspecified)
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
ND advertised default router preference is Medium
Hosts use stateless autoconfig for addresses.
Hosts use DHCP to obtain other configuration.
R1#
```

24

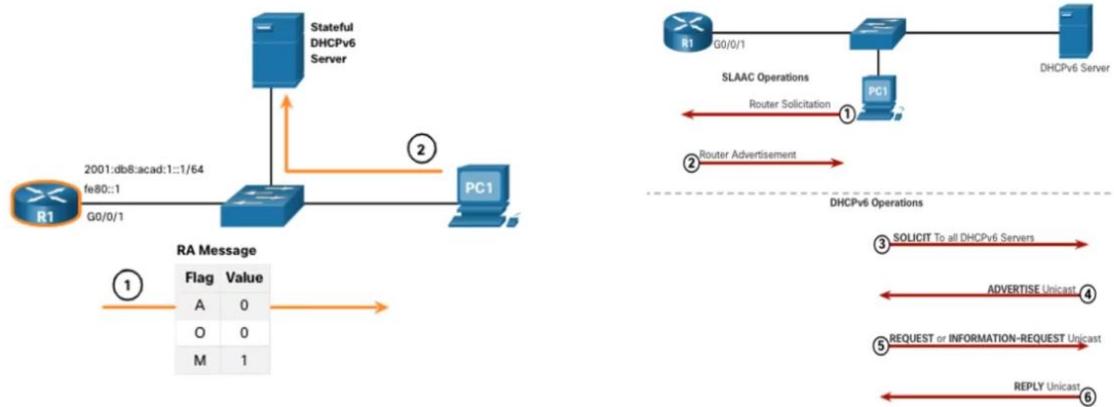
Stateless DHCP



Stateful DHCP



Stateful DHCP



Настройка использования Stateful DHCP

Установка флага O

```
R(config-if)# ipv6 nd managed-config-flag
```

```
R1(config)# int g0/0/1
R1(config-if)# ipv6 nd managed-config-flag
R1(config-if)# end
R1#
R1# show ipv6 interface g0/0/1 | begin ND
    ND DAD is enabled, number of DAD attempts: 1
    ND reachable time is 30000 milliseconds (using 30000)
    ND advertised reachable time is 0 (unspecified)
    ND advertised retransmit interval is 0 (unspecified)
    ND router advertisements are sent every 200 seconds
    ND router advertisements live for 1800 seconds
    ND advertised default router preference is Medium
    Hosts use DHCP to obtain routable addresses.
R1#
```

Настройка Stateless DHCPv6-сервера

Включить IPv6 маршрутизацию

```
R(config)# ipv6 unicast-routing
```

Создать пул DHCP

```
R(config)# ipv6 dhcp pool POOL_NAME
```

Настроить параметры

```
R(config-dhcpv6)# dns-server ipv6_address
R(config-dhcpv6)# domain-name name
```

Привязать пул к интерфейсу и настроить флаг О

```
R(config-if)# ipv6 dhcp server POOL_NAME
R(config-if)# ipv6 nd other-config-flag
```

Пример

```
R1(config)# ipv6 unicast-routing
R1(config)#
R1(config)# ipv6 dhcp pool IPV6-STATELESS
R1(config-dhcpv6)#
R1(config-dhcpv6)# dns-server 2001:db8:acad:1::254
R1(config-dhcpv6)# domain-name example.com
R1(config-dhcpv6)# exit
R1(config)#
R1(config)# interface GigabitEthernet0/0/1
R1(config-if)# description Link to LAN
R1(config-if)# ipv6 address fe80::1 link-local
R1(config-if)# ipv6 address 2001:db8:acad:1::1/64
R1(config-if)# ipv6 nd other-config-flag
R1(config-if)# ipv6 dhcp server IPV6-STATELESS
R1(config-if)# no shut
```

Проверка на хосте

```
C:\PC1> ipconfig /all
Windows IP Configuration
Ethernet adapter Ethernet0:
  Connection-specific DNS Suffix . : example.com
  Description . . . . . : Intel(R) 82574L Gigabit Network Connection
  Physical Address . . . . . : 00-05-9A-3C-7A-00
  DHCP Enabled . . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes
  IPv6 Address . . . . . : 2001:db8:acad:1:1de9:c69:73ee:ca8c (Preferred)
  Link-local IPv6 Address . . . . . : fe80::fb:1d54:839f:f595%21(Preferred)
  IPv4 Address . . . . . : 169.254.102.23 (Preferred)
  Subnet Mask . . . . . : 255.255.0.0
  Default Gateway . . . . . : fe80::1%6
  DHCPv6 IAID . . . . . : 318768538
  DHCPv6 Client DUID . . . . . : 00-01-00-01-21-F3-76-75-54-E1-AD-DE-DA-9A
  DNS Servers . . . . . : 2001:db8:acad:1::254
  NetBIOS over Tcpip. . . . . : Enabled
```

Настройка Stateless DHCPv6-клиента

Включить IPv6 маршрутизацию

```
R(config)# ipv6 unicast-routing
```

```
R3(config)# ipv6 unicast-routing
R3(config)#
```

Настроить LLA

```
R(config-if)# ipv6 enable
```

```
R3(config)# interface g0/0/1
R3(config-if)# ipv6 enable
```

Настроить SLAAC

```
R(config-if)# ipv6 address autoconfig
```

```
R3(config-if)# ipv6 address autoconfig
R3(config-if)# end
```

Проверка

```
R3# show ipv6 interface brief
GigabitEthernet0/0/0    [up/up]
  unassigned
GigabitEthernet0/0/1    [up/up]
  FE80::2FC:BAFF:FE94:29B1
  2001:DB8:ACAD:1:2FC:BAFF:FE94:29B1
Serial0/1/0              [up/up]
  unassigned
Serial0/1/1              [up/up]
  unassigned
```

```
R3# show ipv6 dhcp interface g0/0/1
GigabitEthernet0/0/1 is in client mode
Prefix State is IDLE (0)
Information refresh timer expires in 23:56:06
Address State is IDLE
List of known servers:
  Reachable via address: FE80::1
  DUID: 000300017079B3923640
  Preference: 0
  Configuration parameters:
    DNS server: 2001:DB8:ACAD:1::254
    Domain name: example.com
    Information refresh time: 0
  Prefix Rapid-Commit: disabled
  Address Rapid-Commit: disabled
```

Настройка Stateful DHCPv6-сервера

Включить IPv6 маршрутизацию

```
R(config)# ipv6 unicast-routing
```

Создать пул DHCP

```
R(config)# ipv6 dhcp pool POOL_NAME
```

Настроить параметры

```
R(config-dhcpv6)# address prefix prefix/prefix_length
R(config-dhcpv6)# dns-server ipv6_address
R(config-dhcpv6)# domain-name name
```

Привязать пул к интерфейсу и настроить флаг M

```
R(config-if)# ipv6 dhcp server POOL_NAME
R(config-if)# ipv6 nd managed-config-flag
R(config-if)# ipv6 nd prefix default no-autoconfig
```

Пример

```
R1(config)# ipv6 unicast-routing
R1(config)#
R1(config)# ipv6 dhcp pool IPV6-STATEFUL
R1(config-dhcpv6)#
R1(config-dhcpv6)# address prefix 2001:db8:acad:1::/64
R1(config-dhcpv6)# dns-server 2001:4860:4860::8888
R1(config-dhcpv6)# domain-name example.com
R1(config)# interface GigabitEthernet0/0/1
R1(config-if)# description Link to LAN
R1(config-if)# ipv6 address fe80::1 link-local
R1(config-if)# ipv6 address 2001:db8:acad:1::1/64
R1(config-if)# ipv6 nd managed-config-flag
R1(config-if)# ipv6 nd prefix default no-autoconfig
R1(config-if)# ipv6 dhcp server IPV6-STATEFUL
R1(config-if)# no shut
```

Настройка Stateful DHCPv6-клиента

Включить IP-маршрутизацию

```
R(config)# ipv6 unicast-routing
```

```
R3(config)# ipv6 unicast-routing  
R3(config)#
```

Настройте LLA

```
R(config-if)# ipv6 enable
```

```
R3(config)# interface g0/0/1  
R3(config-if)# ipv6 enable
```

Настроить использование DHCPv6

```
R(config-if)# dhcp ipv6 address
```

```
R3(config-if)# ipv6 address dhcp  
R3(config-if)# end
```

Проверка

```
R3# show ipv6 interface brief
GigabitEthernet0/0/0    [up/up]
  unassigned
GigabitEthernet0/0/1    [up/up]
  FE80::2FC:BAFF:FE94:29B1
  2001:DB8:ACAD:1:B4CB:25FA:3C9:747C
Serial0/1/0              [up/up]
  unassigned
Serial0/1/1              [up/up]
  unassigned
```

```
R3# show ipv6 dhcp interface g0/0/1
GigabitEthernet0/0/1 is in client mode
Prefix State is IDLE
Address State is OPEN
Renew for address will be sent in 11:56:33
List of known servers:
  Reachable via address: FE80::1
  DUID: 00030001707983923640
  Preference: 0
Configuration parameters:
  IA NA: IA ID 0x00060001, T1 43200, T2 69120
  Address: 2001:DB8:ACAD:1:B4CB:25FA:3C9:747C/128
  preferred lifetime 86400, valid lifetime 172800
  expires at Sep 29 2019 11:52 AM (172593 seconds)
  DNS server: 2001:4860:4860::8888
  Domain name: example.com
  Information refresh time: 0
  Prefix Rapid-Commit: disabled
  Address Rapid-Commit: disabled
```

Проверка работы сервера

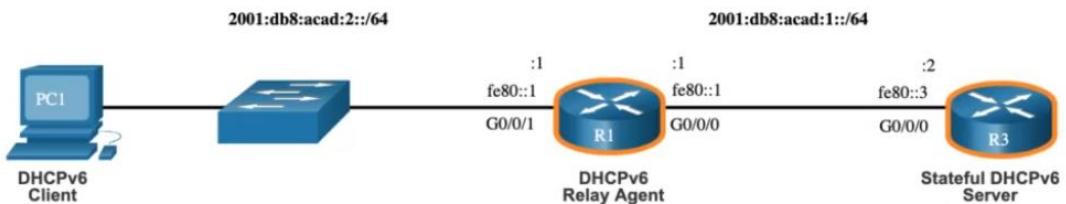
show ipv6 dhcp pool

```
R1# show ipv6 dhcp pool
DHCPv6 pool: IPV6-STATEFUL
  Address allocation prefix: 2001:DB8:ACAD:1::/64 valid 172800 preferred 86400 (2 in use, 0
conflicts)
  DNS server: 2001:4860:4860::8888
  Domain name: example.com
  Active clients: 2
R1#
```

show ipv6 dhcp binding

```
R1# show ipv6 dhcp binding
Client: FE80::192F:6FBC:9DB:B749
  DUID: 0001000125148183005056B327D6
  Username : unassigned
  VRF : default
  IA NA: IA ID 0x03000C29, T1 43200, T2 69120
  Address: 2001:DB8:ACAD:1:A43C:FD28:9D79:9E42
    preferred lifetime 86400, valid lifetime 172800
    expires at Sep 27 2019 09:10 AM (171192 seconds)
Client: FE80::2FC:BAFF:FE94:29B1
  DUID: 0003000100FCBA9429B0
  Username : unassigned
  VRF : default
  IA NA: IA ID 0x00060001, T1 43200, T2 69120
  Address: 2001:DB8:ACAD:1:B4CB:25FA:3C9:747C
    preferred lifetime 86400, valid lifetime 172800
    expires at Sep 27 2019 09:29 AM (172339 seconds)
R1#
```

Настройка ретрансляции



```
R1(config)# interface gigabitethernet 0/0/1
R1(config-if)# ipv6 dhcp relay destination 2001:db8:acad:1::2 G0/0/0
R1(config-if)# exit
R1(config)#
```

Проверка

```
R1# show ipv6 dhcp interface
GigabitEthernet0/0/1 is in relay mode
Relay destinations:
 2001:DB8:ACAD:1::2
 2001:DB8:ACAD:1::2 via GigabitEthernet0/0/0
R1#
```

```
R3# show ipv6 dhcp binding
Client: FE80::5C43:EE7C:2959:DA68
DUID: 0001000124F5CEA2005056B3636D
Username : unassigned
VRF : default
IA NA: IA ID 0x03000C29, T1 43200, T2 69120
Address: 2001:DB8:ACAD:2:9C3C:64DE:AADA:7857
  preferred lifetime 86400, valid lifetime 172800
  expires at Sep 29 2019 08:26 PM (172710 seconds)
R3#
```

с. Использование сообщений ICMPv6 для работы IPv6.

Несмотря на то, что IP является протоколом наибольших усилий, пакет TCP/IP обеспечивает сообщения об ошибках и информационные сообщения при взаимодействии с другим IP-устройством. Эти сообщения отправляются с помощью ICMP-сервисов. Назначение таких сообщений — предоставить обратную связь о проблемах, связанных с обработкой IP-пакетов в определенных условиях, а не повышать надежность протокола IP. Из соображений безопасности сообщения ICMP не обязательны и часто даже не разрешены в сети.

ICMP может использоваться как с IPv4, так и с IPv6. ICMPv4 — это протокол обмена сообщениями для IPv4. Протокол ICMPv6 предоставляет те же сервисы для IPv6, но при этом включает в себя дополнительные функциональные возможности. В рамках данного курса термин ICMP будет использоваться для обозначения как ICMPv4, так и ICMPv6.

Существует множество типов ICMP-сообщений и причин их отправки. Сообщения ICMP, общие для ICMPv4 и ICMPv6 и обсуждаемые в этом модуле, включают:

- Достигимость узла
- Узел назначения или сервис недоступен
- Превышен интервал ожидания

Доступность узла

Эхо-сообщение ICMP можно использовать для проверки доступности узла в IP-сети. Локальный узел отправляет узлу эхо-запрос ICMP. Если узел доступен, узел назначения отправляет эхо-ответ. Такое использование ping-запросов по протоколу ICMP легло в основу утилиты **ping**.

Узел назначения или сервис недоступен

Когда узел или шлюз получает пакет, который не может доставить, он может использовать ICMP-сообщение «Узел назначения недоступен» (Destination Unreachable), чтобы сообщить источнику о том, что узел назначения или сервис для этого пакета недоступен. Такое сообщение содержит код, определяющий причину, по которой пакет не может быть доставлен.

Примеры некоторых кодов сообщений о недоступном узле назначения для ICMPv4:

- 0 — сеть недоступна;
- 1 — узел недоступен;
- 2 — протокол недоступен;
- 3 — порт недоступен.

Примеры некоторых кодов сообщений о недоступном узле назначения для ICMPv6:

- 0 - нет маршрута до пункта назначения
- 1 - Связь с пунктом назначения административно запрещена (например, брандмауэр)
- 2 — За пределами области адреса источника
- 3 - Адрес недоступен
- 4 — порт недоступен.

Превышен интервал ожидания

Сообщения ICMPv4 о превышении интервала ожидания (Time Exceeded) используется маршрутизатором для указания на то, что пакет невозможно переслать, поскольку значение в поле «Время существования» (Time to Live, TTL) пакета было изменено на 0. Если маршрутизатор получает пакет и изменяет значение в поле TTL IPv4-пакета на нуль, он отбрасывает пакет и отправляет на исходный узел сообщение о превышении интервала ожидания.

Протокол ICMPv6 также отправляет сообщение о превышении интервала ожидания, в случае если маршрутизатор не может переслать IPv6-пакет из-за истечения его срока действия. В протоколе IPv6 поле TTL отсутствует; чтобы выяснить, не истек ли срок действия пакета, используется поле «предел переходов» (hop limit).

Примечание: Инструмент **traceroute** использует сообщения о превышении времени.

Сообщения ICMPv6

Информационные сообщения и сообщения об ошибках, возникающие в протоколе ICMPv6, очень похожи на сообщения о контроле и ошибках, используемые протоколом ICMPv4. Однако протокол ICMPv6 отличается расширенной функциональностью и новыми возможностями, которых нет в ICMPv4. Сообщения ICMPv6 инкапсулируются в IPv6-пакеты.

ICMPv6 включает четыре новых протокола в составе протокола обнаружения соседних узлов (Neighbor Discovery Protocol, ND или NDP).

Обмен сообщениями между маршрутизатором IPv6 и устройством IPv6, включая динамическое распределение адресов, осуществляется следующим образом:

- Сообщение «Запрос к маршрутизатору» (Router Solicitation, RS)
- Сообщение «Ответ маршрутизатора» (Router Advertisement, RA)

Обмен сообщениями между устройствами IPv6, включая обнаружение повторяющихся адресов и разрешение адресов, осуществляется следующим образом:

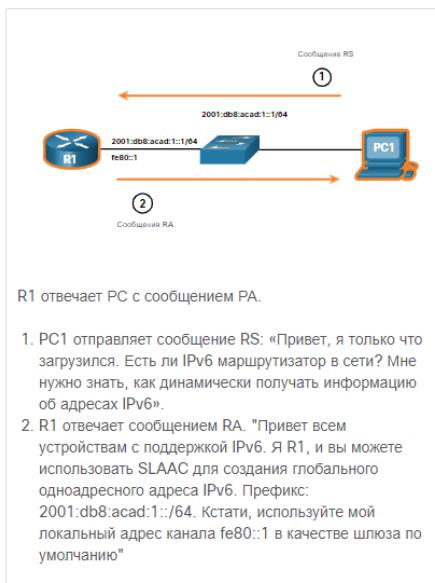
- Сообщение с запросом поиска соседей (NS)
- Сообщение об объявлении соседних узлов (NA)

Сообщения RA отправляются маршрутизаторами с поддержкой IPv6 каждые 200 секунд для предоставления информации об адресации узлам с поддержкой IPv6. Сообщение RA может включать такие данные об адресах для хостов, как префикс, длина префикса, DNS-адрес и доменное имя. Узел, использующий SLAAC, установит в качестве своего шлюза по умолчанию локальный адрес канала маршрутизатора, отправившего RA.



R1 отправляет сообщение RA: "Привет всем устройствам с поддержкой IPv6. Я R1, и вы можете использовать SLAAC для создания глобального одноадресного адреса IPv6. Префикс: 2001:db8:acad:1::/64. Кстати, используйте мой локальный адрес канала fe80::1 в качестве шлюза по умолчанию."

Маршрутизатор с поддержкой IPv6 также отправит сообщение RA в ответ на сообщение RS. На рисунке PC1 отправляет сообщение RS, чтобы определить, как получать информацию об адресах IPv6 динамически.



R1 отвечает PC с сообщением RA.

- PC1 отправляет сообщение RS: «Привет, я только что загрузился. Есть ли IPv6 маршрутизатор в сети? Мне нужно знать, как динамически получать информацию об адресах IPv6».
- R1 отвечает сообщением RA. «Привет всем устройствам с поддержкой IPv6. Я R1, и вы можете использовать SLAAC для создания глобального одноадресного адреса IPv6. Префикс: 2001:db8:acad:1::/64. Кстати, используйте мой локальный адрес канала fe80::1 в качестве шлюза по умолчанию»

Сообщения RA

Сообщение RS

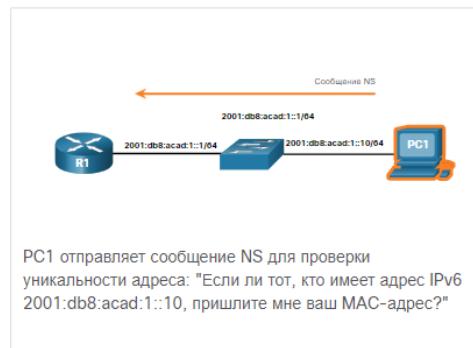
Сообщение NS

Сообщение NA

Когда устройству назначается глобальный одноадресный IPv6-адрес или одноадресный локальный адрес канала, оно может выполнить обнаружение дублированного адреса (DAD), чтобы гарантировать, что адрес IPv6 является уникальным. Для проверки уникальности адреса устройство отправляет сообщение NS с собственным IPv6-адресом в качестве целевого, как показано на рисунке.

Если другому устройству в сети присвоен этот адрес, оно ответит сообщением NA. Это сообщение NA уведомляет устройство-отправителя о том, что данный адрес уже используется. Если соответствующее сообщение NA не возвращается в течение определенного периода времени, индивидуальный адрес признается уникальным и допустимым к использованию.

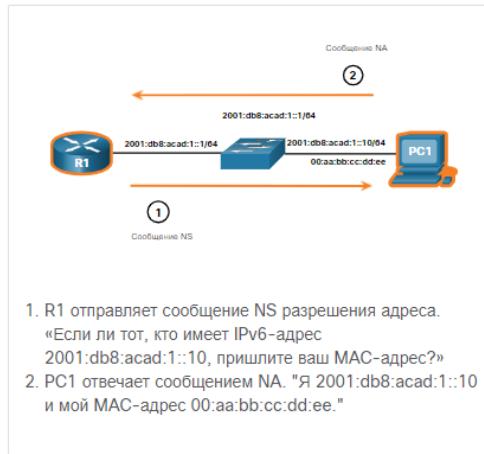
Примечание: Процесс обнаружения дублирующихся адресов не обязательен. Однако, документ RFC 4861 рекомендует выполнять его для индивидуальных адресов.



PC1 отправляет сообщение NS для проверки уникальности адреса: "Если ли тот, кто имеет адрес IPv6 2001:db8:acad:1::10, пришлите мне ваш MAC-адрес?"

Протокол разрешения адресов используется в том случае, когда устройству в локальной сети (LAN) известен индивидуальный IPv6-адрес назначения, но неизвестен MAC-адрес Ethernet. Для того чтобы определить MAC-адрес назначения, устройство отправляет сообщение NS на адрес запрашиваемого узла. Сообщение включает известный (целевой) IPv6-адрес. Устройство с целевым IPv6-адресом отправляет в ответ сообщение NA, содержащее его MAC-адрес Ethernet.

На рисунке R1 отправляет сообщение NS в 2001:db8:acad:1::10 с запросом его MAC-адреса.



7. Управление оборудованием:

а. Протоколы обнаружения устройств: определение, принцип работы, характеристики.

Первое, что вы хотите знать о вашей сети, это то, что в ней? Где эти компоненты? Как они связаны? В принципе, вам нужна карта. В этом разделе объясняется, как можно использовать протокол Cisco Discovery Protocol (CDP) для создания карты сети.

Протокол Cisco Discovery Protocol (CDP) — это проприетарный протокол компании Cisco уровня 2, который служит для сбора информации об устройствах Cisco, использующих один и тот же канал передачи данных. CDP не зависит от среды передачи данных и других протоколов; он включен на всех устройствах Cisco, таких как маршрутизаторы, коммутаторы и серверы доступа.

Периодически устройство отправляет объявления CDP на подключенные устройства, как показано на рисунке.



Посредством объявлений осуществляется обмен информацией о типах обнаруженных устройств, их именах, количестве и типах интерфейсов.

Поскольку большинство сетевых устройств подключены к другим устройствам, протокол CDP может помочь в проектировании сетей, поиске и устранении неполадок, а также во внесении изменений в оборудование. CDP также можно использовать в качестве сетевого средства обнаружения для получения информации о соседних устройствах. Собранная таким образом информация помогает построить логическую топологию сети в случае отсутствия документации или ее недостаточной детализации.

Cisco Discovery Protocol (CDP)

- Служит для сбора информации об устройствах, использующих один и тот же канал передачи данных
- Проприетарный Cisco-протокол второго уровня
- Использует MAC-адрес **01-00-0c-cc-cc-cc**
- Рассылка каждые 60с. в сетях Ethernet, Frame Relay и ATM
- Просмотр таблицы:
 - `show cdp neighbors [detail]`
 - `show cdp entry {* | device-name} [protocol | version]`
- Dead-timeout 180с.



Проприетарный – разработанный конкретным поставщиком.

Работает следующим образом: отправляются cdp сообщения на выделенный мак адрес на все включенные интерфейсы. Отправляются каждые 60 секунд. Проверяется доступность устройств на КАНАЛЬНОМ уровне.

Заголовок CDP

00 биты	07 биты	08 биты	15 биты	16 биты	23 биты	24 биты	31
Версия	TTL	Контрольная сумма					
Тип		Длина					
Данные							

Данные

- Идентификатор устройства;
- Номер и тип локального интерфейса;
- Время удержания информации (время, по истечению которого записи из CDP-таблицы удаляются);
- Тип устройства (маршрутизатор, коммутатор, сетевой мост и т.д.);
- Физическую платформу устройства (модель подключенного устройства);
- Номер и тип удаленного интерфейса;
- Доменное имя VTP;
- Номер собственной сети;
- Информация о дуплексности

Пакет в wireshark

```
Cisco Discovery Protocol
Version: 2
TTL: 180 seconds
Checksum: 0xc2c3
Device ID: LAN354802
    Type: Device ID (0x0001)
    Length: 13
    Device ID: LAN354802
Addresses Type:
    Addresses (0x0002)
    Length: 17
    Number of addresses: 1
    IP address: 192.168.2.62
        Protocol type: NLPID
        Protocol length: 1
        Protocol: IP
        Address length: 4
        IP address: 192.168.2.62
Port ID: FastEthernet0/7
    Type: Port ID (0x0003)
    Length: 19
    Sent through Interface: FastEthernet0/7
```

Настройка и проверка CDP

На устройствах Cisco протокол CDP включен по умолчанию. Иногда из соображений безопасности может потребоваться отключить CDP на сетевом устройстве — глобально или для отдельных интерфейсов. Когда CDP включен, злоумышленник может получить ценную информацию о структуре сети, такую как IP-адреса, версии IOS и типы устройств.

Чтобы проверить состояние CDP и отобразить сведения о нем, введите команду **show cdp**, как показано в примере.

```
Router# show cdp
Global CDP information:
    Sending CDP packets every 60 seconds
    Sending a holdtime value of 180 seconds
    Sending CDPv2 advertisements is enabled
```

Чтобы включить CDP сразу для всех поддерживаемых интерфейсов устройства, введите команду **cdp run** в режиме глобальной конфигурации. Чтобы отключить CDP сразу для всех интерфейсов устройства, введите команду **no cdp run** в режиме глобальной конфигурации.

```
Router(config)# no cdp run
Router(config)# exit
Router# show cdp
CDP is not enabled
Router# configure terminal
Router(config)# cdp run
```

Отключение CDP

```
R(config)# no cdp run  
R(config-if)# no cdp enable
```

Чтобы отключить CDP для определенного интерфейса, например используемого для подключения к интернет-провайдеру, введите **no cdp enable** режиме настройки интерфейса. Протокол CDP по-прежнему включен на устройстве, однако объявления CDP больше не передаются через этот интерфейс. Чтобы снова включить CDP для конкретного интерфейса, введите **cdp enable**, как показано в примере.

```
Switch(config)# interface gigabitethernet 0/0/1  
Switch(config-if)# cdp enable
```

Чтобы проверить состояние CDP и просмотреть список соседних устройств, выполните команду **show cdp neighbors** в привилегированном режиме EXEC. Команда **show cdp neighbors** отображает важную информацию о соседних устройствах CDP. Как видно из результатов выполнения команды **show cdp neighbors**, которые показаны в примере, в настоящее время у данного устройства нет соседей, поскольку оно физически не подключено к другим устройствам.

```
Router# show cdp neighbors  
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge  
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,  
                  D - Remote, C - CVTA, M - Two-port Mac Relay  
  
Device ID        Local Intrfce     Holdtme   Capability Platform Port ID  
  
Total cdp entries displayed : 0
```

Команда **show cdp interface** отображает интерфейсы устройства, на которых включен протокол CDP. Кроме того, выводится состояние каждого интерфейса. На рисунке показано, что протокол CDP включен на пяти интерфейсах маршрутизатора, при этом имеется только одно активное подключение к другому устройству.

```

Router# show cdp interface
GigabitEthernet0/0/0 is administratively down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0/0/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0/0/2 is down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial0/1/0 is administratively down, line protocol is down
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
Serial0/1/1 is administratively down, line protocol is down
  Encapsulation HDLC
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0 is down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
  cdp enabled interfaces : 6
    interfaces up       : 1
    interfaces down     : 5

```

Поиск устройств с помощью CDP

Предположим, что документация по топологии, показанной на рисунке, отсутствует. Администратор сети знает только, что R1 подключен к другому устройству.

Если в сети включен протокол CDP, структуру сети можно определить с помощью команды **show cdp neighbors**

```

R1# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID      Local Intrfce     Holdtme   Capability Platform Port ID
S1            Gig 0/0/1        179        S I       WS-C3560- Fas 0/5

```

Нет никакой информации об остальной части сети. Команда **show cdp neighbors** отображает полезную информацию о каждом соседнем устройстве CDP, в том числе следующие данные:

- **Идентификатор устройства** — имя хоста соседнего устройства (S1).
- **Идентификатор порта** — имя локального или удаленного порта (Gig 0/0/1 и Fas 0/0/5 соответственно).
- **Список возможностей** — сведения о том, является ли устройство маршрутизатором или коммутатором (S обозначает коммутатор; I обозначает IGMP и в данном курсе не рассматривается).

- **Платформа** — аппаратная платформа устройства (WS-C3560 обозначает коммутатор Cisco 3560).

Выходные данные показывают, что к интерфейсу G0/0/1 на R1 подключено другое устройство Cisco S1. Кроме того, S1 подключается через F0/5, как показано в обновленной топологии.



Сетевой администратор использует **show cdp neighbors detail** для обнаружения IP-адреса S1. Как показано в выходных данных, адрес S1 — 192.168.1.2.

```

R1# show cdp neighbors detail
-----
Device ID: S1
Entry address(es):
  IP address: 192.168.1.2
Platform: cisco WS-C3560-24TS,  Capabilities: Switch IGMP
Interface: GigabitEthernet0/0/1,  Port ID (outgoing port): FastEthernet0/5
Holdtime : 136 sec

Version :
Cisco IOS Software, C3560 Software (C3560-LANBASEK9-M), Version 15.0(2)SE7, R
RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2014 by Cisco Systems, Inc.
Compiled Thu 23-Oct-14 14:49 by prod_rel_team

advertisement version: 2
Protocol Hello: OUI=0x00000C, Protocol ID=0x0112; payload len=27,
value=00000000FFFFFFFFFF010221FF000000000000002291210380FF0000
VTP Management Domain: ''
Native VLAN: 1
Duplex: full
Management address(es):
  IP address: 192.168.1.2

Total cdp entries displayed : 1
  
```

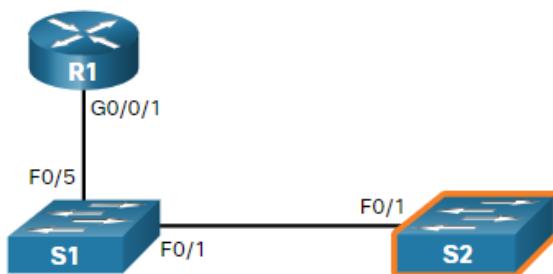
Получив удаленный доступ к коммутатору S1 через подключение SSH либо физический доступ через консольный порт, сетевой администратор может узнать, какие еще устройства подключены к S1, с помощью команды **show cdp neighbors**. На рисунке показан результат выполнения этой команды.

```

S1# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay
Device ID      Local Intrfce     Holdtme    Capability  Platform  Port ID
S2              Fas 0/1          150         S I        WS-C2960- Fas 0/1
R1              Fas 0/5          179         R S I     ISR4331/K Gig 0/0/1

```

В выходных данных появляется еще один коммутатор – S2. S2 использует F0/1 для подключения к интерфейсу F0/1 на S1, как показано на рисунке.



Кроме того, администратор сети может использовать **show cdp neighbors detail** для обнаружения IP-адреса для S2, а затем удаленного доступа к нему. После успешного входа в систему администратор сети использует команду **show cdp neighbors**, чтобы определить, есть ли больше устройств.

```

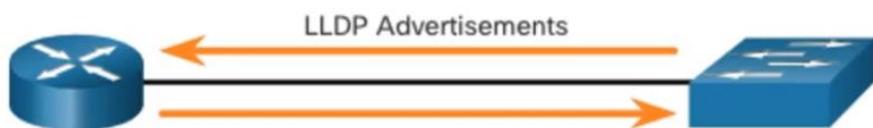
S2# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay
Device ID      Local Intrfce     Holdtme    Capability  Platform  Port ID
S1              Fas 0/1          141         S I        WS-C3560- Fas 0/1

```

К коммутатору S2 подключено только одно устройство – коммутатор S1. Следовательно, в топологии больше нет доступных для обнаружения устройств. Сетевой администратор может обновить документацию сети, указав в ней обнаруженные устройства.

Link Layer Discovery Protocol (LLDP)

- Не зависящий от производителя протокол второго уровня
- Протокол обнаружения соседей



Протокол обнаружения уровня канала (LLDP) делает то же самое, что и CDP, но он не специфичен для устройств Cisco. В качестве бонуса вы можете использовать его, если у вас есть устройства Cisco. Так или иначе, вы получите карту сети.

LLDP - не зависящий от производителя протокол обнаружения соседей, подобный CDP. LLDP работает с сетевыми устройствами, такими как маршрутизаторы, коммутаторы и точки доступа к беспроводной сети LAN. Этот протокол объявляет себя и свои возможности другим устройствам и получает данные от физически подключенных устройств уровня 2.

Настройка и проверка протокола LLDP

На некоторых устройствах протокол LLDP может быть включен по умолчанию. Чтобы включить LLDP для всех интерфейсов сетевого устройства Cisco, введите команду **lldp run** в режиме глобальной конфигурации. Чтобы отключить протокол LLDP, введите команду **no lldp run** в режиме глобальной конфигурации.

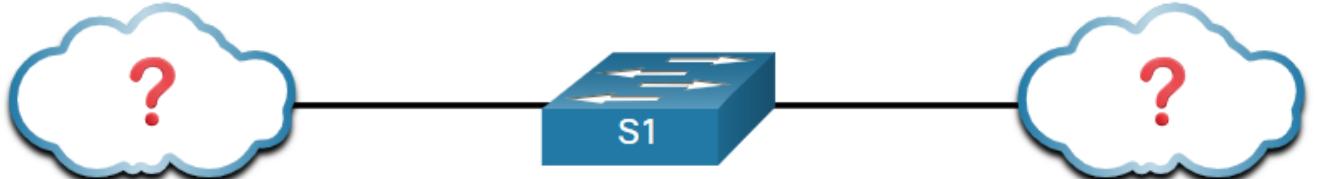
Как и протокол CDP, протокол LLDP можно включить и отключить на конкретных интерфейсах. Однако передачу и прием пакетов LLDP необходимо настраивать отдельно, как показано на рисунке.

Чтобы убедиться в том, что протокол LLDP был включен на устройстве, введите команду **show lldp** в привилегированном режиме EXEC.

```
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# lldp run
Switch(config)# interface gigabitethernet 0/1
Switch(config-if)# lldp transmit
Switch(config-if)# lldp receive
Switch(config-if)# end
Switch# show lldp
Global LLDP Information:
  Status: ACTIVE
  LLDP advertisements are sent every 30 seconds
  LLDP hold time advertised is 120 seconds
  LLDP interface reinitialisation delay is 2 seconds
```

Поиск устройств с помощью LLDP

Предположим, что документация по топологии, показанной на рисунке, отсутствует. Сетевой администратор знает только, что коммутатор S1 подключен к двум устройствам.



При включенном LLDP соседние устройства могут быть обнаружены с помощью **show lldp neighbors** команды, как показано в выходных данных.

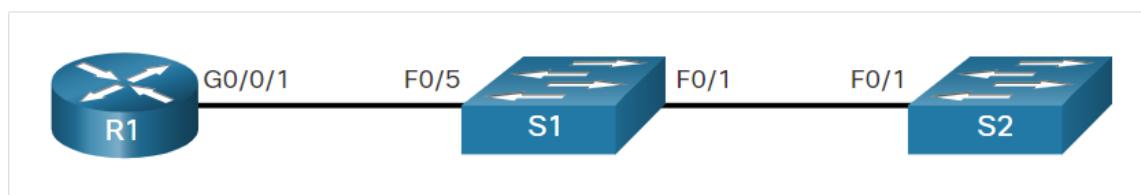
```

S1# show lldp neighbors
Capability codes:
  (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
  (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID          Local Intf     Hold-time  Capability      Port ID
R1                Fa0/5        117         R             Gi0/0/1
S2                Fa0/1        112         B             Fa0/1
Total entries displayed: 2

```

Сетевой администратор узнает о том, что у коммутатора S1 два соседних устройства: маршрутизатор и коммутатор. В этих выходных данных буква B (слово bridge - мост) может также означать коммутатор.

Из результатов **show lldp neighbors** можно построить топологию из S1, как показано на рисунке.



Если нужна более подробная информация о соседних устройствах, можно воспользоваться командой **show lldp neighbors detail**, которая предоставляет такие сведения, как версии IOS, IP-адреса и функции соседних устройств.

```

S1# show lldp neighbors detail
-----
Chassis id: 848a.8d44.49b0
Port id: Gi0/0/1
Port Description: GigabitEthernet0/0/1
System Name: R1

System Description:
Cisco IOS Software [Fuji], ISR Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version 16.9.4, RELEASE
SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2019 by Cisco Systems, Inc.
Compiled Thu 22-Aug-19 18:09 by mcpre

Time remaining: 111 seconds
System Capabilities: B,R
Enabled Capabilities: R
Management Addresses - not advertised
Auto Negotiation - not supported
Physical media capabilities - not advertised
Media Attachment Unit type - not advertised
Vlan ID: - not advertised

-----
Chassis id: 0025.83e6.4b00

```

Настройка LLDP

Включение:

```
R(config)# lldp run
```

Настройка передачи пакетов LLDP:

```
R(config)# interface interface-id
```

```
R(config-if)# lldp transmit
```

```
R(config-if)# lldp receive
```

Проверка:

```
R# show lldp
```

Обнаружение устройств:

```
R# show lldp neighbors
```

Выключен по умолчанию!!!

b. Службы времени: способы настройки системных часов, протокол NTP, его характеристика и принцип работы.

Настройка системных часов

- Основным источником информации о времени в системе являются программные часы маршрутизатора или коммутатора
- Синхронизируется время на всех устройствах в сети
- Параметры даты и времени на маршрутизаторе или коммутаторе можно задать вручную или через протокол сетевого времени (NTP).

Задание времени вручную:

```
R# clock set hh:mm:ss {day month | month day} year
```

```
R1# clock set 20:36:00 nov 15 2019
R1#
*Nov 15 20:36:00.000: %SYS-6-CLOCKUPDATE: System clock has been
updated from 21:32:31 UTC Fri Nov 15 2019 to 20:36:00 UTC Fri Nov 15
2019, configured from console by console.
```

Минусы:

1. Точность
2. Настройка на каждом устройстве
3. Если устройства выключили и включили, таймер пойдет с момента последнего выключения

Прежде чем вы действительно углубитесь в управление сетью, вам следует быть уверенным в том, что все ваши компоненты настроены на одно и то же время и дату.

Основным источником информации о времени в системе являются программные часы маршрутизатора или коммутатора, которые запускаются при загрузке системы. Важно, чтобы время на всех устройствах в сети было синхронизировано, поскольку все аспекты управления, безопасности, устранения неполадок и планирования сети требуют точных меток времени. Если время на устройствах не синхронизировано, определить порядок событий и их причину невозможно.

Как правило, параметры даты и времени на маршрутизаторе или коммутаторе можно задать одним из двух способов. Можно вручную настроить дату и время, как показано в примере, или настроить протокол сетевого времени (NTP).

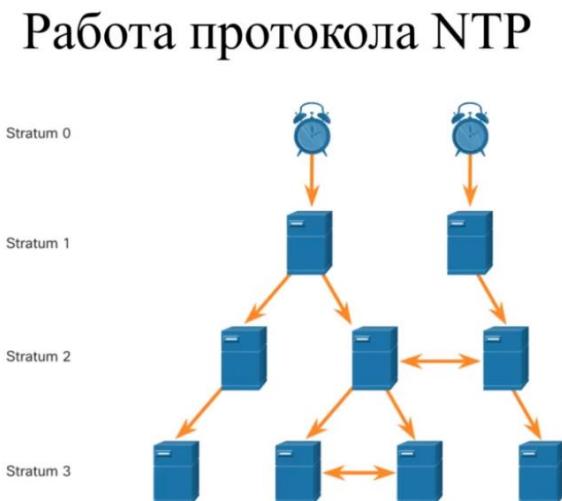
```
R1# clock set 16:01:00 sept 25 2020
*Sep 25 16:01:00.000: %SYS-6-CLOCKUPDATE: System clock has been updated from 13:09:49 UTC Fri Sep 25
2020 to 16:01:00 UTC Fri Sep 25 2020, configured from console by console.
Sep 25 16:01:00.001: %PKI-6-AUTHORITATIVE_CLOCK: The system clock has been set.
R1#
```

По мере роста сети становится все труднее обеспечивать синхронизацию времени на всех устройствах инфраструктуры. И даже в небольшой сетевой среде ручной метод не является оптимальным вариантом. Если произошла перезагрузка маршрутизатора, как ему узнать точную дату и метку времени?

Более эффективным решением является настройка в сети протокола NTP. С помощью этого протокола маршрутизаторы по сети могут синхронизировать свои настройки времени с NTP-сервером. Клиенты NTP, которые получают сведения о времени и дате из одного источника, используют более корректные настройки времени. Если в сети реализован протокол NTP, его можно настроить на синхронизацию с частным тактовым генератором или общедоступным сервером NTP в Интернете.

Протокол NTP использует порт UDP 123 и задокументирован в RFC 1305.

Работа протокола NTP



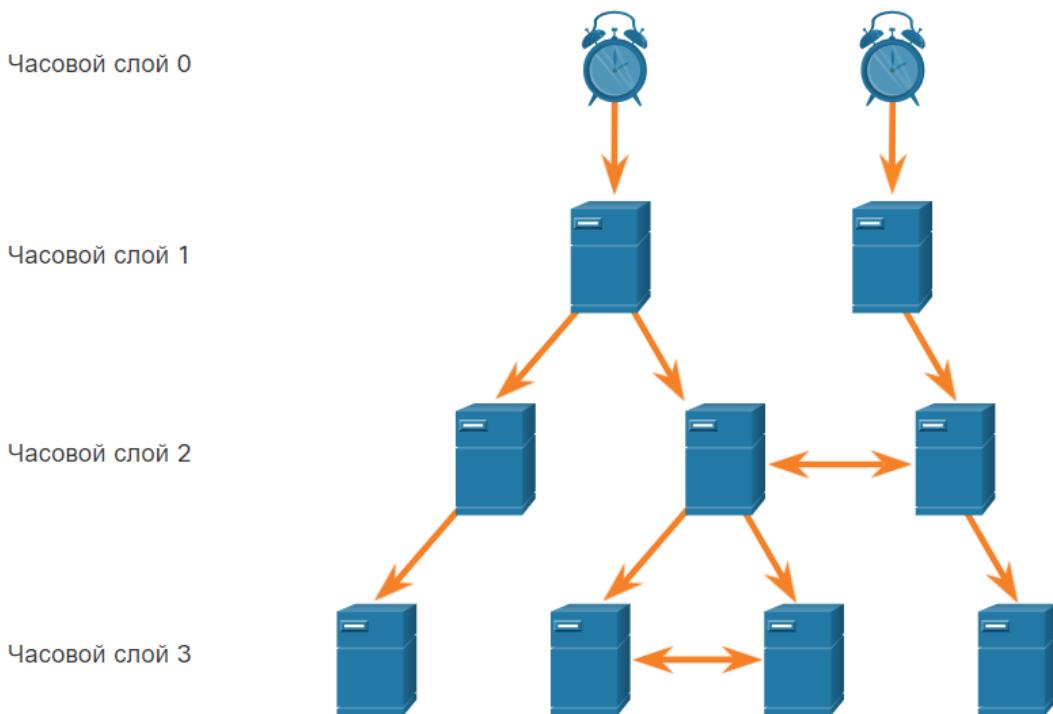
- Stratum 0 — доверенные источники времени
- Stratum 1 — подключены к stratum 0, основной стандарт сетевого времени
- Stratum 2 и далее — получают время от stratum 1

Максимальное количество переходов = 15

Порт UDP 123, RFC 1305

В сетях NTP используется иерархическая система источников времени. Каждый уровень этой иерархической системы называется часовым слоем (stratum). Уровень часового слоя определяется как количество переходов от доверенного источника. Для распределения синхронизированной информации о времени по сети используется протокол NTP. На рисунке показан пример сети NTP.

Серверы NTP расположены на трех уровнях, соответствующих трем часовым слоям. Часовой слой 1 подключен к часам часового слоя 0.



Часовой слой 0

Сеть NTP получает информацию о времени от доверенных источников времени. Эти доверенные источники времени, также называемые устройствами часового слоя 0, являются высокоточными устройствами хранения времени, которые считаются точными и работают практически без задержек. Устройства слоя 0 изображены на рисунке в виде часов.

Часовой слой 1

Устройства слоя 1 подключены напрямую к доверенным источникам времени. Они выступают в роли основного стандарта сетевого времени.

Часовой слой 2 и ниже

Серверы часового слоя 2 подключены к устройствам часового слоя 1 через сеть. Устройства часового слоя 2, например клиенты NTP, синхронизируют свое время с помощью пакетов NTP, которые они получают от серверов часового слоя 1. Эти устройства могут также выступать в роли серверов для устройств часового слоя 3.

Чем меньше номер часового слоя, тем ближе сервер расположен к доверенному источнику времени. Чем больше номер часового слоя, тем ниже его уровень. Максимальное количество переходов равно 15. Часовой слой 16 имеет самый низкий уровень и указывает на то, что устройство не синхронизировано. Серверы времени, находящиеся в одном часовом слое, могут работать как равноправные серверы времени на одном уровне часового слоя для обеспечения резервирования или проверки правильности времени.

Настройка и проверка NTP

Настройка NTP

Просмотр текущего времени:

```
R# show clock
```

Настройка клиента:

```
R(config)# ntp server ip-address
```

Настройка сервера на маршрутизаторе:

```
R(config)# interface interface-id
```

```
R(config-if)# ntp broadcast
```

```
R(config)# ntp master [stratum]
```

На рисунке показана топология, используемая для демонстрации конфигурации и проверки NTP.



Перед настройкой протокола NTP в сети введите команду **show clock**, которая отображает текущее время программных часов, как показано в примере. Обратите внимание, что с опцией **detail** источником времени является пользовательская конфигурация. Это означает, что время было настроено вручную с помощью **clock** команды.

```
R1# show clock detail  
20:55:10.207 UTC Fri Nov 15 2019  
Time source is user configuration
```

Команда **ntp server ip-address** в режиме глобальной конфигурации используется, чтобы настроить адрес 209.165.200.225 в качестве сервера NTP для маршрутизатора R1. Чтобы убедиться, что в качестве источника времени выбран NTP, выполните команду **show clock detail**. Обратите внимание, что теперь источником времени является NTP.

```
R1(config)# ntp server 209.165.200.225  
R1(config)# end  
R1# show clock detail  
21:01:34.563 UTC Fri Nov 15 2019  
Time source is NTP
```

Команды проверки:

```
R# show ntp associations
```

```
R# show ntp status
```

В следующем примере команды **show ntp associations** и **show ntp status** используются для проверки синхронизации R1 с сервером NTP в 209.165.200.225. Обратите внимание: маршрутизатор R1 синхронизирован с сервером NTP часового слоя 1 по адресу 209.165.200.225, который синхронизирован с часами GPS. Команда **show ntp status** показывает, что теперь маршрутизатор R1 является устройством часового слоя 2, которое синхронизировано с сервером NTP по адресу 209.165.220.225.

Примечание: Выделенный текст **st** обозначает слой.

```
R1# show ntp associations
  address      ref clock      st    when   poll reach  delay  offset  disp
*~209.165.200.225 .GPS.          1      61     64    377  0.481   7.480  4.261
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

R1# show ntp status
Clock is synchronized, stratum 2, reference is 209.165.200.225
nominal freq is 250.0000 Hz, actual freq is 249.9995 Hz, precision is 2**19
ntp uptime is 589900 (1/100 of seconds), resolution is 4016
reference time is DA088DD3.C4E659D3 (13:21:23.769 PST Fri Nov 15 2019)
clock offset is 7.0883 msec, root delay is 99.77 msec
root dispersion is 13.43 msec, peer dispersion is 2.48 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000001803 s/s
system poll interval is 64, last update was 169 sec ago.
```

Затем часы на S1 настроены для синхронизации с R1 с помощью команды **ntp server**, а затем конфигурация проверяется с помощью команды **show ntp associations**, как показано на экране.

```
S1(config)# ntp server 192.168.1.1
S1(config)# end
S1# show ntp associations
  address      ref clock      st    when   poll reach  delay  offset  disp
*~192.168.1.1    209.165.200.225  2      12     64    377  1.066  13.616  3.840
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

Команда **show ntp associations** позволяет убедиться, что часы на S1 теперь синхронизированы с R1 с адресом 192.168.1.1 по протоколу NTP. Маршрутизатор R1 является устройством часового слоя 2 и сервером NTP для коммутатора S1. Коммутатор S1 теперь является устройством часового слоя 3, которое может предоставлять службу NTP для остальных устройств в сети, например оконечных устройств.

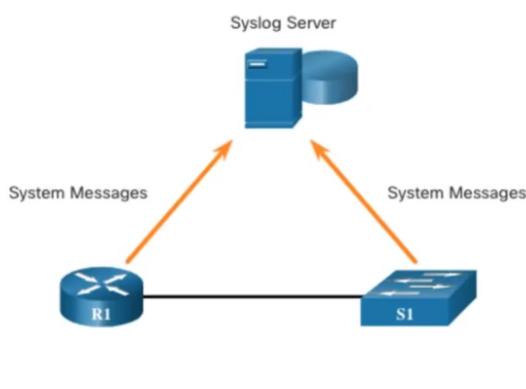
```

S1# show ntp status
Clock is synchronized, stratum 3, reference is 192.168.1.1
nominal freq is 119.2092 Hz, actual freq is 119.2088 Hz, precision is 2**17
reference time is DA08904B.3269C655 (13:31:55.196 PST Tue Nov 15 2019)
clock offset is 18.7764 msec, root delay is 102.42 msec
root dispersion is 38.03 msec, peer dispersion is 3.74 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000003925 s/s
system poll interval is 128, last update was 178 sec ago.

```

с. Системный журнал: протокол **Syslog**, характеристика и принцип работы, формат сообщений **Syslog**, уровни важности.

Syslog

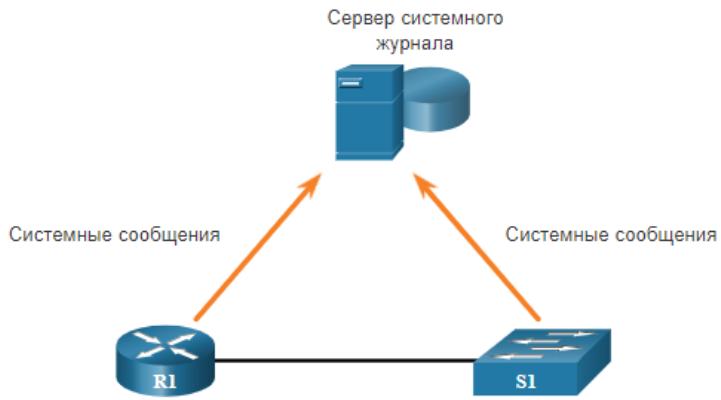


- Протокол системного журнала
- Стандарт
- Порт UDP 514
- Служба журналирования syslog предоставляет три основные возможности:
 - сбор информации в журнал для мониторинга и отладки;
 - выбор типа информации, сбор которой будет осуществляться;
 - определение получателей собранных сообщений syslog (буфер ведения журнала, порт консоли, тиния терминала, сервер Syslog)

Как и индикатор Check Engine на приборной панели автомобиля, компоненты в вашей сети могут сказать вам, если что-то не так. Протокол syslog был разработан для обеспечения того, чтобы вы могли получать и понимать эти сообщения. При возникновении определенных событий в сети сетевые устройства, используя доверенные механизмы, уведомляют администратора с помощью подробных системных сообщений. Эти сообщения могут быть некритическими или существенно важными. В распоряжении сетевых администраторов — широкий ассортимент вариантов хранения, интерпретации и отображения различных сообщений. Они также могут быть оповещены о тех сообщениях, которые могут оказать наибольшее влияние на сетевую инфраструктуру.

Самый распространенный способ получения системных сообщений — это использование протокола под названием **syslog**.

Термин **syslog** используется для описания стандарта. Он также используется для описания протокола, разработанного для этого стандарта. **Syslog** использует порт UDP 514 для отправки сообщений с уведомлением о событиях по сетям IP на средства сбора сообщений о событиях, как показано на рисунке.



Syslog поддерживают многие сетевые устройства, включая маршрутизаторы, коммутаторы, серверы приложений, межсетевые экраны и др. Протокол syslog позволяет сетевым устройствам отправлять системные сообщения по сети на серверы syslog.

Сервис ведения системного журнала выполняет три основные функции:

- сбор информации в журнал для мониторинга и устранения неполадок;
- выбор типа информации, сбор которой будет осуществляться;
- определение получателей собранных сообщений syslog.

Принцип работы Syslog

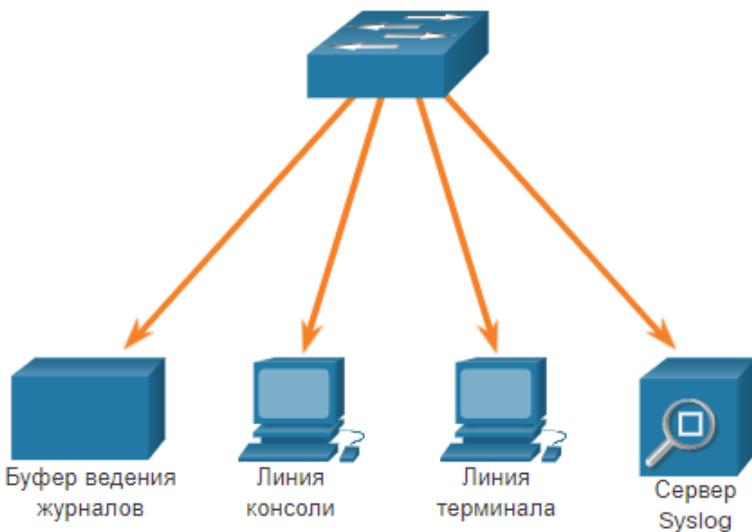
На сетевых устройствах Cisco протокол syslog начинает с отправки системных сообщений и вывода процесса **debug** в локальный процесс ведения журналов соответствующего устройства. Таким образом процесс ведения журналов управляет этими сообщениями и выводом, зависит от настроек устройства. Например, сообщения syslog могут отправляться по сети на внешний сервер syslog. Эти сообщения можно прочитать без необходимости доступа к самому устройству. Сообщения журнала и выходные данные, хранящиеся на внешнем сервере, могут включаться в различные отчеты для упрощения их прочтения.

Кроме того, сообщения syslog могут отправляться во внутренний буфер. Сообщения, отправленные во внутренний буфер, можно просматривать только через интерфейс командной строки устройства.

Наконец, сетевой администратор может указать, какие типы системных сообщений будут отправляться в различные места назначения. Например, можно настроить устройство, чтобы все системные сообщения отправлялись на внешний сервер syslog. Однако сообщения уровня debug будут пересыпаться во внутренний буфер и будут доступны только администратору через интерфейс командной строки.

Как показано на рисунке, в число популярных назначений для сообщений syslog входят следующие:

- буфер ведения журналов (ОЗУ в маршрутизаторе или коммутаторе);
- порт консоли;
- линия терминала;
- Сервер Syslog.



Можно удаленно наблюдать за системными сообщениями путем просмотра журналов на сервере Syslog или путем доступа к устройству по протоколам Telnet, SSH или через порт консоли.

Формат сообщений syslog

Устройства Cisco создают сообщения syslog при определенных сетевых событиях. Во всех сообщениях syslog указывается уровень важности (severity level) и объект (facility).

Чем меньше назначаемое число, тем более важным является оповещение syslog. В настройках уровня важности сообщений можно установить, куда отправлять сообщения каждого типа (например на консоль или в другие места назначения). Полный перечень уровней syslog представлен в таблице.

Название уровня серьезности	Уровень серьезности	Описание
Чрезвычайная ситуация	Уровень 0	Систему нельзя использовать
Предупреждение	Уровень 1	Требуется принять немедленные меры
Критический	Уровень 2	Критическое состояние
Ошибка	Уровень 3	Состояние ошибки
Предупреждение	Уровень 4	Состояние предупреждения
Уведомление	Уровень 5	Нормальное, но требующее внимания состояние
Информационный	Уровень 6	Информационное сообщение
Отладка	Уровень 7	Сообщение отладки

Каждый уровень syslog имеет собственный смысл:

- **Предупреждение Уровень 4 - Чрезвычайная ситуация Уровень 0:** Это сообщения о сбоях программного или аппаратного обеспечения. Эти типы сообщений свидетельствуют о том, что затронута работа устройства. Назначаемый уровень syslog зависит от серьезности проблемы.
- **Уведомление Уровень 5:** Этот уровень уведомлений об обычных, но важных событиях. На уровне уведомления, например, отображаются сообщения об изменении состояния интерфейса на активное или неактивное или о перезапуске системы.
- **Информационный Уровень 6:** Это обычные информационные сообщения, которые не влияют на работу устройства. Например, при загрузке устройства Cisco может появиться следующее

- информационное сообщение: %LICENSE-6-EULA_ACCEPT_ALL: The Right to Use End User License Agreement is accepted.
- **Отладка Уровень 7:** Сообщения этого уровня содержат выходные данные, полученные в результате выполнения различных команд **debug**.

Объекты Syslog

Формат сообщений Syslog

```
seq no: timestamp: %facility-severity-MNEMONIC:  
description
```

Пример:

```
00:00:46: %LINK-3-UPDOWN: Interface Port-  
channel1, changed state to up
```

Помимо указания уровня важности в сообщениях syslog также содержатся сведения об объекте. Объекты syslog (syslog facilities) — это идентификаторы сервисов, которые определяют и классифицируют данные о состоянии системы для отчетов об ошибках и событиях. Доступные варианты объектов ведения журнала зависят от конкретного сетевого устройства.

Ниже приведены некоторые из общепринятых объектов сообщений syslog, которые регистрируются на маршрутизаторах Cisco IOS:

- IP
- Протокол OSPF
- Операционная система SYS
- Протокол IPSec
- IP интерфейса (IF)

По умолчанию формат сообщений syslog в ПО Cisco IOS выглядит следующим образом:

```
%facility-severity-MNEMONIC: description
```

Пример выходных данных об изменении состояния канала EtherChannel коммутатора Cisco на активное будет выглядеть следующим образом:

```
%LINK-3-UPDOWN: Interface Port-channel1, changed state to up
```

В этом примере объектом является LINK, назначен уровень серьезности 3, в качестве КРАТКОГО КОДА выступает UPDOWN.

Наиболее распространенными сообщениями являются сообщения об изменении состояния каналов на активное и неактивное, а также сообщения, создаваемые устройством при выходе из режима настройки. Если настроено журналирование в списках контроля доступа, устройство создает сообщения syslog, если пакеты соответствуют заданным условиям.

Настройка временной метки системного журнала

Настройка меток времени:

```
R(config)# service timestamps log datetime
```

По умолчанию в сообщениях журнала нет метки времени. В примере, интерфейс GigabitEthernet 0/0/0 маршрутизатора R1 отключен. Сообщение, зарегистрированное на консоли, не показывает, когда состояние интерфейса было изменено. Сообщения журнала должны иметь метку времени. Потому что, когда они отправляются следующему адресату, например на сервер системного журнала, появляется запись о создании сообщения.

```
R1# configure terminal
R1(config)# interface g0/0/0
R1(config-if)# shutdown
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to down
R1(config-if)# exit
R1(config)# service timestamps log datetime
R1(config)# interface g0/0/0
R1(config-if)# no shutdown
*Mar  1 11:52:42: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/0, changed state to down
*Mar  1 11:52:45: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/0, changed state to up
*Mar  1 11:52:46: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0,
changed state to up
R1(config-if)#

```

Команда **service timestamps log datetime** позволяет принудительно отображать дату и время для зарегистрированных событий. Как показано на рисунке, теперь при повторном включении интерфейса GigabitEthernet 0/0/0 на маршрутизаторе сообщения журнала содержат дату и время.

Примечание: При использовании ключевого слова **datetime** часы на сетевом устройстве необходимо настроить либо вручную, либо с помощью NTP, как упоминалось ранее.

Просмотр журнала сообщений

Вывод журнала на консоль:

```
R(config)# logging console
```

Сохранение журнала в буфер:

```
R(config)# logging buffered
```

Просмотр журнала сообщений:

```
R# show logging
```

Просмотр журнала сообщений

```
R1# show logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

    Console logging: level debugging, 32 messages logged, xml disabled,
                      filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                      filtering disabled
    Buffer logging: level debugging, 32 messages logged, xml disabled,
                      filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 34 message lines logged
        Logging Source-Interface:          VRF Name:
    Log Buffer (8192 bytes):

*Jan 2 00:00:02.527: %LICENSE-6-EULA_ACCEPT_ALL: The Right to Use End User License
Agreement is accepted
*Jan 2 00:00:02.631: %IOS_LICENSE_IMAGE_APPLICATION-6-LICENSE_LEVEL: Module name =
c1900 Next reboot level = ipbasek9 and License = ipbasek9
*Jan 2 00:00:02.851: %IOS_LICENSE_IMAGE_APPLICATION-6-LICENSE_LEVEL: Module name =
c1900 Next reboot level = securityk9 and License = securityk9
*Jun 12 17:46:01.619: %IFMGR-7-NO_IFINDEX_FILE: Unable to open nvram:/ifIndex-table No
such file or directory
```

Отправка сообщений на сервер Syslog

Настройка адресата:

```
R(config)# logging host
```

Настройка отправляемых сообщений:

```
R(config)# logging trap level
```

Настройка интерфейса источника:

```
R(config)# logging source-interface interface-type
interface-number
```

Пример

```
R1(config)# logging 192.168.1.3
R1(config)# logging trap 4
R1(config)# logging source-interface g0/0
R1(config)# interface loopback 0
R1(config-if)#
*Jun 12 22:06:02.902: %LINK-3-UPDOWN: Interface Loopback0, changed state to up
*Jun 12 22:06:03.902: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
changed state to up
*Jun 12 22:06:03.902: %SYS-6-LOGGINGHOST_STARTSTOP: Logging to host 192.168.1.3
port 514 started - CLI initiated
R1(config-if)# shutdown
R1(config-if)#
*Jun 12 22:06:49.642: %LINK-5-CHANGED: Interface Loopback0, changed state to
administratively down
*Jun 12 22:06:50.642: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
changed state to down
R1(config-if)# no shutdown
R1(config-if)#
*Jun 12 22:09:18.210: %LINK-3-UPDOWN: Interface Loopback0, changed state to up
*Jun 12 22:09:19.210: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
changed state to up
```

show logging

```
R1# show logging | include changed state to up
*Jun 12 17:46:26.143: %LINK-3-UPDOWN: Interface
GigabitEthernet0/1, changed state to up
*Jun 12 17:46:26.143: %LINK-3-UPDOWN: Interface Serial0/0/1,
changed state to up
*Jun 12 17:46:27.263: %LINEPROTO-5-UPDOWN: Line protocol on
Interface GigabitEthernet0/1, changed state to up
*Jun 12 17:46:27.263: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Serial0/0/1, changed state to up
*Jun 12 20:28:43.427: %LINK-3-UPDOWN: Interface
GigabitEthernet0/0, changed state to up
*Jun 12 20:28:44.427: %LINEPROTO-5-UPDOWN: Line protocol on
Interface GigabitEthernet0/0, changed state to up
*Jun 12 22:04:11.862: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Loopback0, changed state to up
*Jun 12 22:06:02.902: %LINK-3-UPDOWN: Interface Loopback0,
changed state to up
*Jun 12 22:06:03.902: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Loopback0, changed state to up
*Jun 12 22:09:18.210: %LINK-3-UPDOWN: Interface Loopback0,
changed state to up
*Jun 12 22:09:19.210: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Loopback0, changed state to up
*Jun 12 22:35:55.926: %LINK-3-UPDOWN: Interface Loopback0,
changed state to up
*Jun 12 22:35:56.926: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Loopback0, changed state to up
```

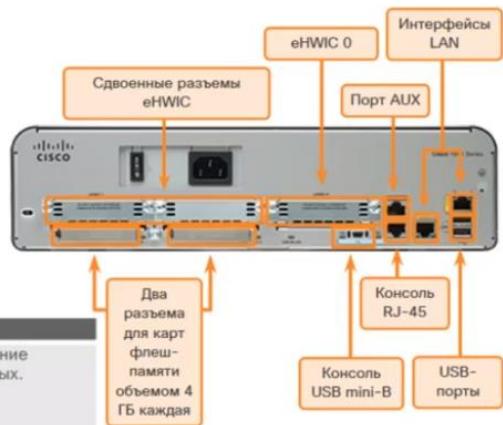
show logging

```
R1# show logging | begin Jun 12 22:35
*Jun 12 22:35:46.206: %LINK-5-CHANGED: Interface Loopback0,
changed state to administratively down
*Jun 12 22:35:47.206: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Loopback0, changed state to down
*Jun 12 22:35:55.926: %LINK-3-UPDOWN: Interface Loopback0,
changed state to up
*Jun 12 22:35:56.926: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Loopback0, changed state to up
*Jun 12 22:49:52.122: %SYS-5-CONFIG_I: Configured from console by
console
*Jun 12 23:15:48.418: %SYS-5-CONFIG_I: Configured from console by
console
```

8. Обслуживание сетевого оборудования:

а. Операционные и файловые системы сетевого оборудования.

Память устройств



Память	Описание
Оперативная память (RAM)	Энергозависимая память, которая обеспечивает временное хранение различных приложений и процессов, в том числе следующих данных. <ul style="list-style-type: none">Текущая версия IOSФайл текущей конфигурацииТаблица IP-маршрутизации и таблица ARPБуфер пакетов
Постоянное запоминающее устройство (ПЗУ)	Энергонезависимая память, которая обеспечивает постоянное хранение следующих данных. <ul style="list-style-type: none">Указания по начальной загрузкеБазовое программное обеспечение для диагностикиВерсия IOS с ограниченной функциональностью на случай, если маршрутизатору не удастся загрузить полнофункциональную версию IOS
Энергонезависимая оперативная память (NVRAM)	Энергонезависимая память, которая обеспечивает постоянное хранение следующих данных. <ul style="list-style-type: none">Файл загрузочной конфигурации
Флеш-память	Энергонезависимая память, которая обеспечивает постоянное хранение следующих данных. <ul style="list-style-type: none">IOSПрочие системные файлы

3

Файловые системы маршрутизаторов

Если вы думаете, что вы не можете вспомнить, как вы настроили каждое устройство в вашей сети, вы не одиноки. В большой сети невозможно вручную настроить каждое устройство. К счастью, существует множество способов скопировать или обновить конфигурации, а затем просто вставить их. Для этого вам нужно знать, как просматривать файловые системы и управлять ими.

Файловая система Cisco IOS (IFS) позволяет администраторам перемещаться по различным каталогам, отображать список файлов в каталоге. Администратор также может создавать подкаталоги во флэш-памяти или на диске. Для различных устройств перечень доступных папок может отличаться.

В примере показан результат выполнения команды **show file systems**, которая в этом примере выводит список всех доступных файловых систем на маршрутизаторе Cisco 4221r.

```

Router# show file systems
File Systems:
  Size(b)      Free(b)       Type  Flags  Prefixes
  -           -   opaque    rw    system:
  -           -   opaque    rw    tmpsys:
* 7194652672  6294822912   disk   rw   bootflash: flash:##
  256589824   256573440    disk   rw   usb0:
  1804468224  1723789312   disk   ro   webui:
  -           -   opaque    rw   null:
  -           -   opaque    ro   tar:
  -           -   network   rw   tftp:
  -           -   opaque    wo   syslog:
33554432     33539983    nvram  rw   nvram:
  -           -   network   rw   rcp:
  -           -   network   rw   ftp:
  -           -   network   rw   http:
  -           -   network   rw   scp:
  -           -   network   rw   sftp:
  -           -   network   rw   https:
  -           -   opaque    ro   cns:

```

Router#

Данная команда предоставляет полезную информацию, например объем общей и свободной памяти, тип файловой системы и ее разрешения. Доступны следующие разрешения: «только чтение» (ro), «только запись» (wo) и «чтение и запись» (rw). Они отображаются в столбце Flags в выходных данных команды.

Хотя в списке доступно несколько файловых систем, нас интересуют в первую очередь файловые системы TFTP, флеш-память и NVRAM.

Обратите внимание, что перед файловой системой флеш-памяти всегда указывается символ звездочки. Это означает, что флеш-память является текущей файловой системой по умолчанию. Загружаемая IOS размещена во флеш-памяти. Таким образом, к списку флеш-памяти добавляется символ решетки (#), указывая на то, что это загрузочный диск.

Файловая система Flash

В примере показаны результаты выполнения команды **dir** (directory).

```

Router# dir
Directory of flash0:/

 1 -rw-      2903 Sep  7 2012 06:58:26 +00:00  cpcconfig-
                                              19xx.cfg
 2 -rw-    3000320 Sep  7 2012 06:58:40 +00:00  cpexpress.tar
 3 -rw-     1038 Sep  7 2012 06:58:52 +00:00  home.shtml
 4 -rw-   122880 Sep  7 2012 06:59:02 +00:00  home.tar
 5 -rw-  1697952 Sep  7 2012 06:59:20 +00:00  securedesktop-
                                              ios-3.1.1.45-k9.pkg
 6 -rw-   415956 Sep  7 2012 06:59:34 +00:00  sslclient-win-
                                              1.1.4.176.pkg
 7 -rw- 67998028 Sep 26 2012 17:32:14 +00:00  c1900-
                                              universalk9-
                                              mz.SPA.152-4.M1.bin

256487424 bytes total (183234560 bytes free)

```

Поскольку флеш-память является файловой системой по умолчанию, то в результатах выполнения команды **dir** указывается содержимое флеш-памяти. Во флеш-памяти размещено несколько файлов, однако в первую очередь нас интересует последняя запись. Это имя текущего образа файла Cisco IOS, запущенного в ОЗУ.

Файловая система NVRAM

Чтобы просмотреть содержимое NVRAM, необходимо изменить текущую файловую систему по умолчанию, используя команду **cd** (изменить каталог), как показано в примере.

```

Router#
Router# cd nvram:
Router# pwd
nvram:/
Router# dir
Directory of nvram:/
 32769 -rw-          1024          startup-config
 32770 ----          61           private-config
 32771 -rw-          1024          underlying-config
 1 ----             4            private-KS1
 2 -rw-          2945          cwmplib_inventory
 5 ----             447          persistent-data
 6 -rw-          1237          ISR4221-2x1GE_0_0_0
 8 -rw-             17          ecfm_ieee_mib
 9 -rw-              0           ifIndex-table
10 -rw-          1431          NIM-2T_0_1_0
12 -rw-          820           IOS-Self-Sig#1.cer
13 -rw-          820           IOS-Self-Sig#2.cer
33554432 bytes total (33539983 bytes free)
Router#

```

Нынешняя команда рабочего каталога: **pwd**. Эта команда подтверждает, что просматривается именно каталог NVRAM. И наконец, команда **dir** создает список содержимого энергонезависимой памяти. Хотя в

списке представлено несколько файлов конфигурации, в первую очередь нас интересует файл конфигурации начальной загрузки.

Файловые системы коммутатора

Используя файловую систему флеш-памяти коммутатора Cisco 2960, можно скопировать файлы конфигурации и архивировать (скачивать и закачивать) образы ОС.

Для просмотра файловых систем на коммутаторе Catalyst используется та же команда, что и для маршрутизатора Cisco: **show file systems**, как показано в примере.

```
Switch# show file systems
File Systems:
  Size(b)  Free(b)   Type  Flags  Prefixes
* 32514048  20887552    flash  rw    flash:# 
      -        -  opaque  rw    vb: 
      -        -  opaque  ro    bs: 
      -        -  opaque  rw    system: 
      -        -  opaque  rw    tmpsys: 
  65536     48897    nvram  rw    nvram: 
      -        -  opaque  ro    xmodem: 
      -        -  opaque  ro    ymodem: 
      -        -  opaque  rw    null: 
      -        -  opaque  ro    tar: 
      -        -  network rw    tftp: 
      -        -  network rw    rcp: 
      -        -  network rw    http: 
      -        -  network rw    ftp: 
      -        -  network rw    scp: 
      -        -  network rw    https: 
      -        -  opaque  ro    cns: 

Switch#
```

b. Загрузка коммутатора и маршрутизатора. Начальный загрузчик.

Загрузка коммутатора

- Самотестирование питания POST (программа тестирует ЦП, процессор, оперативную динамическую память (DRAM) и часть флеш-устройств)
- Запуск начального загрузчика
- Начальный загрузчик выполняет низкоуровневую инициализацию ЦП. Он инициализирует регистры ЦП, которые контролируют место отображения физической памяти, количество памяти и ее скорость
- Запуск файловой системы флеш-памяти на материнской плате
- Загрузка IOS, передача управления коммутатором



Перед настройкой коммутатора необходимо включить его и разрешить ему пройти через пять шагов последовательности загрузки. Этот раздел посвящен основам настройки коммутатора и включает в себя лабораторную работу в конце.

После включения коммутатор Cisco проходит следующие стадии загрузки:

Шаг 1: Во-первых, коммутатор загружает программу самопроверки питания (POST), хранящуюся в ПЗУ. POST проверяет подсистему CPU. Он проверяет процессор, DRAM и часть флэш-устройства, которая составляет файловую систему флэш-памяти.

Шаг 2: После этого на коммутаторе запускается программное обеспечение начального загрузчика. Начальный загрузчик — это небольшая программа, которая хранится в ПЗУ и запускается сразу после успешного завершения проверки POST.

Шаг 3: Начальный загрузчик выполняет низкоуровневую инициализацию центрального процессора. Он инициализирует регистры ЦП, которые контролируют место отображения физической памяти, количество памяти и ее скорость.

Шаг 4: Затем программа запускает файловую систему флеш-памяти на материнской плате.

Шаг 5: Наконец, начальный загрузчик находит и загружает образ операционной системы IOS по умолчанию и передает ей управление коммутатором.

Команда "boot system"

Он пытается выполнить автоматическую загрузку, используя информацию из переменной для среды BOOT. Если эта переменная не установлена, коммутатор пытается загрузить и выполнить первый исполняемый файл, который он может найти.

Затем операционная система IOS инициализирует интерфейсы, используя команды Cisco IOS из файла загрузочной конфигурации, который хранится в энергонезависимом ОЗУ (NVRAM). Файл startup-config называется **config.text** и находится во флэш-памяти.

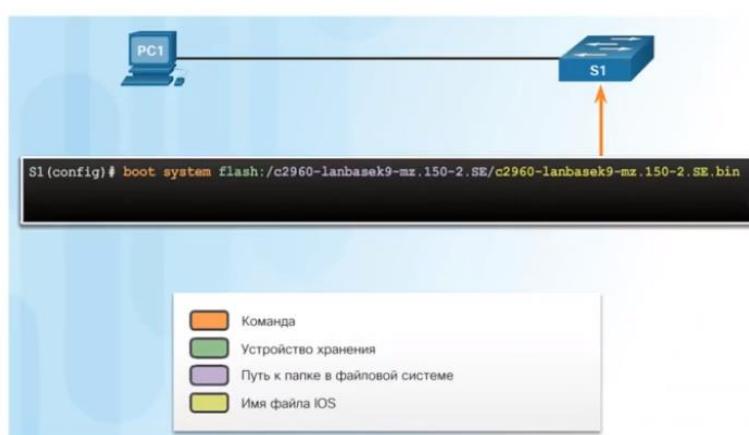
В примере переменная среды BOOT задается с помощью команды режима глобальной конфигурации **boot system**. Обратите внимание, что IOS находится в отдельной папке и указан путь к папке. Используйте команду **show boot**, чтобы узнать, как настроен файл текущей загрузки IOS.

```
S1(config)# boot system flash:/c2960-lanbasek9-mz.150-2.SE/c2960-lanbasek9-mz.150-2.SE.bin
```

Таблица определяет каждую часть команды **boot system**.

Команда	Определение
boot system	Основная команда
flash:	Устройство хранения
c2960-lanbasek9-mz.150-2.SE/	Путь к файловой системе
c2960-lanbasek9-mz.150-2.SE.bin	Имя файла IOS

Загрузка коммутатора



- IOS выполняет автоматическую загрузку (из BOOT или первый исполняемый файл)
- Затем IOS инициализирует интерфейсы, используя команды Cisco IOS из файла загрузочной конфигурации, который хранится в энергонезависимом ОЗУ (NVRAM). Файл startup-config называется config.text и находится во флэш-памяти
- Команда show boot показывает, как настроен файл текущей загрузки IOS

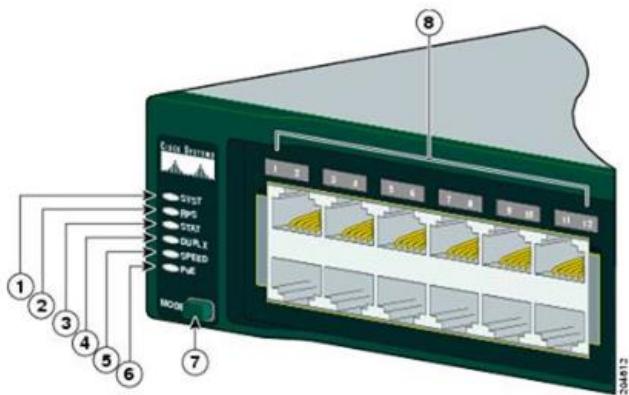
Светодиодные индикаторы коммутатора

Коммутаторы Cisco Catalyst оснащены несколькими индикаторами состояния. Индикаторы коммутатора позволяют быстро оценить активность и производительность коммутатора. Коммутаторы различных моделей и наборов характеристик имеют разные светодиоды, и их размещение на передней панели коммутатора также может отличаться.

Светодиодные индикаторы коммутатора



- **Mode** используется для переключения состояния порта, дуплексного режима порта, скорости порта и, если поддерживается, состояния Power over Ethernet (PoE) светодиодов порта
- **SYST** показывает, есть ли питание системы и функционирует ли она должным образом
- **RPS** показывает статус резервного источника питания
- **STAT** указывает режим статуса порта. Данный режим является режимом по умолчанию
- **DUPLEX** указывает на режим дуплексного порта (откл — полудуплекс, зелёный — дуплекс)
- **SPEED** указывает выбранный режим скорости порта
- **PoE** указывает на состояние PoE на портах



Кнопка Mode (7 на рисунке) используется для переключения состояния порта, дуплексного режима порта, скорости порта и, если поддерживается, состояния Power over Ethernet (PoE) светодиодов порта (8 на рисунке).

1 SYST

Системный индикатор показывает, есть ли питание системы и функционирует ли она должным образом. Если светодиод выключен, это означает, что система не включена. Если индикатор горит зеленым цветом, то вентиляторы работают нормально. Если индикатор горит желтым, система получает питание, но работает с перебоями.

2 RPS

Показывает статус RPS. Если этот индикатор не горит, то система RPS выключена или подключена неправильно. Если индикатор горит зеленым, то резервный источник питания подключен и готов к обеспечению резервного питания. Если индикатор мигает зеленым, то RPS подключена, но недоступна, поскольку обеспечивает питание другому устройству. Если индикатор горит желтым, то резервный источник питания находится в режиме ожидания или неисправен. Если индикатор мигает желтым, произошел сбой внутреннего блока питания коммутатора и резервный источник питания обеспечивает питание коммутатора.

3 STAT

Указывает, что режим статуса порта выбран при зеленом светодиоде. Данный режим является режимом по умолчанию. Если выбран этот параметр, светодиоды портов будут отображать цвета с различным значением. Не светится — нет связи или порт отключен администратором. Если светодиод зеленый, то канал активный. Если светодиод мигает зеленым цветом, есть активность на порту и порт отправляет или принимает данные. Если светодиод чередуется зелено-янтарный, указывает, что есть ошибка соединения. Если индикатор горит желтым, то порт заблокирован, чтобы гарантировать отсутствие петли в домене пересылки (обычно порты находятся в этом состоянии в течение первых 30 секунд после активации). Если светодиод мигает желтым, порт блокируется для предотвращения возможной петли в широковещательном домене.

4 DUPLEX

Указывает, что режим дуплексного порта выбран при зеленом светодиоде. Если выбран этот параметр, отключаемые светодиоды портов находятся в полу duplexном режиме. Если светодиод порта зеленый, порт находится в полнодуплексном режиме.

5 SPEED

Индикатор скорости порта указывает выбранный режим скорости порта. Если выбран этот параметр, светодиоды портов будут отображать цвета с различным значением. Если индикатор не горит, порт работает на скорости 10 Мбит/с. Если индикатор горит зеленым, то порт работает на скорости 100 Мбит/с. Если индикатор мигает зеленым, то порт работает на скорости 1000 Мбит/с.

6 PoE(подача питания через ethernet – power over ethernet)

Если на устройстве поддерживается PoE, будет присутствовать светодиод режима PoE. Если светодиод выключен, он указывает, что режим PoE не выбран и что ни один из портов не был отключен или не был помещен в аварийное состояние. Если светодиод мигает желтым, режим PoE не выбран, но по крайней мере один из портов отключен или имеет сбой PoE. Если светодиод зеленый, он указывает, что выбран режим PoE, а светодиоды порта будут отображать цвета с разными значениями. Если светодиод порта выключен, PoE выключен. Если светодиод порта зеленый, PoE работает. Если индикатор порта мигает зеленым и желтым, в PoE отказано, поскольку подача питания на устройство с питанием превысит выходную мощность коммутатора. Если светодиод мигает желтым, PoE выключен из-за неисправности. Если индикатор горит желтым, PoE для порта отключен.

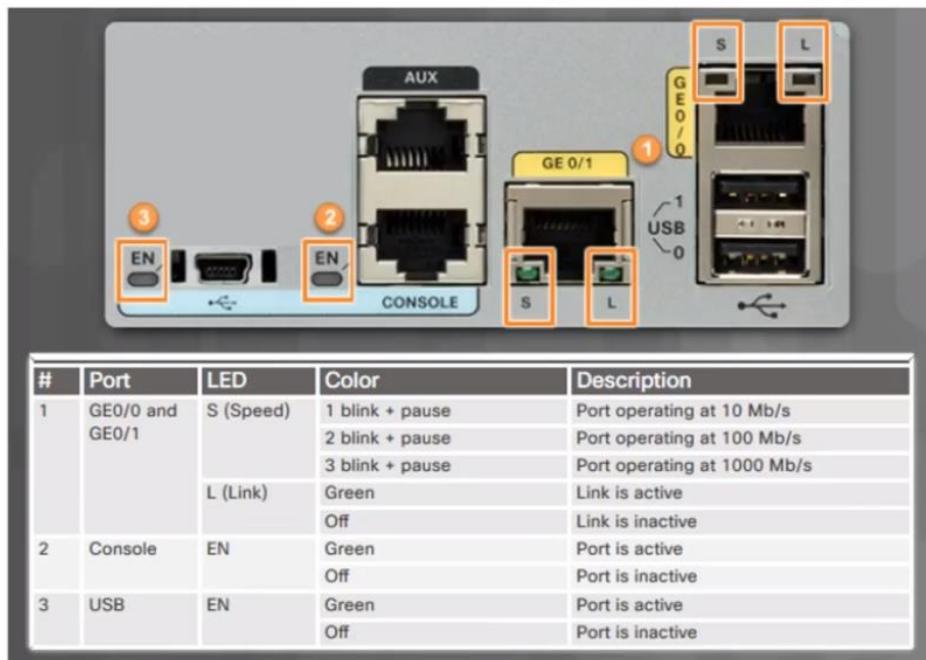
Светодиодные индикаторы коммутатора

	Выкл.	Зеленый	Часто мигающий зеленый	Желтый	Часто мигающий оранжевый	Мигающий зеленый и оранжевый
RPS	Выкл/Нет RPS	RPS готов	RPS включен, но недоступен	Резервный или неисправный RPS	Внутренняя PS не удалось, RPS обеспечивает питание	Нет
PoE	Не выбрано, проблем нет	Выбранный	—	—	Не выбран, проблемы с портом присутствуют	Нет

При выборе именованного режима свет, связанный с каждым физическим портом, указывает:

STAT	Нет связи или выключения	Соединение установлено	Действие	Порт блокирует петлю	Порт блокирует петлю	Ошибка соединения.
Дуплексный режим	Полудуплекс	Полный дуплекс	—	—	—	—
SPEED	10 Мбит/с	100 Мбит/с	1000 Мбит/с	—	—	—
PoE	PoE выключен	PoE включен	Нет	PoE отключен	Питание PoE отключено из-за ошибки	PoE отклонено (сверх бюджета)

Светодиодные индикаторы маршрутизатора

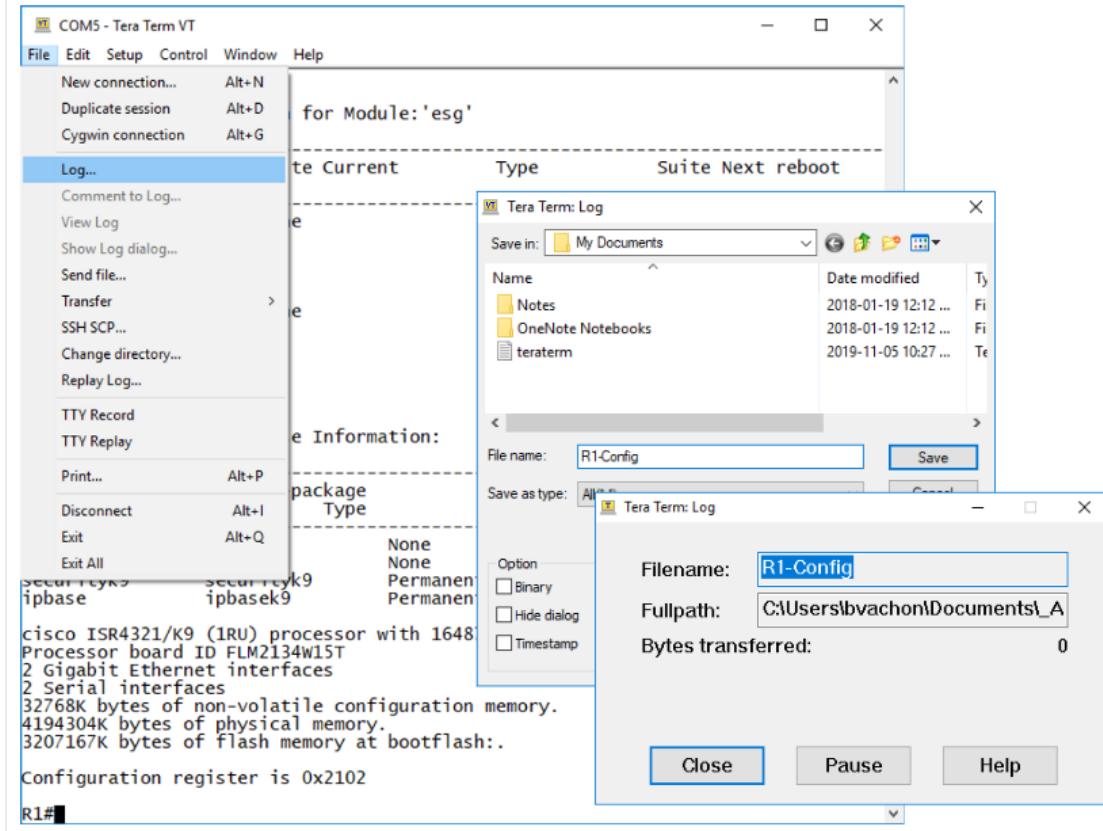


8

с. Резервное копирование и восстановление конфигурации.

Использование текстового файла для создания резервной копии конфигурации

Файлы конфигурации можно сохранить в текстовом файле, используя программу Tera Term, как показано на рисунке



Шаг 1. В меню File (Файл) выберите Log.

Шаг 2. Выберите путь для сохранения файла. Программа Tera Term запустит процесс захвата текста.

Шаг 3. После начала данного процесса в командной строке привилегированного режима EXEC выполните команду **show running-config** или **show startup-config**. Текст, отображаемый в окне терминала, будет отправлен в выбранный файл.

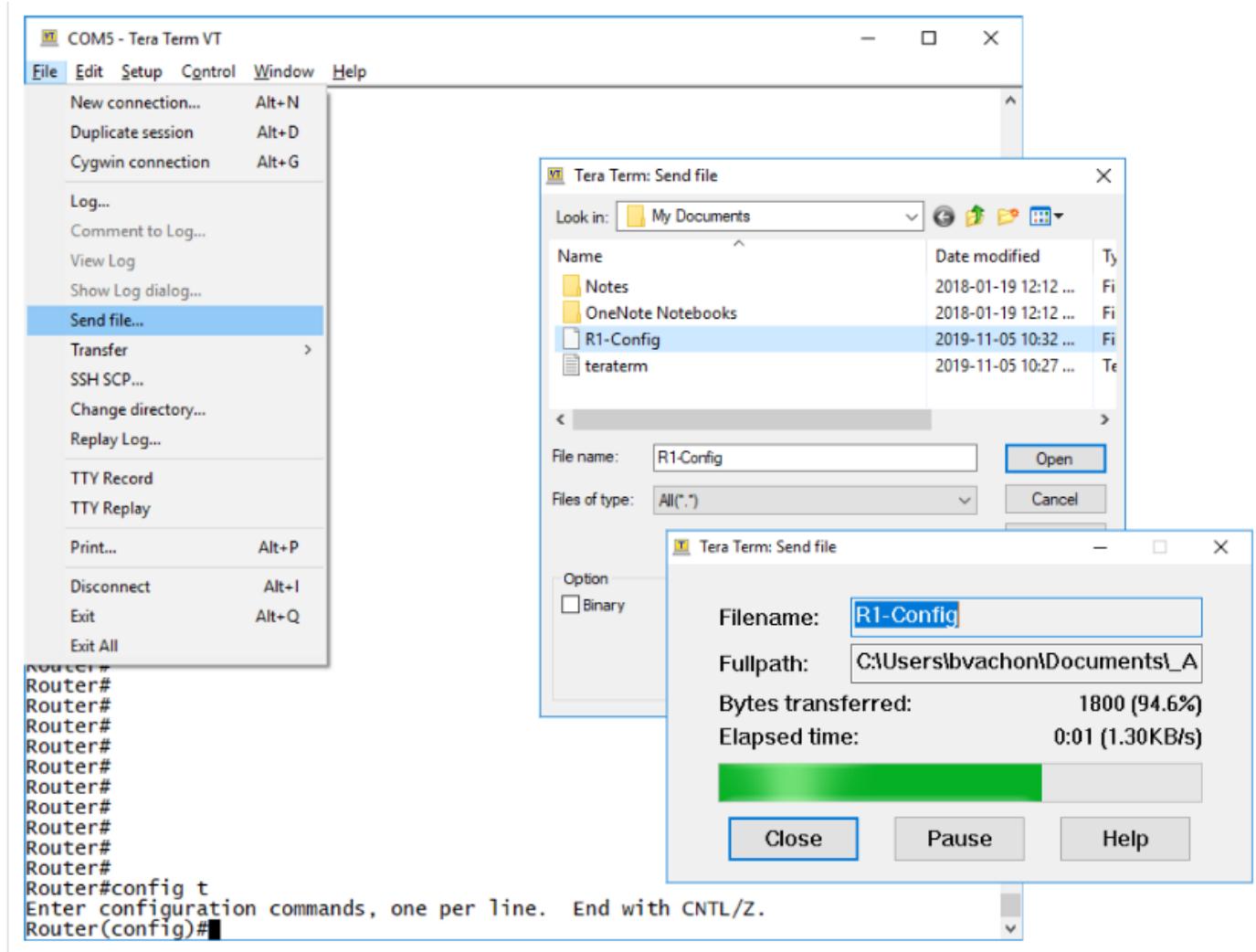
Шаг 4. When the capture is complete, select Close in the Tera Term: Log window.

Шаг 5. Просмотрите файл, чтобы убедиться в том, что он не поврежден.

Использование текстового файла для восстановления конфигурации

Конфигурацию можно скопировать из файла, а затем напрямую вставить на устройство. IOS выполняет каждую строку текста конфигурации в виде команды. Это означает, что файл необходимо будет отредактировать, чтобы зашифрованные пароли имели текстовый формат. Также необходимо удалить сообщения операционной среды IOS и весь не относящийся к командам текст типа --More--. Кроме того, перед вставкой конфигурации вы можете добавить **enable** и **configure terminal** в начало файла или перейти в режим глобальной конфигурации. Этот процесс рассматривается в рамках лабораторной работы после этой главы.

Вместо копирования и вставки конфигурацию можно восстановить из текстового файла с помощью Tera Term, как показано на рисунке.



При использовании программы Tera Term необходимо выполнить следующие действия.

Шаг 1. В меню File (Файл) выберите пункт **Send file** (Отправить файл).

Шаг 2. Укажите путь к файлу, который необходимо скопировать на данное устройство, и нажмите **Open** (Открыть).

Шаг 3. После этого программа Tera Term вставит этот файл в память устройства.

В интерфейсе CLI текстовое содержимое этого файла будет использоваться в качестве команд и станет текущей конфигурацией устройства.

Использование TFTP для резервного копирования и восстановления конфигурации

Использование TFTP для сохранения резервной копии конфигурации

Копии файлов конфигурации необходимо хранить как файлы резервных копий на случай возникновения проблем. Файлы конфигурации можно хранить на сервере простого протокола передачи файлов (TFTP) или на USB-накопителе. Файл конфигурации также необходимо включить в сетевую документацию.

Чтобы сохранить текущую конфигурацию или конфигурацию начальной загрузки на TFTP-сервер, используйте команду **copy running-config tftp** или **copy startup-config tftp**, как показано в примере.

```
R1# copy running-config tftp
Remote host []?192.168.10.254
Name of the configuration file to write[R1-config]? R1-Jan-2019
Write file R1-Jan-2019 to 192.168.10.254? [confirm]
Writing R1-Jan-2019 !!!!!! [OK]
```

Для резервного копирования текущей конфигурации на TFTP-сервер выполните указанные ниже действия.

Шаг 1. Введите следующую команду: **copy running-config tftp**.

Шаг 2. Введите IP-адрес узла, куда следует сохранить файл конфигурации.

Шаг 3. Введите имя, которое следует присвоить файлу конфигурации.

Шаг 4. Нажмите клавишу **Enter** для подтверждения каждого последующего действия.

Использование TFTP для восстановления резервной копии конфигурации

Чтобы восстановить текущую конфигурацию или конфигурацию начальной загрузки с TFTP-сервера, используйте команду **copy tftp running-config** или **copy tftp startup-config**. Для восстановления текущей конфигурации с TFTP-сервера выполните указанные ниже действия.

Шаг 1. Введите следующую команду: **copy tftp running-config**.

Шаг 2. Введите IP-адрес узла, на котором хранится файл конфигурации.

Шаг 3. Введите имя, которое следует присвоить файлу конфигурации.

Шаг 4. Нажмите клавишу **Enter** для подтверждения каждого последующего действия.

USB-порты на маршрутизаторе Cisco

Функция хранения с использованием универсальной последовательной шины (USB) обеспечивает поддержку USB-накопителей отдельными моделями маршрутизаторов Cisco. Поддержка USB-накопителей обеспечивает дополнительные функции хранения и возможность использования дополнительного загрузочного устройства. Образы, конфигурации и другие файлы можно копировать с USB-накопителя Cisco и на него, и это так же надежно, как хранение и получение файлов с помощью карты Compact Flash. Кроме того, модульные маршрутизаторы с интегрированными сетевыми сервисами могут загружать любой образ программного обеспечения Cisco IOS, сохраненный на USB-накопитель. Теоретически на USB-накопителе может храниться несколько копий Cisco IOS и несколько конфигураций маршрутизатора. На рисунке показаны порты USB маршрутизатора Cisco 4321.



Используйте команду **dir** для отображения содержимого USB-накопителя, как это показано в примере

```
Router# dir usbflash0:
Directory of usbflash0:/
1 -rw- 30125020 Dec 22 2032 05:31:32 +00:00 c3825-entservicesk9-mz.123-14.T
63158272 bytes total (33033216 bytes free)
```

Использование USB для резервного копирования и восстановления конфигурации

Во время резервного копирования через USB-порт рекомендуется запустить команду **show file systems**, чтобы проверить наличие USB-накопителя и подтвердить его имя, как показано в примере.

```
R1# show file systems
File Systems:
      Size(b)     Free(b)   Type  Flags  Prefixes
      -          - opaque  rw    archive:
      -          - opaque  rw    system:
      -          - opaque  rw    tmpsys:
      -          - opaque  rw    null:
      -          - network rw    tftp:
*   256487424   184819712 disk   rw    flash0: flash:# 
      -          - disk   rw    flash1:
262136       249270   nvram  rw    nvr am:
      -          - opaque  wo    syslog:
      -          - opaque  rw    xmodem:
      -          - opaque  rw    ymodem:
      -          - network rw    rcp:
      -          - network rw    http:
      -          - network rw    ftp:
      -          - network rw    scp:
      -          - opaque  ro    tar:
      -          - network rw    https:
      -          - opaque  ro    cns:
4050042880   3774152704 usbflash rw    usbflash0:
R1#
```

Обратите внимание, что последняя строка вывода показывает порт USB и имя: «usbflash0».

Далее используйте команду **copy run usbflash0:** /, чтобы скопировать файл конфигурации на USB-накопитель. Обязательно используйте то имя флеш-накопителя, которое указано в файловой системе. Косую черту вводить необязательно (она обозначает корневой каталог USB-накопителя).

IOS запросит имя файла. Если файл уже существует на USB-накопителе, маршрутизатор запросит операцию записи, как показано в примере.

При копировании на USB-устройство флэш-памяти, без ранее существующего файла будет отображаться следующий вывод.

```
R1# copy running-config usbflash0:
Destination filename [running-config]? R1-Config
5024 bytes copied in 0.736 secs (6826 bytes/sec)
```

При копировании на USB-устройство флэш-памяти, с тем же файлом конфигурации уже на диске будет отображаться следующий вывод.

```
R1# copy running-config usbflash0:
Destination filename [running-config]? R1-Config
%Warning:There is a file already existing with this name
Do you want to over write? [confirm]
5024 bytes copied in 1.796 secs (2797 bytes/sec)
R1#
```

Используйте команду **dir** для просмотра файла на USB-накопителе, а также команду **more** для просмотра содержимого, как показано в примере.

```
R1# dir usbflash0:/  
Directory of usbflash0:/  
    1  drw-      0  Oct 15 2010 16:28:30 +00:00  Cisco  
   16 -rw-  5024  Jan  7 2013 20:26:50 +00:00  R1-Config  
4050042880 bytes total (3774144512 bytes free)  
R1#  
R1# more usbflash0:/R1-Config  
!  
! Last configuration change at 20:19:54 UTC Mon Jan  7 2013 by  
admin version 15.2  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname R1  
!  
boot-start-marker  
boot-end-marker  
!  
logging buffered 51200 warnings  
!  
no aaa new-model  
!  
no ipv6 cef  
R1#
```

Восстановление конфигураций с USB-накопителя

Чтобы скопировать файл обратно на устройство, потребуется внести изменения в файл R1-Config на USB-носителе с помощью текстового редактора. Если предположить, что именем файла будет **R1-Config**, используйте команду **copy usbflash0:/R1-Config running-config**, чтобы восстановить текущую конфигурацию.

Процедура восстановления пароля

Пароли на устройствах служат для защиты от несанкционированного доступа. Если пароль зашифрован (как, например, секретные пароли для входа в режим настройки), то после восстановления его необходимо заменить. В зависимости от устройства, детали процедуры восстановления пароля варьируются. Однако все процедуры восстановления пароля основаны на том же принципе:

- Шаг 1.** Войдите в режим ROMMON.
- Шаг 2.** Измените значение регистра конфигурации.
- Шаг 3.** Скопируйте startup-config в running-config.
- Шаг 4.** Изменить пароль
- Шаг 5.** Сохраните running-config как новый startup-config.
- Шаг 6.** Перезагрузите устройство.

Для восстановления пароля необходим доступ с консоли к устройству через терминал или ПО эмуляции терминала на ПК. Для доступа к устройству используются следующие настройки терминала:

- Скорость передачи данных 9600 бод
- Без бита четности
- 8 бит данных;
- 1 стоповый бит;

- Без управления потоком

Шаг 1. Войдите в режим ROMMON.

Шаг 2. Измените значение регистра конфигурации.

Шаг 3. Скопируйте startup-config в running-config.

Шаг 4. Измените пароль

Шаг 5. Сохраните running-config как

Шаг 1. Войдите в режим ROMMON.

Шаг 2. Измените значение регистра конфигурации.

Шаг 3. Скопируйте startup-config в running-config.

Шаг 4. Измените пароль

Шаг 5. Сохраните running-config как новый startup-config.

Шаг 6. Перезагрузите устройство.

Шаг 1. Войдите в режим ROMMON..

При наличии консольного доступа пользователь может войти в режим ROMMON, используя специальную комбинацию клавиш во время процесса загрузки или вынув внешнюю флеш-память, когда устройство отключено. При успешном выполнении отображается подсказка **rommon 1 >**, как показано в примере.

Примечание: В терминале PuTTY используется комбинация клавиш Ctrl+Break. Список стандартных комбинаций клавиш для других эмуляторов терминалов и операционных систем можно найти в интернете.

Readonly ROMMON initialized

```
monitor: command "boot" aborted due to user interrupt  
rommon 1 >
```

Шаг 2. Измените значение регистра конфигурации.

Программное обеспечение ROMMON поддерживает ряд основных команд, например **confreg**. Команда **confreg 0x2142** позволяет установить для регистра конфигурации значение 0x2142. Если значение регистра конфигурации равно 0x2142, устройство будет игнорировать файл загрузочной конфигурации во время запуска. В файле загрузочной конфигурации хранятся забытые пароли. Изменив значение регистра конфигурации на 0x2142, введите **reset** в командной строке, чтобы перезапустить устройство. Во время перезапуска устройства и распаковки IOS введите комбинацию клавиш прерывания текущего процесса. В примере показан экран терминала маршрутизатора 1941, который вошел в режим ROMMON после отправки сигнала "break" во время процесса загрузки.

```
rommon 1 > confreg 0x2142  
rommon 2 > reset
```

```
System Bootstrap, Version 15.0(1r)M9, RELEASE SOFTWARE (fc1)  
Technical Support: http://www.cisco.com/techsupport  
Copyright (c) 2010 by cisco Systems, Inc.  
( дальне выходные данные опущены)
```

Шаг 1. Войдите в режим ROMMON.

Шаг 2. Измените значение регистра конфигурации.

Шаг 3. Скопируйте startup-config в running-config.

Шаг 4. Измените пароль

Шаг 5. Сохраните running-config как

Шаг 3. Скопируйте startup-config в running-config.

После перезагрузки устройства скопируйте текущую конфигурацию в загрузочную конфигурацию, используя команду **copy startup-config running-config**, как показано в примере. Обратите внимание, что приглашение маршрутизатора изменилось на **R1#** так как имя хоста установлено в R1 в startup-config.

ВНИМАНИЕ! Не вводите **copy running-config startup-config**. Эта команда удалит исходную загрузочную конфигурацию.

```
Router# copy startup-config running-config  
Destination filename [running-config]?
```

```
1450 bytes copied in 0.156 secs (9295 bytes/sec)  
R1#
```

Шаг 1. Войдите в режим ROMMON.

Шаг 2. Измените значение регистра конфигурации.

Шаг 3. Скопируйте startup-config в running-config.

Шаг 4. Измените пароль

Шаг 4. Измените пароль.

Поскольку вы находитесь в привилегированном режиме EXEC, вы можете настроить все необходимые пароли, как показано в примере.

Пароль **Примечание:** The password **cisco** является ненадежным и используется здесь только для примера.

```
R1# configure terminal
```

Введите построчно команды настройки. В конце нажмите CRTL/Z.

```
R1(config)# enable secret cisco
```

Шаг 1. Войдите в режим ROMMON.

Шаг 2. Измените значение регистра конфигурации.

Шаг 3. Скопируйте startup-config в running-config.

Шаг 4. Измените пароль

Шаг 5. Сохраните running-config как новый startup-config.

Шаг 5. Сохраните running-config как новый startup-config.

После настройки новых паролей измените значение регистра конфигурации обратно на 0x2102 с помощью команды **config-register 0x2102** в режиме глобальной конфигурации. Сохраните текущую конфигурацию в файле загрузочной конфигурации, как показано в примере.

```
R1(config)# config-register 0x2102
```

```
R1(config)# end
```

```
R1# copy running-config startup-config
```

```
Destination filename [startup-config]?
```

```
Building configuration...
```

```
[OK]
```

```
R1#
```

Шаг 1. Войдите в режим ROMMON.

Шаг 2. Измените значение регистра конфигурации.

Шаг 3. Скопируйте startup-config в running-config.

Шаг 4. Измените пароль

Шаг 5. Сохраните running-config как новый startup-config.

Шаг 6. Перезагрузите устройство.

Шаг 6. Перезагрузите устройство.

Перезагрузите устройство, как показано в примере. Теперь для аутентификации на устройстве используется новый пароль. С помощью команд **show** убедитесь, что все конфигурации сохранены. Например, убедитесь, что после восстановления пароля не отключены необходимые интерфейсы.

Подробные инструкции по восстановлению пароля на конкретном устройстве можно найти в интернете.

```
R1# reload
```

Восстановление пароля

```
 Readonly ROMMON initialized

monitor: command "boot" aborted due to user interrupt
rommon 1 > confreg 0x2142
rommon 2 > reset

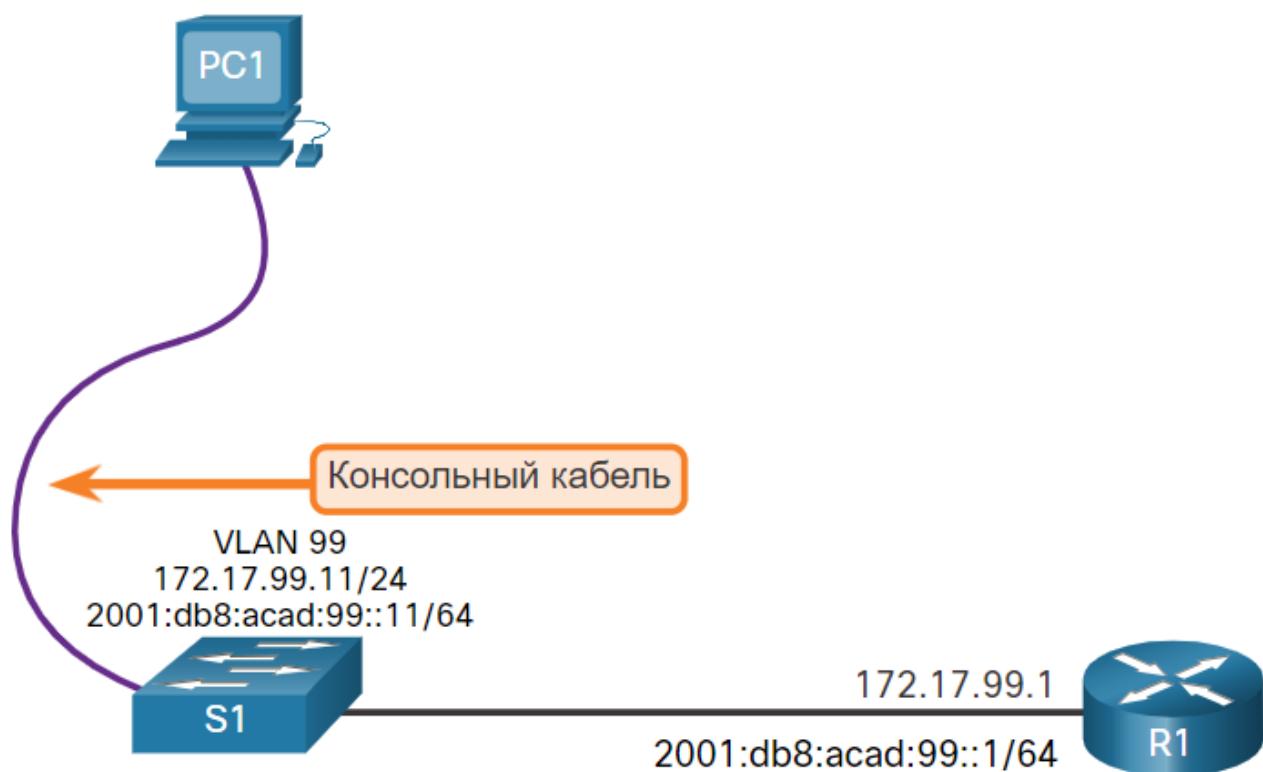
System Bootstrap, Version 15.0(1r)M9, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2010 by cisco Systems, Inc.
```

```
Router# copy startup-config running-config
Destination filename [running-config]?

1450 bytes copied in 0.156 secs (9295 bytes/sec)
Router# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# enable secret cisco
Router(config)# config-register 0x2102
Router(config)# end
Router# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router# reload
```

Доступ к управлению коммутатором

Чтобы подготовить коммутатор для доступа к удаленному управлению, он должен быть настроен с IP-адресом и маской подсети. Имейте в виду, что для управления коммутатором из удаленной сети на коммутаторе должен быть настроен на шлюз по умолчанию. Это очень похоже на настройку IP-адресов на хост-устройствах. На рисунке виртуальный интерфейс коммутатора (SVI) на S1 должен быть назначен IP-адрес. SVI — это виртуальный интерфейс, а не физический порт коммутатора. Кабель консоли используется для подключения к ПК, так что коммутатор может быть изначально настроен.



Начальный загрузчик коммутатора

Доступ к начальному загрузчику

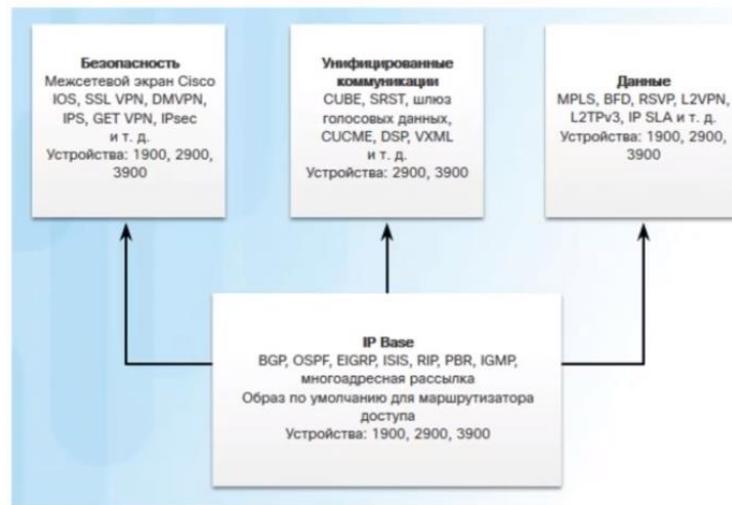
- Подключиться по консоли
- Отключить кабель питания
- Подключить кабель питания и в течение 15 сек удерживать Mode
- Подождать сигнал индикатора SYST: мигнёт жёлтым, а затем загорится постоянный зелёный
- Увидеть запрос switch:
- Команда set показывает значение BOOT, команда flash_init инициализирует flash

```
Switch# dir flash:  
Directory of flash:/  
  
2 -rwx 11607161 Mar 1 2013 03:10:47 +00:00 c2960-lanbasek9-mz.150-2.SE.bin  
3 -rwx 1809 Mar 1 2013 00:02:48 +00:00 config.text  
5 -rwx 1919 Mar 1 2013 00:02:48 +00:00 private-config.text  
6 -rwx 59416 Mar 1 2013 00:02:49 +00:00 multiple-fs  
  
32514048 bytes total (20841472 bytes free)  
Switch#
```

2

d. Управление образами IOS. Имена файлов образов. Резервное копирование и восстановление образов IOS.

Services on Demand



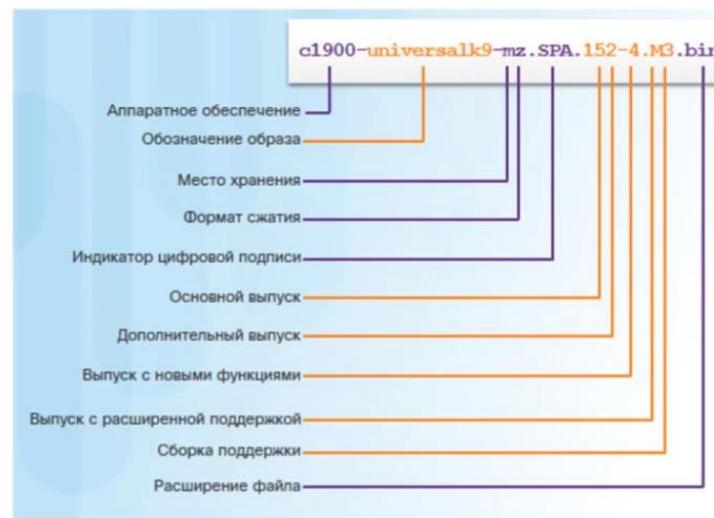
Маршрутизатор поставляется с одной универсальной Cisco IOS, и для включения пакетов определенных наборов функций используется лицензия

Для маршрутизаторов Cisco выпускаются два типа универсальных образов:

- Универсальные образы, имя которых содержит «**universalk9**», содержат все функции Cisco IOS, в том числе мощные средства шифрования полезной нагрузки, такие как IPSec VPN, SSL VPN и Secure Unified Communications
- Универсальные образы, имя которых содержит «**universalk9_pre**», — это шифровальные средства, предоставляемые системой активации Cisco (Cisco Software Activation), которые удовлетворяют экспортным ограничениям на криптографические средства

Начиная с 15 версии ios маршрутизаторы поставляются с универсальной версией cisco ios. Разные модули отдельно активируются. Это называется услуги по требованию.

Имена файлов образов IOS



```
R1# show flash0:  
-# - --length-- -----date/time----- path  
  
8 68831808 Apr 2 2013 21:29:58 +00:00 c1900-universalk9-ms.SPA.152-4.M3.bin  
182394880 bytes available (74092544 bytes used)  
R1#
```

mz — это наиболее распространенное обозначение для формата участка памяти и сжатия. Первая буква указывает, где маршрутизатор будет хранить образ во время работы.

Возможные места:

- f — флеш-память
- m — ОЗУ
- g — ПЗУ
- l — переместимый

Образ может быть сжат в формате z (zip) или x (mzip)

Основной выпуск (версия ios) – 15. 2 – дополнительный выпуск.

SPA – цифровая подпись cisco.

Использование TFTP-серверов для хранения резервной копии

По мере роста сети образы и файлы конфигурации Cisco IOS могут храниться на центральном TFTP-сервере, как показано на рисунке. Это позволяет контролировать количество образов IOS и их версии, а также нуждающиеся в обслуживании файлы настроек ОС.



isr4200-universalk9_ias.16.09.04.SPA.bin

Производственные интерсети обычно занимают обширные области и содержат несколько маршрутизаторов. В любой сети рекомендуется сохранить резервную копию образа ОС Cisco IOS на случай повреждения или случайного удаления образа системы на маршрутизаторе.

Маршрутизаторам, находящимся на большом расстоянии друг от друга, необходим источник или место для хранения резервных образов программного обеспечения Cisco IOS. Использование сетевого TFTP-сервера позволяет загружать файлы образов и конфигураций по сети. Такой TFTP-сервер может быть маршрутизатором, рабочей станцией или хостом.

Пример создания резервной копии образа IOS на TFTP-сервере

На рисунке администратор сети хочет создать резервную копию текущего файла образа на маршрутизаторе (isr4200-universalk9_ias.16.09.04.SPA.bin) на сервере TFTP по адресу 172.16.1.100.



Шаг 1. Отправьте эхо-запрос на TFTP-сервер.

Шаг 1. Отправьте эхо-запрос на TFTP-сервер.

Убедитесь в доступности TFTP-сервера. Как показано в примере, отправьте эхо-запрос на TFTP-сервер для проверки подключения.

Шаг 2. Проверьте размер образа на флэш-памяти.

Шаг 3. Скопируйте образ на TFTP-сервер

```
R1# ping 172.16.1.100
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.100, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5),
round-trip min/avg/max = 56/56/56 ms
```

Шаг 1. Отправьте эхо-запрос на TFTP-сервер.

Шаг 2. Проверьте размер образа на флэш-памяти.

Убедитесь, что на диске TFTP-сервера достаточно места для размещения образа программного обеспечения Cisco IOS. Для определения размера файла образа Cisco IOS можно воспользоваться командой **show flash0**: на маршрутизаторе. Длина файла на примере составляет 517153193 байт.

Шаг 3. Скопируйте образ на TFTP-сервер

```
R1# show flash0:
--length-- -----date/time----- path
8 517153193 Apr 2 2019 21:29:58 +00:00
    isr4200-universalk9_ias.16.09.04.SPA.bin
( дальне выходные данные опущены)
```

Шаг 1. Отправьте эхо-запрос на TFTP-сервер.

Шаг 3. Скопируйте образ на TFTP-сервер.

Скопируйте образ на TFTP-сервер с помощью команды **copy source-url destination-url**.

После выполнения команды с использованием заданных URL-адресов источника и назначения пользователь получит запрос на ввод имени файла источника, адреса удаленного узла и имени файла назначения. Как правило, вы вводите **Enter**, чтобы принять имя исходного файла в качестве имени файла назначения. После этого начнется передача.

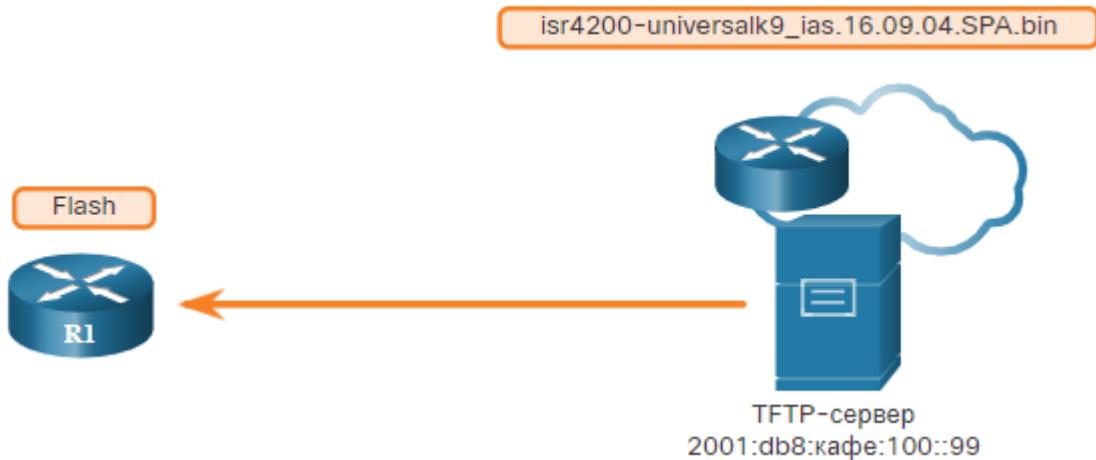
Шаг 3. Скопируйте образ на TFTP-сервер

```
R1# copy flash: tftp:
Source filename []? isr4200-universalk9_ias.16.09.04.SPA.bin
Address or name of remote host []? 172.16.1.100
Destination filename [isr4200-universalk9_ias.16.09.04.SPA.bin]?
Writing isr4200-universalk9_ias.16.09.04.SPA.bin...
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
( дальне выходные данные опущены)
517153193 bytes copied in 863.468 secs (269058 bytes/sec)
```

Пример копирования образа IOS на устройство

Cisco постоянно выпускает новые выпуски Cisco IOS, чтобы устранить возникающие ошибки и предложить новые функции. В этом примере для передачи используется IPv6 в целях демонстрации того, что TFTP может использоваться в IPv6-сетях.

На рисунке показан процесс копирования образа Cisco IOS с TFTP-сервера. Новый файл образа (isr4200-universalk9_ias.16.09.04.SPA.bin) будет скопирован на маршрутизатор с TFTP-сервера по адресу 2001:DB8:CAFE:100::99.



Шаг 1. Отправьте эхо-запрос на TFTP-сервер.

Шаг 1. Отправьте эхо-запрос на TFTP-сервер.

Убедитесь в доступности TFTP-сервера. Как показано в примере, отправьте эхо-запрос на TFTP-сервер для проверки подключения.

Шаг 2. Проверьте количество свободной флеш-памяти.

```
R1# ping 2001:db8:cafe:100::99
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:CAFE:100::99,
timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5),
round-trip min/avg/max = 56/56/56 ms
```

Шаг 3. Скопируйте новый образ IOS в флэш-память.

Шаг 1. Отправьте эхо-запрос на TFTP-сервер.

Шаг 2. Проверьте количество свободной флеш-памяти.

Убедитесь, что на маршрутизаторе доступен необходимый объем флеш-памяти. Объем свободной флеш-памяти можно проверить с помощью команды **show flash**. Сравните свободный объем флеш-памяти с размером нового файла образа. В примере команда **show flash**: используется для проверки свободной флеш-памяти. Объем свободной флеш-памяти в данном примере составляет 6294806528 байт.

Шаг 2. Проверьте количество свободной флеш-памяти.

Шаг 3. Скопируйте новый образ IOS в флэш-память.

```
R1# show flash:
-# - --length-- -----date/time----- path
( дальне выходные данные опущены)
6294806528 bytes available (537251840 bytes used)
R1#
```

Шаг 1. Отправьте эхо-запрос на TFTP-сервер.

Шаг 2. Проверьте количество свободной флеш-памяти.

Шаг 3. Скопируйте новый образ IOS в флеш-память.

Шаг 3. Скопируйте новый образ IOS в флэш-память.

Скопируйте файл образа IOS с TFTP-сервера на маршрутизатор с помощью команды **copy**, показанной в примере. После выполнения этой команды с указанными URL-адресами источника и назначения пользователь получит запрос на ввод IP-адреса удаленного узла, имен файлов источника и назначения. Как правило, вы вводите **Enter**, чтобы принять имя исходного файла в качестве имени файла назначения. После этого начнется передача файла.

```
R1# copy tftp: flash:  
Address or name of remote host []?2001:DB8:CAFE:100::99  
Source filename []? isr4200-universalk9_ias.16.09.04.SPA.bin  
Destination filename [isr4200-universalk9_ias.16.09.04.SPA.bin]?  
Accessing tftp://2001:DB8:CAFE:100::99/ isr4200-universalk9_ias.16.09.04.SPA.bin...  
Loading isr4200-universalk9_ias.16.09.04.SPA.bin  
from 2001:DB8:CAFE:100::99 (via  
GigabitEthernet0/0/0): !!!!!!!!!!!!!!!  
  
[OK - 517153193 bytes]  
517153193 bytes copied in 868.128 secs (265652 bytes/sec)
```

Команда "boot system"

Чтобы установить скопированный образ IOS после того, как он сохранен во флеш-памяти маршрутизатора, настройте маршрутизатор на загрузку нового образа во время запуска с помощью команды **boot system**, как показано в примере. Сохраните конфигурацию. Перезагрузите маршрутизатор, чтобы он загрузился с новым образом

```
R1# configure terminal  
R1(config)# boot system flash0:isr4200-universalk9_ias.16.09.04.SPA.bin  
R1(config)# exit  
R1#  
R1# copy running-config startup-config  
R1#  
R1# reload  
Proceed with reload? [confirm]  
  
*Mar 1 12:46:23.808: %SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload Command.
```

После запуска маршрутизатора загрузчик ищет в файле загрузочной конфигурации (startup configuration) команды **boot system** с указанными в них именем и расположением образа Cisco IOS, который он должен будет загрузить. Чтобы обеспечить отказоустойчивую загрузку, можно ввести несколько команд **boot system**.

Если в конфигурации нет команд **boot system**, маршрутизатор по умолчанию загружает первый допустимый образ Cisco IOS из флеш-памяти и запускает его.

После загрузки маршрутизатора убедитесь, что новый образ загрузился. Используйте для этого команду **show version**, как показано в примере.

```
R1# show version
Cisco IOS XE Software, Version 16.09.04
Cisco IOS Software [Fuji], ISR Software (X86_64_LINUX_IOSD-UNIVERSALK9_IAS-M), Version 16.9.4, RELEASE
SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2019 by Cisco Systems, Inc.
Compiled Thu 22-Aug-19 18:09 by mcpre
Cisco IOS-XE software, Copyright (c) 2005-2019 by cisco Systems, Inc.
All rights reserved. Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0. The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0. For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.

ROM: IOS-XE ROMMON
Router uptime is 2 hours, 19 minutes
Uptime for this control processor is 2 hours, 22 minutes
System returned to ROM by PowerOn
System image file is "flash:isr4200-universalk9_ias.16.09.04.SPA.bin"
(output omitted)
```

9. Проектирование сети:

а. Конвергентные сети.

Необходимость масштабирования сети

Наш цифровой мир непрерывно развивается. Возможность доступа в Интернет и корпоративные сети больше не ограничиваются территорией офиса, географическим местоположением или часовыми поясами. В современных компаниях сотрудники могут получить доступ к необходимым ресурсам и информации практически из любой точки мира, в любое время и с любого устройства. Подобные требования приводят к необходимости выстраивать сети нового поколения — сети, обеспечивающие большую безопасность, надежность и доступность.

Сети нового поколения должны не только соответствовать существующим ожиданиям и поддерживать современное оборудование, но также взаимодействовать с устаревшими платформами. Организации все больше полагаются на свои сети, предоставляющие критически важные сервисы. По мере роста и развития предприятия его штат увеличивается, открываются новые филиалы и компания выходит на международный уровень. Эти изменения непосредственно влияют на требования сети, которая должна быть в состоянии масштабироваться для удовлетворения потребностей бизнеса.

Сеть должна обеспечивать обмен сетевым трафиком различного вида, включая файлы данных, электронные сообщения, IP-телефонию и работу с видео для нескольких бизнес-подразделений. Все корпоративные сети должны иметь возможность выполнять следующие действия:

- поддерживать работу критически важных приложений;
- поддерживать трафик в конвергентных сетях;
- соответствовать различным требованиям бизнеса;
- обеспечивать возможность централизованного административного управления.

Конвергентные сети



Передача голоса и видео по одной сети

Дополнительные сервисы:

- Управление вызовами
- Голосовые сообщения
- Мобильная связь
- Автоответчик

++ Одна физическая сеть для разного типа трафика

Но! Требуется продуманная архитектура и структурированное проектирование

Конвергентные – современные типы сетей.

Передаем разные типы трафика через одну сеть, не строим выделенные каналы передачи данных для разного типа трафика (трафик данных, голосовой трафик, видео трафик).

Надо гарантировать, чтобы не было задержек, искажений. Это создает дополнительные требования к сети. Преимущество в том, что нужна лишь одна сеть чтобы передавать разные типы трафика.

Минусы – требуется продуманная архитектура и хорошее структурирование. Надо хорошо подумать как обеспечить нормальную передачу трафика.

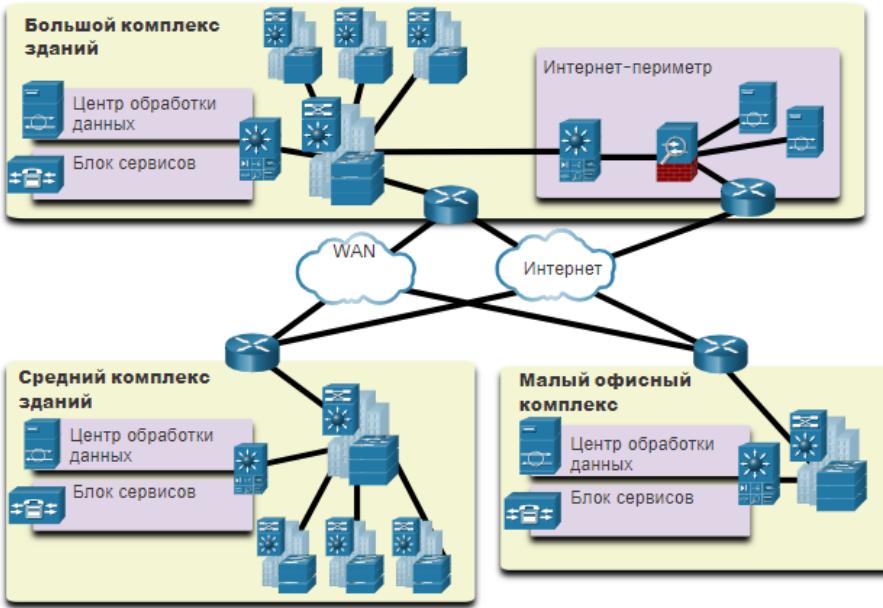
b. Модели построения сетей без границ. Описание, характеристики. Задачи уровней.

Коммутируемые сети без границ(borderless network)

Ввиду растущих требований к объединенным сетям развитие последних требует нового архитектурного подхода, учитывающего внедрение интеллекта, упрощение операций и масштабируемость сети в зависимости от потребностей будущих пользователей. Одна из последних разработок в области проектирования сетей — концепция сети без границ Cisco.

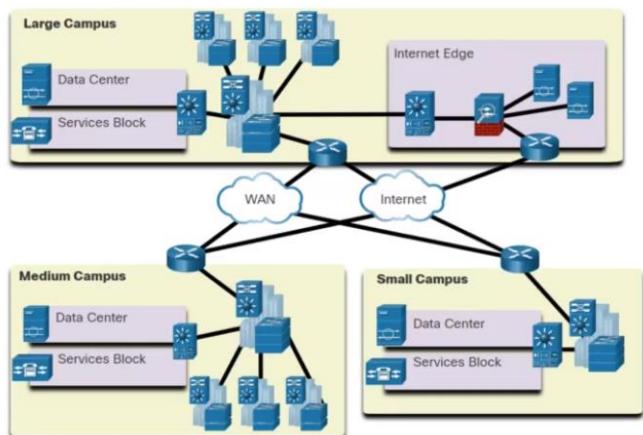
Сеть без границ Cisco — это сетевая архитектура, которая объединяет инновационную идею и проектирование. Основываясь на этой архитектуре, организации могут обеспечить поддержку сети без границ, которая безопасно, надежно и удобно связывает пользователей любых устройств, в любое время и в любом месте. Эта сетевая архитектура разработана специально для решения информационных и деловых вопросов, например поддержки объединенной сети и непрерывно меняющихся схем организации работ.

Сеть без границ Cisco предоставляет платформу для объединения средств проводного и беспроводного доступа, включая применение политик, разграничение доступа и управление производительностью устройств различных типов. Сеть без границ, использующая такую архитектуру, строится на основе масштабируемой и отказоустойчивой иерархической инфраструктуры аппаратного обеспечения.



Объединяя эту аппаратную инфраструктуру с программными решениями на основе политик, сеть без границ Cisco предоставляет два главных набора сервисов: сетевые сервисы и сервисы для пользователей и оконечных устройств под эгидой интегрированного решения управления. В результате разные сетевые элементы могут совместно работать, пользователи получают повсеместный доступ к ресурсам в любое время, обеспечиваются оптимизация, масштабируемость и безопасность.

Borderless Networks



- Предоставляет платформу для объединения средств проводного и беспроводного доступа, включая применение политик, разграничение доступа и управление производительностью устройств различных типов
- Строится на основе масштабируемой и отказоустойчивой иерархической инфраструктуры аппаратного обеспечения
- Пользователи получают повсеместный доступ к ресурсам в любое время
- Способна передавать конвергированные данные

Иерархия в коммутируемой сети без границ

Принцип построения

Иерархичность: у каждого уровня своя роль

Модульность: легко масштабировать и внедрять новые сервисы

Отказоустойчивость: бесперебойная работа сети

Гибкость: рациональное распределение ресурсов

Для обеспечения максимальной доступности, гибкости, безопасности и удобства эксплуатации коммутируемой сети без границ в процессе ее создания необходимо следовать четким принципам

проектирования. Руководство по проектированию коммутируемой сети без границ построено на следующих принципах:

- **Иерархичность** — упрощает понимание роли каждого устройства на каждом уровне, обеспечивает поддержку в процессе развертывания, эксплуатации и управления, а также снижает количество неполадок на каждом уровне.
- **Модульность** - позволяет при необходимости легко расширять сеть и внедрять интегрированные сервисы по запросу.
- **Отказоустойчивость** - обеспечивает бесперебойную работу сети в соответствии с ожиданиями пользователей.
- **Гибкость** — обеспечивает рациональное распределение нагрузки трафика за счет использования всех сетевых ресурсов.

Перечисленные принципы зависят друг от друга. Именно поэтому крайне важно понимать природу и способы их взаимодействия в рамках коммутируемой сети. Иерархическое проектирование коммутируемой сети без границ создает основу, которая позволяет сетевым разработчикам объединять функции безопасности, мобильности и унифицированной коммуникации. Основой иерархического проектирования сетей кампусного типа являются дважды проверенные и одобренные к применению трехуровневые и двухуровневые модели.

Три основных уровня в рамках рассматриваемых многоуровневых проектов представляют собой уровни доступа, распределения и ядра. Каждый уровень можно рассматривать как четкий, структурированный модуль кампусной сети, наделенный определенными ролями и функциями. Введение принципа модульности в иерархическую архитектуру сети дает дополнительную гарантию — кампусные сети модульных конструкций демонстрируют большую надежность и гибкость в отношении обеспечения важнейших сетевых сервисов. Модульность также способствует расширению сети и внесению изменений, происходящих с течением времени.

Функции уровней доступа, распределения и ядра

Уровень доступа

Уровень доступа представляет периметр сети, где трафик входит или покидает сеть кампусного типа. Традиционно основная функция коммутатора уровня доступа заключается в обеспечении пользователю сетевого доступа. Коммутаторы уровня доступа подключаются к коммутаторам уровня распределения, которые реализуют технологии сетевой инфраструктуры, такие как маршрутизация, качество обслуживания и безопасность.

Уровень распределения

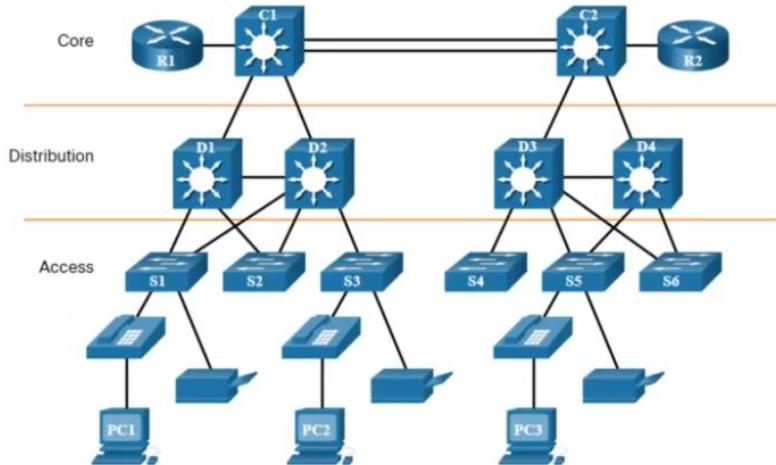
Уровень распределения взаимодействует между уровнем доступа и уровнем ядра для обеспечения многих важных функций:

- возможность агрегации больших проводных сетей в коммуникационном шкафу;
- агрегация широковещательных доменов уровня 2 и границ маршрутизации уровня 3;
- предоставление доступа интеллектуальной коммутации, маршрутизации и функций политики доступа к остальной части сети;
- обеспечение высокого уровня доступности ядра для конечных пользователей и наличие маршрутов равной стоимости посредством резервных коммутаторов уровня распределения;
- предоставление дифференцированных услуг приложениям с различными классами обслуживания по периметру сети.

Уровень ядра

Уровень ядра — это сетевая магистраль. Данный уровень объединяет несколько уровней сети кампусного типа. Уровень ядра служит агрегатором для всех устройств уровня распределения и связывает кампус вместе с остальной частью сети. Основная задача уровня ядра заключается в обеспечении изоляции сбоев и высокоскоростного магистрального подключения.

Уровни доступа, распределения и ядра



Уровень доступа (access layer): предоставление доступа пользователю

Уровень распределения (distribution layer):

- Маршрутизация
- Агрегация трафика
- Политики доступа (QoS)

Уровень ядра (core layer): магистраль сети

Уровень распределения пересыпает трафик внутри сети либо на уровень ядра (во внешнюю сеть). Устройства могут фильтровать трафик, обеспечивать качество обслуживания. Лучше использовать коммутаторы 3 уровня – маршрутизуемые коммутаторы. Это коммутатор, который умеет маршрутизировать. Отличие от маршрутизатора в том, что коммутатор 3 уровня быстрее передает трафик, так как передают трафик аппаратно. Маршрутизаторы передают трафик программно, поэтому медленнее (идет обработка через ЦП для обработки каждого пакета).

Через уровень ядра происходит соединение с другими сетями.

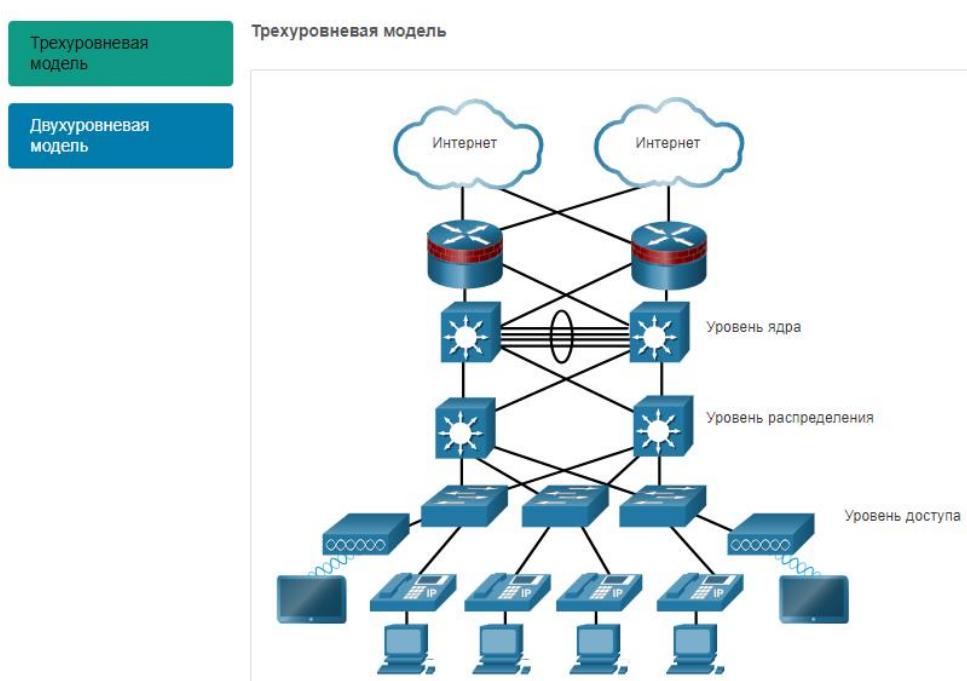
Нужно резервировать устройства на уровне распределения.

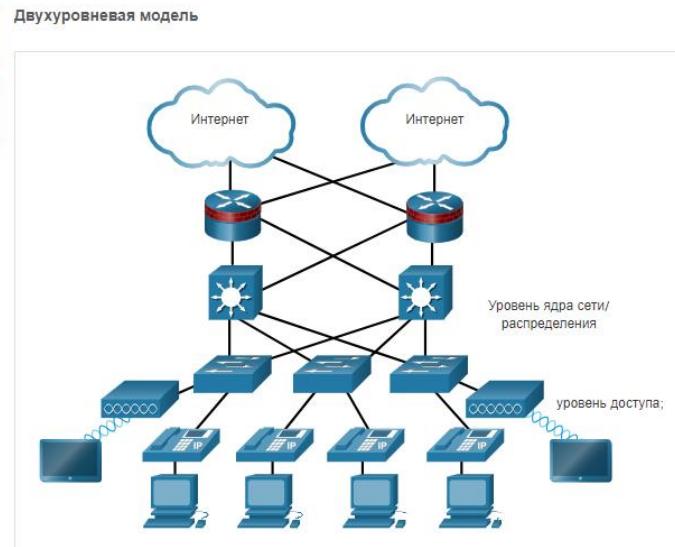
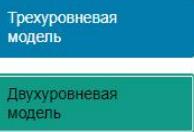
Домен отказоустойчивости – набор тех устройств, которые потеряют доступ к сети в случае сбоя.

На 100% обеспечить отказоустойчивость невозможно!

Гибкость – равномерно распределить пользователей на коммутаторы.

Обычно 100мб порты для пользователя и 1000мб на исходящий интерфейс.

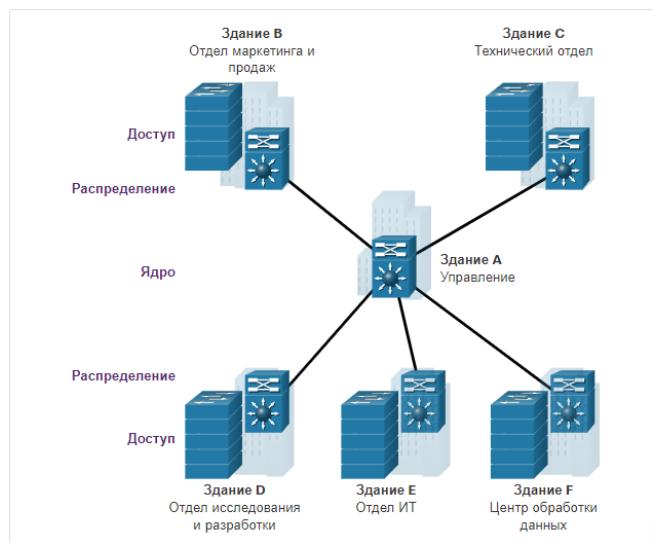




Если сеть не слишком большая, то лучше использовать 2-х уровневую модель, чтобы не тратить ресурсы для закупки дополнительного оборудования.

Пример трехуровневой сети

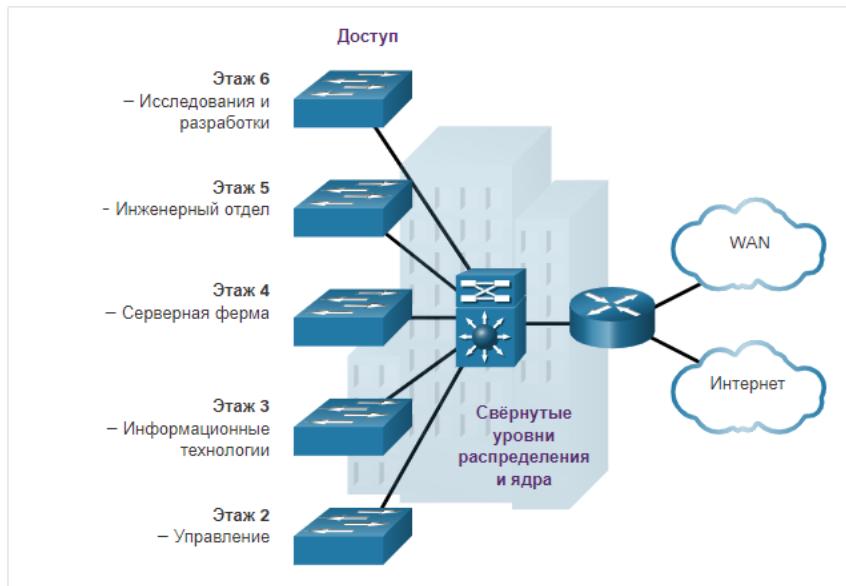
На рисунке представлена трехуровневая архитектура сети кампусного типа для организаций, в которых уровни доступа, распределения и ядра являются отдельными уровнями. Для создания упрощенного, масштабируемого, рентабельного и эффективного проекта физической структуры кабельной сети рекомендуется выстраивать физическую топологию сети по типу расширенной звезды от центрального здания до всех остальных зданий в рамках одного комплекса.



Использовать если как минимум несколько зданий.

Пример двухуровневой сети

В некоторых случаях, когда отсутствует высокая масштабируемость физической инфраструктуры или сети, разграничение уровней распределения и ядра не требуется. Разделение между уровнем ядра и уровнем распределения может не понадобиться в небольшой кампусной сети, в которой количество подключенных к сети пользователей невелико, или когда подразделение кампуса состоит из одного здания. При таком варианте рекомендуется использовать альтернативную двухуровневую схему сети комплекса зданий, также называемую схемой сети со свернутым ядром, как показано на рисунке.



Плоские сети



Может быть и цепочка коммутаторов. Не будет отказоустойчивости. Сеть не иерархична. Справа пример плоской шины. Если один элемент сломался, трафик больше не передается нигде. Не масштабируются. Каждый новый пользователь добавляет трафик в сети.

Проектирование для обеспечения масштабируемости

Масштабируемость — это термин для сети, которая может расти без потери доступности и надежности.

Для поддержки крупной, средней или малой сети проектировщик должен разработать стратегию, чтобы обеспечить доступность и эффективную масштабируемость сети. Базовая стратегия проектирования сети включает в себя следующие рекомендации.

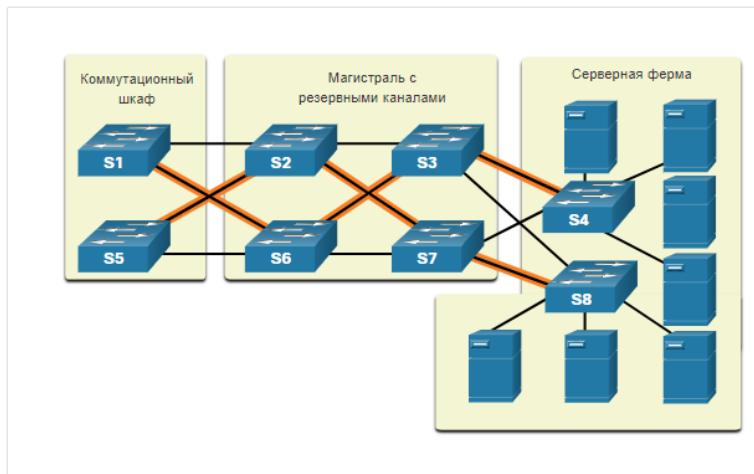
- Следует использовать расширяемое модульное оборудование или кластерные устройства, которые можно легко модернизировать для увеличения их возможностей. Для поддержки новых функций и устройств к уже существующему оборудованию можно просто добавить новые аппаратные модули, и существенная модернизация оборудования не потребуется. Некоторые устройства можно интегрировать в кластер, чтобы они работали как одно устройство. Это упрощает управление и настройку.
- Иерархическую сеть следует проектировать с учетом возможностей добавления, обновления и изменения модулей в случае необходимости, не затрагивая при этом другие функциональные

области сети. Например, можно создать отдельный уровень доступа, который можно расширять, не затрагивая уровни распределения и ядра кампусной сети.

- Используйте иерархическую стратегию адресации IPv4 и IPv6. При продуманном планировании адресации для организации поддержки дополнительных пользователей и сервисов вам не потребуется заново настраивать адреса во всей сети.
- Выберите маршрутизаторы или многоуровневые коммутаторы, чтобы ограничивать широковещательные рассылки и отфильтровывать из сети нежелательный трафик. Используйте устройства уровня 3 для фильтрации и сокращения объема трафика к ядру сети.

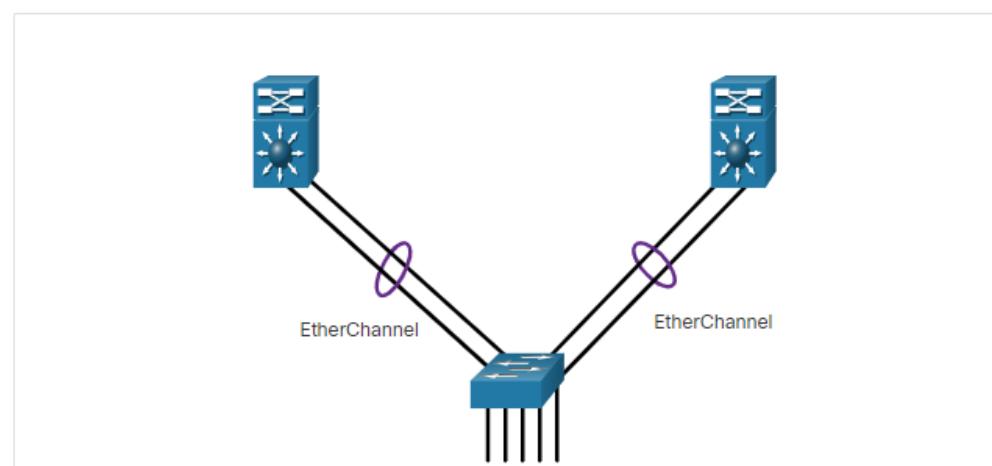
Резервные линии

Внедрите резервные каналы в сети между критически важными устройствами, а также между устройствами уровня доступа и уровня ядра;



Несколько каналов связи

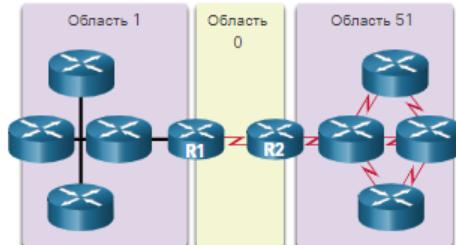
Внедрите несколько каналов связи между различными устройствами с использованием функций агрегации каналов (EtherChannel) или распределением нагрузки в соответствии с равной стоимостью в целях увеличения пропускной способности; объединение нескольких каналов Ethernet в единую конфигурацию EtherChannel с распределенной нагрузкой, что позволяет увеличить доступную пропускную способность; используйте технологию EtherChannel, если в связи с ограничениями бюджета невозможно приобрести высокоскоростные интерфейсы и оптоволоконные кабели;



Несколько интерфейсов в один канал связи, чтобы увеличить пропускную способность.

Масштабируемый протокол маршрутизации

Использование масштабируемого протокола маршрутизации и реализация в этом протоколе функций, позволяет изолировать обновления маршрутизации и минимизировать размер таблицы маршрутизации;

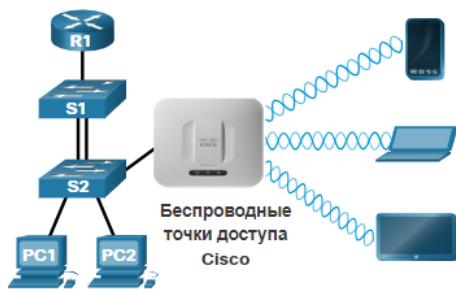


Протокол OSPF. Уметь

делить крупные сети на подсети.

Беспроводное соединение

Внедрение беспроводного подключения для поддержки мобильности и расширения.



с. Выбор оборудования при построении сети.

Характеристики коммутаторов

- Формфактор
- Плотность портов
- Питание (поддержка PoE)
- Надёжность
- Скорость портов
- Размер буфера порта
- Масштабируемость
- Стоимость



Буфер порта – количество трафика, которое может накапливать коммутатор, прежде чем его передать. Влияет на заторы в сети.

Одним из простых способов создания иерархических и масштабируемых сетей является использование подходящего оборудования для выполнения задания. Существует множество платформ коммутаторов, форм-факторов и других функций, которые следует учитывать перед выбором коммутатора.

При проектировании сети важно выбрать аппаратное обеспечение, соответствующее текущим требованиям к сети, а также обеспечить возможность расширения сети. В корпоративной сети как коммутаторы, так и маршрутизаторы играют критически важную роль в обмене данными по сети.

Кампусные коммутаторы локальной сети

Для масштабирования производительности корпоративной локальной сети используются коммутаторы уровней ядра, распределения, доступа, а также компактные коммутаторы. Эти платформы коммутации могут иметь разную конструкцию: от безвентиляторных коммутаторов с восемью фиксированными портами до коммутаторов с 13 платами, поддерживающих несколько сотен портов. К платформам коммутации для кампусных локальных сетей относятся коммутаторы Cisco серий 2960, 3560, 3650, 3850, 4500, 6500 и 6800.



Коммутаторы с облачным управлением

Коммутаторы доступа с управлением в облакной среде Cisco Meraki обеспечивают возможность виртуального стекирования коммутаторов. Они осуществляют мониторинг и настройку тысяч портов коммутации по сети Интернет без вмешательства локальных ИТ-специалистов.



Коммутаторы для центров обработки данных

Центр обработки данных должен создаваться на базе коммутаторов, обеспечивающих расширенные возможности масштабирования инфраструктуры, бесперебойную работу и гибкость транспортных решений. К платформам коммутации ЦОД относятся коммутаторы серий Cisco Nexus.



Коммутаторы операторов связи

Коммутаторы операторов связи разделяются на две категории: коммутаторы агрегации и коммутаторы доступа по Ethernet



Виртуальные сети

Виртуализация все чаще используется в современных сетях



При выборе коммутаторов сетевой администратор должен определить **формфактор коммутатора**. К формфакторам относятся конфигурации: фиксированная, модульная, со стеком и без стека.

Коммутаторы с фиксированной конфигурацией

Функции и опции коммутаторов с фиксированной конфигурацией ограничены теми, которые изначально поставляются с коммутатором.



Коммутаторы с модульной конфигурацией

Шасси модульных коммутаторов принимает линейные платы, заменяемые в полевых условиях.



Коммутаторы со стекируемой конфигурацией

Для подключения установленных в стек коммутаторов используются специальные кабели, которые позволяют им эффективно работать как один большой коммутатор.



Толщина

Высота коммутатора, измеряемая количеством стоечных модулей, также имеет значение для коммутаторов, если они устанавливаются в стойку. Например, все переключатели фиксированной конфигурации, показанные на рисунке, имеют высоту в одну стойку (1U) или 1,75 дюйма (44,45 мм).



Плотность портов

Под плотностью портов коммутатора подразумевается количество портов, доступных на одном коммутаторе. На рисунке показана плотность портов трех различных коммутаторов.

Коммутаторы с фиксированной конфигурацией поддерживают различные варианты плотности портов. Cisco Catalyst 3850 поставляются в конфигурациях 12, 24, 48 портов, как показано на рисунке. 48-портовый коммутатор предусматривает поддержку дополнительных портов для устройств малого формфактора (SFP).



Модульные коммутаторы поддерживают очень высокую плотность портов за счет добавления нескольких линейных плат портов коммутатора. Модульный коммутатор Catalyst 9400, показанный на следующем рисунке, поддерживает 384 коммутационных порта.



В крупных сетях с несколькими тысячами сетевых устройств для оптимального использования физического пространства и потребления электроэнергии требуются модульные коммутаторы высокой плотности. Без модульного коммутатора высокой плотности для работы сети потребуется множество коммутаторов с фиксированной конфигурацией, чтобы поддерживать все устройства, которым нужен доступ к сети. При таком подходе необходимо большое количество розеток электропитания и много места в коммутационном шкафу.

Проектировщик сети также должен учитывать проблему «узких мест» восходящего канала: при большом числе коммутаторов с фиксированной конфигурацией для достижения заданной производительности может потребоваться большое количество дополнительных портов с целью агрегации пропускной способности между этими коммутаторами. При использовании одного модульного коммутатора агрегация пропускной способности не представляет большой проблемы, поскольку объединительная плата шасси обеспечивает пропускную способность, необходимую для поддержки устройств, подключенных к линейным платам портов коммутатора.

Скорость передачи трафика

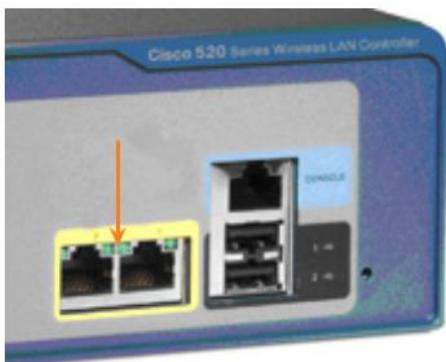
Скорость передачи трафика определяет возможную производительность коммутатора, оценивая объем данных, которые могут быть обработаны коммутатором в течение секунды. Коммутаторы классифицируются по скорости пересылки. Коммутаторы начального уровня имеют более низкую скорость передачи трафика, чем коммутаторы корпоративного уровня. При выборе коммутатора крайне важно учитывать скорость передачи трафика. Если скорость передачи трафика коммутатора слишком низкая, коммутатор не сможет обеспечить на всех своих портах обмен данными с полной скоростью, на которую рассчитана среда передачи данных. Номинальная скорость среды передачи данных — это скорость передачи данных, которую способен обеспечить каждый Ethernet-порт на коммутаторе. Скорость передачи данных может составлять 100 Мбит/с, 1 Гбит/с, 10 Гбит/с или 100 Гбит/с.

Питание по сети Ethernet

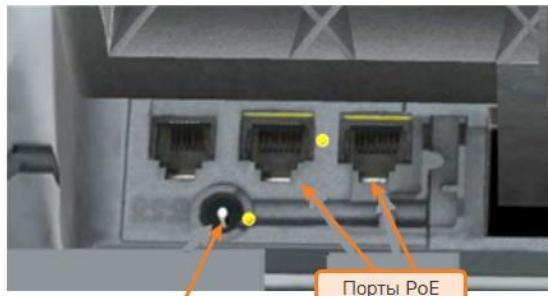
Технология PoE позволяет коммутатору осуществлять подачу питания на устройство по кабелю Ethernet. Эта функция может использоваться IP-телефонами и некоторыми беспроводными точками доступа, что позволяет устанавливать их везде, где есть кабель Ethernet. Сетевой администратор должен убедиться в том, что функции PoE действительно необходимы, поскольку коммутаторы с поддержкой PoE стоят недешево.

Коммутатор

Порты PoE выглядят точно так же, как и любой другой порт коммутатора. Выберите модель коммутатора, чтобы определить, поддерживает ли порт питание через Ethernet.



IP-телефон



Внешний источник питания

Порты PoE

WAP

Порты PoE на точке беспроводного доступа выглядят точно так же, как и любой другой порт коммутатора. Выберите модель точки беспроводного доступа, чтобы определить, поддерживает ли порт питание через Ethernet.



В таблице представлен ряд распространенных факторов, имеющих немаловажное значение при выборе коммутаторов для корпоративной сети.

Рассмотрение	Описание
Стоимость	Стоимость коммутатора будет зависеть от количества и скорости интерфейсов, поддерживаемых функций и возможностей расширения.
Плотность портов	Сетевые коммутаторы должны поддерживать соответствующее количество устройств в сети.
Питание	Теперь точки доступа, IP-телефоны и даже компактные коммутаторы получают питание через Ethernet (PoE). Кроме PoE, некоторые коммутаторы на основе шасси поддерживают резервные источники питания.
Надежность	Коммутатор должен обеспечивать непрерывный доступ к сети.
Скорость порта	Скорость подключения к сети является основным фактором выбора для всех конечных пользователей.
Буферы кадров	Способность коммутатора хранить кадры важна в сети, где могут быть перегружены порты на сервере или другие области сети.
Масштабируемость	Количество пользователей в сети, как правило, растет с течением времени; поэтому коммутатор должен обеспечить возможность для роста сети.

Требования к маршрутизатору

Маршрутизаторы используют сетевую часть (префикс) IP-адреса назначения для направления пакетов к нужному месту назначения. Они выбирают альтернативный путь, если канал не работает. IP-адрес

интерфейса локального маршрутизатора задается в IP-конфигурации для всех узлов локальной сети. Интерфейс маршрутизатора — это шлюз по умолчанию. Эффективная маршрутизация и восстановление после отказов сетевых каналов исключительно важны для доставки пакетов по месту назначения.

Маршрутизаторы выполняют и другие полезные функции:

- Они обеспечивают сдерживание широковещательных рассылок, ограничивая их до локальной сети.
- Они соединяют географически удаленные друг от друга местоположения.
- Маршрутизаторы логически группируют пользователей, которые имеют общие потребности и нуждаются в доступе к одним и тем же ресурсам, например, по отделам компании.
- Они обеспечивают повышенную безопасность, фильтруя нежелательный трафик через списки контроля доступа.

Маршрутизаторы филиала

Маршрутизаторы для филиалов, как показано на рисунке, позволяют оптимизировать сервисы филиала на базе единой платформы, обеспечивая при этом оптимальное взаимодействие с приложениями в инфраструктурах филиала и глобальной сети. Для обеспечения максимальной доступности сервисов для филиала требуется круглосуточная работоспособность сетей



Границные маршрутизаторы сети

Границные маршрутизаторы сети позволяют организовать на периметре сети работу высокопроизводительных, безопасных и надежных сервисов для объединения кампусных сетей, сетей ЦОД и сетей филиалов.



Маршрутизаторы операторов связи

Маршрутизаторы поставщиков услуг, показанные на рисунке, предоставляют комплексные масштабируемые решения и услуги, поддерживаемые абонентами. Операторы должны оптимизировать свою работу, сократить расходы и повысить масштабируемость и гибкость, чтобы обеспечить возможность доступа к интернет-технологиям нового поколения для всех устройств, независимо от местоположения.



Промышленные решения

Промышленные маршрутизаторы, показанные на рисунке, предназначены для обеспечения функций корпоративного класса в жестких и суровых условиях. Их компактная, модульная, прочная конструкция отлично подходит для критически важных приложений



Формфакторы маршрутизатора

Как и коммутаторы, маршрутизаторы доступны в различных формфакторах. Сетевые администраторы в корпоративной среде должны уметь работать с различными типами маршрутизаторов — от небольших настольных моделей до стоечных устройств и блейд-моделей.

Cisco серии 900

Маршрутизаторы для офисов небольших филиалов. Он сочетает в себе WAN, коммутацию, безопасность и расширенные возможности подключения в компактной платформе для малого и среднего бизнеса.



ASR серии 9000 и 1000

Эти маршрутизаторы обеспечивают плотность и отказоустойчивость с возможностью программирования для масштабируемой периферии сети.



Серия 5500

Эти маршрутизаторы предназначены для эффективного масштабирования между крупными центрами обработки данных и крупными корпоративными сетями, веб-сетями и сетями поставщиков услуг WAN и агрегации.



Cisco 800

Этот маршрутизатор компактен и предназначен для суровых условий эксплуатации.



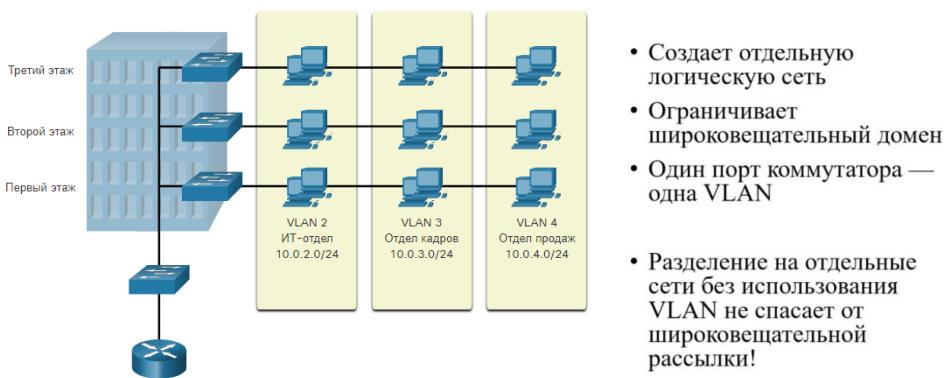
Кроме того, маршрутизаторы могут иметь фиксированную или модульную конфигурацию. Маршрутизаторы фиксированной конфигурации поставляются со встроенными интерфейсами. Модульные маршрутизаторы предлагаются с несколькими слотами, которые позволяют администратору менять интерфейсы маршрутизатора. Маршрутизаторы поставляются с разными интерфейсами, такими как Fast Ethernet, Gigabit Ethernet, последовательный и оптоволоконный.

10. VLAN:

a. Определение, типы VLAN.

В коммутируемых сетях сети VLAN обеспечивают адаптивность сегментации и организации. Группа устройств в сети VLAN взаимодействует так, будто устройства подключены с помощью одного кабеля. Сети VLAN основываются не на физических, а на логических подключениях.

Как показано на рисунке, VLAN в коммутируемой сети позволяют пользователям различных отделов (например, ИТ, HR и Sales) подключаться к одной и той же сети независимо от используемого физического коммутатора или местоположения в локальной сети кампуса.



Сети VLAN позволяют администратору производить сегментацию сети по функциям, проектным группам или областям применения, независимо от физического расположения пользователя или устройства.

Каждая VLAN считается отдельной логической сетью. Устройства в пределах VLAN работают таким образом, будто находятся в собственной независимой сети, даже если делят одну общую инфраструктуру с другими VLAN. Любой коммутационный порт может принадлежать сети VLAN. Пакеты одноадресной, широковещательной и многоадресной рассылки пересыпаются и рассыпаются только на оконечные устройства в пределах исходной сети VLAN этих пакетов. Пакеты, адресованные устройствам, которые не относятся к VLAN, должны пересыпаться через устройство, поддерживающее маршрутизацию.

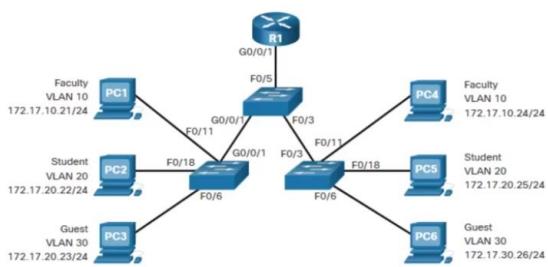
В коммутируемой сети может быть несколько подсетей IP без использования нескольких сетей VLAN. Однако устройства будут находиться в одном и том же домене широковещательной рассылки уровня 2. Это означает, что все широковещательные рассылки уровня 2, например ARP-запрос, будут приниматься всеми устройствами в коммутируемой сети, даже теми, которые не предназначены для приема данной рассылки.

VLAN создает логический домен широковещательной рассылки, который может охватывать несколько физических сегментов LAN. Разделяя крупные широковещательные домены на более мелкие сети, VLAN повышают производительность сети. Если устройство в одной VLAN передает широковещательный кадр Ethernet, то этот кадр получают все устройства в рамках этой VLAN, устройства в других сетях VLAN этот кадр не получают.

Сети VLAN позволяют реализовывать политику обеспечения доступа и безопасности, учитывая интересы различных групп пользователей. Каждый порт коммутатора может быть назначен только одной VLAN (за исключением порта, подключенного к IP-телефону или другому коммутатору).

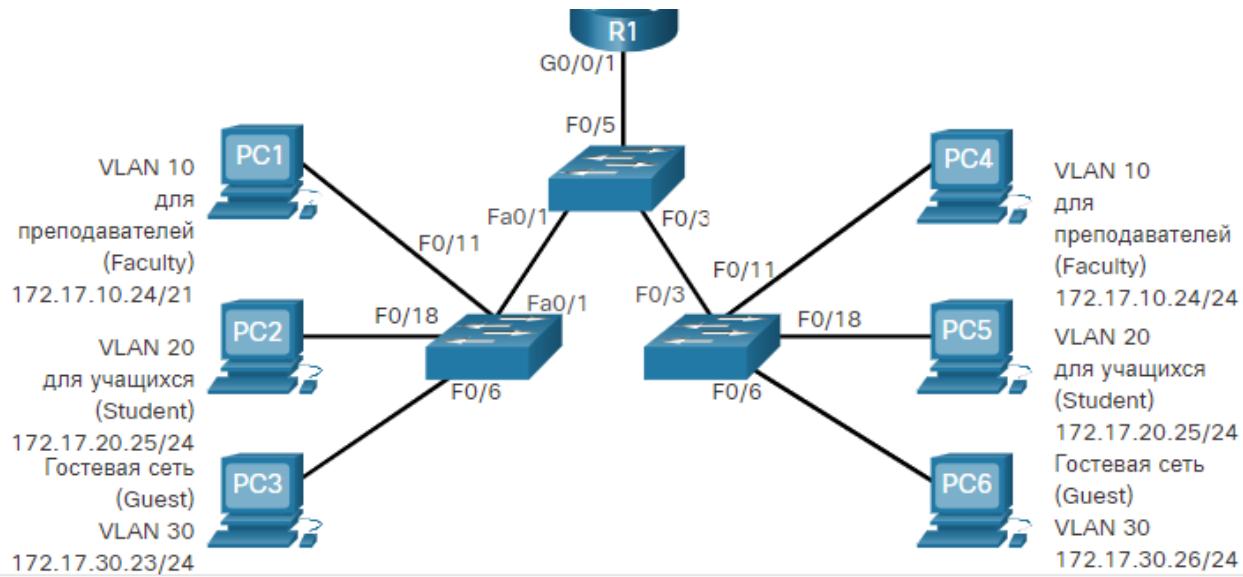
Преимущества виртуальных локальных сетей (VLAN)

Преимущества VLAN



- Безопасность
- Снижение расходов
- Улучшение производительности (за счёт уменьшения количества трафика)
- Уменьшение размера широковещательного домена
- Повышение производительности IT-сотрудников

Каждой VLAN в коммутируемой сети соответствует IP-сеть. Таким образом, в проекте VLAN необходимо учитывать использование иерархической схемы сетевой адресации. Иерархическая адресация подразумевает упорядоченное назначение номеров IP-сети сегментам или сетям VLAN с учетом работы сети в целом. Блоки смежных сетевых адресов резервируются и настраиваются на устройствах в определённой области сети.



Преимущество	Описание
Меньший размер широковещательных доменов	<ul style="list-style-type: none"> Разделение сети на VLAN уменьшает количество устройств в домене широковещательной рассылки. На рисунке есть шесть компьютеров в сети, но только три широковещательных домена (например, Faculty, Student и Guest).
Повышение безопасности	<ul style="list-style-type: none"> Только пользователи одной и той же сети VLAN могут общаться вместе. На рисунке, сетевой трафик факультета на VLAN 10 полностью отделен и защищен от пользователей в других VLAN. Улучшена эффективность ИТ-инфраструктуры
Повышение эффективности ИТ-инфраструктуры	<ul style="list-style-type: none"> VLAN упрощают управление сетью, поскольку пользователи с аналогичными требованиями к сети используют одну и ту же сеть VLAN. VLAN можно назначить соответствующее имя, чтобы упростить идентификацию. На рисунке VLAN 10 названа «Faculty», VLAN 20 «Student», и VLAN 30 «Guest».
Снижение затрат	VLAN сокращают потребность в дорогостоящем обновлении сети и используют существующую пропускную способность и восходящие каналы более эффективно, что приводит к сокращению затрат.
Повышение производительности	Меньшие широковещательные домены уменьшают ненужный трафик в сети, а также повышения производительности.
Упрощенная форма управления проектами и приложениями	<ul style="list-style-type: none"> VLAN объединяют пользователей и сетевые устройства для поддержки бизнес или географических требований. Наличие отдельных функций позволяет управлять проектом или работать с специализированным приложением; примером такого приложения является платформа развития электронного обучения для преподавателей.

Типы виртуальных локальных сетей

Типы VLAN

```
Switch# show vlan brief
VLAN Name      Status Ports
----- 
1   default     active  Fa0/1, Fa0/2, Fa0/3, Fa0/4
                  Fa0/5, Fa0/6, Fa0/7, Fa0/8
                  Fa0/9, Fa0/10, Fa0/11, Fa0/12
                  Fa0/13, Fa0/14, Fa0/15, Fa0/16
                  Fa0/17, Fa0/18, Fa0/19, Fa0/20
                  Fa0/21, Fa0/22, Fa0/23, Fa0/24
                  Gi0/1, Gi0/2
1002 fddi-default act/unsup
1003 token-ring-default act/unsup
1004 fddinet-default act/unsup
1005 trnet-default act/unsup
```

- Data VLAN
- Default VLAN (= VLAN 1)
- Native VLAN
- Management VLAN
- Voice VLAN (для VoIP)

VLAN используются для разных задач в современных сетях. Некоторые типы VLAN определяются классами трафика. Другие типы VLAN определяются конкретной функцией, которую они обслуживают.

Сеть VLAN по умолчанию

VLAN по умолчанию на коммутаторе Cisco — VLAN 1. Таким образом, все порты коммутатора находятся на VLAN 1, если они явно не настроены на другую VLAN. По умолчанию весь управляющий трафик уровня 2 связан с VLAN 1.

Важные факты, которые следует помнить о VLAN 1, включают следующее:

- По умолчанию все порты назначены сети VLAN 1.
- Сетью VLAN с нетегированым трафиком по умолчанию является сеть VLAN 1.
- Сетью управления VLAN по умолчанию является сеть VLAN 1.
- VLAN 1 нельзя переименовывать или удалять.

Например, в выходных данных команды **show vlan brief** все порты в настоящее время назначены во VLAN 1 по умолчанию. Ни одна сеть не назначена в качестве VLAN с нетегированным трафиком, и ни одна другая сеть VLAN не является активной. Таким образом, сеть VLAN с нетегированным трафиком будет управляющая сеть VLAN. Это считается угрозой безопасности

```
Switch# show vlan brief
VLAN Name Status Ports
----- 
1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4
                  Fa0/5, Fa0/6, Fa0/7, Fa0/8
                  Fa0/9, Fa0/10, Fa0/11, Fa0/12
                  Fa0/13, Fa0/14, Fa0/15, Fa0/16
                  Fa0/17, Fa0/18, Fa0/19, Fa0/20
                  Fa0/21, Fa0/22, Fa0/23, Fa0/24
                  Gi0/1, Gi0/2
1002 fddi-default act/unsup
1003 token-ring-default act/unsup
1004 fddinet-default act/unsup
1005 trnet-default act/unsup
```

Сеть VLAN для данных

VLAN данных — это VLAN, настроенные для разделения пользовательского трафика. Они называются пользовательскими VLAN, поскольку они разделяют сеть на группы пользователей или устройств. Современная сеть будет иметь множество VLAN данных в зависимости от требований организации. Обратите внимание, что голосовой и сетевой трафик управления не должен быть разрешен в VLAN для передачи данных.

VLAN с нетегированным трафиком (Native VLAN)

Пользовательский трафик из VLAN должен быть помечен идентификатором VLAN при его отправке на другой коммутатор. Транк порты используются между коммутаторами для поддержки передачи маркированного трафика. В частности, магистральный порт 802.1Q вставляет 4-байтовый тег в заголовок кадра Ethernet для идентификации VLAN, которой принадлежит кадр.

Коммутатору также может потребоваться отправить непомеченный трафик по магистральному каналу. Непомеченный трафик генерируется коммутатором и может также поступать с устаревших устройств. Транк порт 802.1Q размещает непомеченный трафик на собственной VLAN (Native VLAN). Собственная VLAN на коммутаторе Cisco — VLAN 1 (то есть VLAN по умолчанию).

Рекомендуется настроить собственную VLAN как неиспользуемую VLAN, отличную от VLAN 1 и других VLAN. Фактически, нет ничего необычного в том, чтобы выделять фиксированную VLAN для выполнения роли собственной VLAN для всех магистральных портов в коммутируемом домене

Управляющая VLAN (Management VLAN)

Управляющая VLAN - это VLAN для передачи данных, специально настроенная для трафика управления сетью, включая SSH, Telnet, HTTPS, NHTTP и SNMP. По умолчанию VLAN 1 настраивается как управляющая VLAN на коммутаторе уровня 2

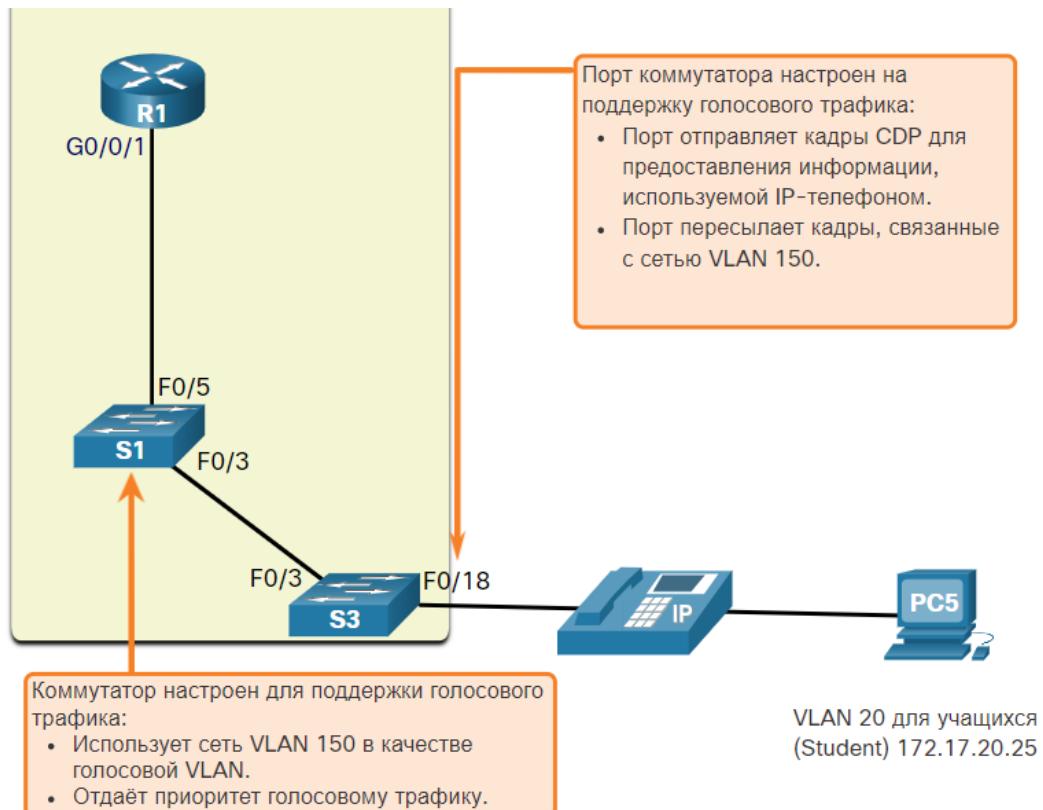
Голосовая VLAN

Для поддержки передачи голоса по IP (VoIP) требуется отдельная голосовая сеть VLAN. Для VoIP-трафика требуется следующее:

- Гарантированная пропускная способность для обеспечения высокого качества передачи голоса.
- Приоритет передачи перед другими типами сетевого трафика
- Возможность маршрутизации в обход перегруженных участков сети.
- Задержка менее 150 мс по всей сети.

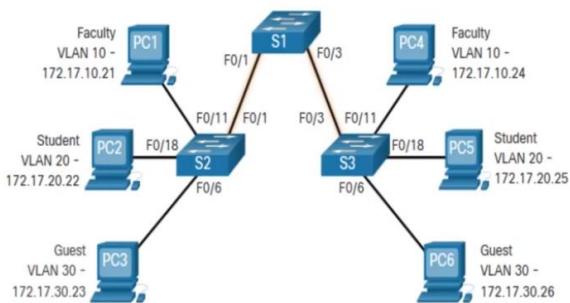
Чтобы удовлетворить эти требования, вся сеть должна быть спроектирована для поддержки VoIP.

На рисунке VLAN 150 предназначена для передачи голосового трафика. Студенческий компьютер PC5 подключен к IP-телефону Cisco, а телефон подключен к коммутатору S3. PC5 находится в VLAN 20, которая используется для данных студентов



b. Добавление тега. Trunk-порты.

Trunk Port



Функции транка Cisco:

- Разрешает несколько VLAN
- Расширяет сети VLAN по всей сети
- По умолчанию поддерживает все VLAN
- Поддерживает 802.1Q

VLAN не были бы очень полезны без транков VLAN. VLAN транки позволяют передавать весь трафик VLAN между коммутаторами. Это позволяет устройствам, подключенным к различным коммутаторам, но в одной и той же VLAN, взаимодействовать без прохождения через маршрутизатор.

Магистраль сетей VLAN (транк) — это двухточечный канал связи, который обслуживает более одной сети VLAN. Транк VLAN расширяет сети VLAN по всей сети. Cisco поддерживает IEEE 802.1Q для координации магистралей на интерфейсах Fast Ethernet, Gigabit Ethernet и 10-Gigabit Ethernet.

Транк VLAN не принадлежит определенной VLAN. Вместо этого он является каналом для нескольких VLAN между коммутаторами и маршрутизаторами. Кроме того, магистраль может использоваться между сетевым устройством и сервером или другим устройством, оснащенным соответствующим сетевым адаптером 802.1Q. По умолчанию на коммутаторе Cisco Catalyst все VLAN поддерживаются на магистральном порту.

На чертеже подвичены линии связи между коммутаторами S1 и S2 и S1 и S3 сконфигурированы для передачи трафика, поступающего из VLAN 10, 20, 30 и 99 (т.е. собственной VLAN) по сети. Данная сеть не сможет работать без магистралей VLAN.

Идентификация сети VLAN с помощью меток.

Стандартный заголовок фрейма Ethernet не содержит информации о VLAN, к которой принадлежит кадр. Поэтому при размещении кадров Ethernet в магистрали необходимо добавить информацию о VLAN, к которым они принадлежат. Этот процесс, называемый маркировка, выполняется с помощью заголовка IEEE 802.1Q, указанного в стандарте IEEE 802.1Q. Заголовок 802.1Q включает в себя 4-байтовый тег, вставленный в исходный заголовок фрейма Ethernet, определяющий VLAN, к которой принадлежит фрейм.

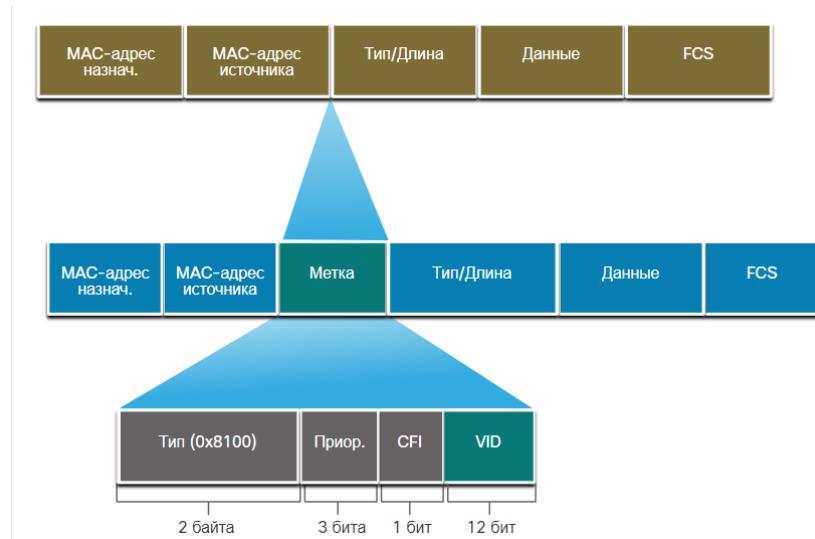
Когда коммутатор получает кадр на порте, настроенном в режиме доступа с назначенной сетью VLAN, он добавляет в заголовок кадра тег VLAN, заново вычисляет проверочную последовательность кадра (FCS) и отправляет этот тегированный кадр из магистрального порта.

Подробнее о поле тега VLAN

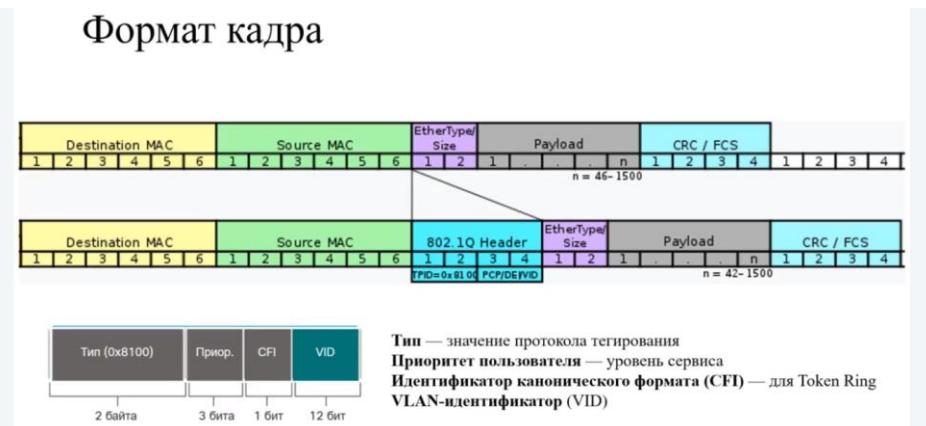
Как показано на рисунке, информационное поле управления тегами VLAN состоит из поля Type, поля Priority, поля Canonical Format Identifier и поля VLAN ID:

- **Тип** — это 2-байтовое значение, которое называется значением идентификатора протокола тегирования (TPID). Для Ethernet установлено шестнадцатеричное значение 0x8100.
- **User priority** - 3-битное значение, которое поддерживает реализацию уровня или сервиса.
- **Canonical Format Identifier (CFI)** - 1-битный идентификатор, который обеспечивает передачу кадров Token Ring по каналам Ethernet.
- **VLAN ID (VID)** — 12-разрядный идентификационный номер VLAN, поддерживающий до 4096 идентификаторов VLAN.

После того, как коммутатор вставляет поля информации управления тегами, он пересчитывает значения FCS и вставляет новую FCS в кадр.

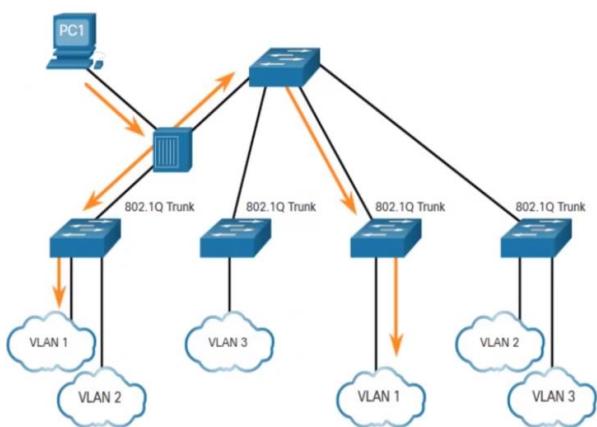


Формат кадра



Данные добавляет коммутатор. 12 бит для номера VLAN.

Native VLAN и тегирование



Основы магистрали 802.1Q:

- Маркировка обычно выполняется на всех VLAN
- Использование Native VLAN было разработано для устаревшего использования
- Если не изменено, VLAN 1 является Native VLAN
- Оба конца магистрального канала должны быть сконфигурированы с одной и той же Native VLAN
- Каждая магистраль настраивается отдельно, поэтому на отдельных магистралях можно иметь разные Native VLAN

VLAN с нетегированным трафиком и тегирование по протоколу 802.1Q

Стандарт IEEE 802.1Q определяет собственную VLAN для магистральных каналов, для которой по умолчанию используется VLAN 1. Когда кадр без тегов поступает на магистральный порт, он назначается собственной VLAN. Кадры управления, отправляемые между коммутаторами, являются примером трафика, который обычно не помечен. Если канал между двумя коммутаторами является магистральным, коммутатор отправляет непомеченный трафик на собственную VLAN.

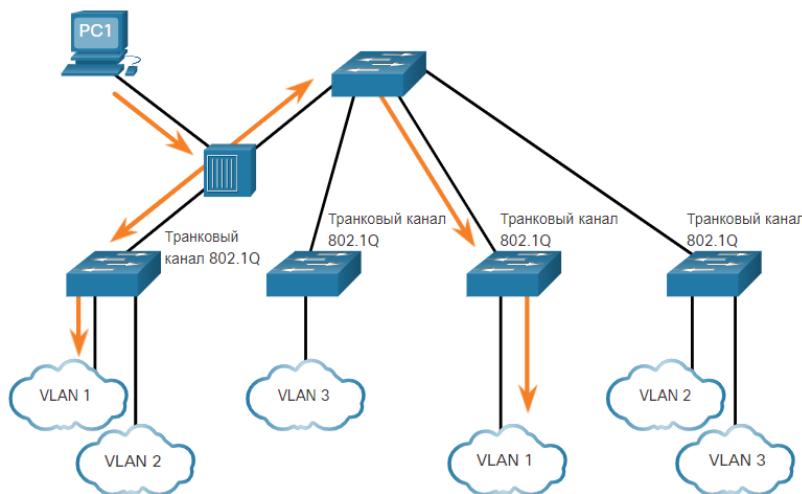
Тегированные кадры в сети native VLAN

Некоторые устройства, поддерживающие транкинг, добавляют тег VLAN в пакеты VLAN с нетегированным трафиком. Управляющий трафик, отправляемый в сети native VLAN, тегировать не следует. Если магистральный порт 802.1Q получает тегированный кадр с таким же идентификатором VLAN, как у сети VLAN с нетегированным трафиком, то он отбрасывает этот кадр. При настройке порта коммутатора Cisco настраивайте устройства таким образом, чтобы они не отправляли тегированные кадры по сети VLAN с нетегированным трафиком. Устройства других производителей, поддерживающие помеченные кадры в собственной VLAN, включают IP-телефоны, серверы, маршрутизаторы и коммутаторы, отличные от Cisco.

Нетегированные кадры в сети native VLAN

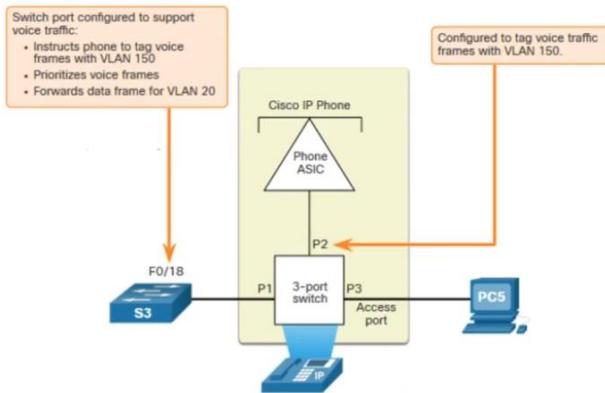
Когда магистральный порт коммутатора Cisco получает кадры без тегов (которые являются необычными для хорошо спроектированной сети), он пересыпает эти кадры в собственную VLAN. Если нет устройств, связанных с собственной VLAN (что не является необычным) и нет других магистральных портов (что не является необычным), то кадр отбрасывается. Сетью VLAN с нетегированным трафиком по умолчанию является сеть VLAN 1. При настройке магистрального порта 802.1Q значение собственного идентификатора VLAN ID (PVID) назначается по умолчанию. Весь непомеченный трафик, поступающий в порт 802.1Q или исходящий из него, перенаправляется на основе значения PVID. Например, если VLAN 99 настроена как собственная VLAN, PVID равно 99 и весь непомеченный трафик перенаправляется в VLAN 99. Если встроенная VLAN не была перенастроена, для параметра PVID устанавливается значение VLAN 1.

На рисунке PC1 подключен концентратором к магистральному каналу 802.1Q.



PC1 отправляет нетегированный трафик, который коммутаторы связывают с сетью VLAN с нетегированным трафиком, настроенной на магистральных портах, и пересыпают его соответствующим образом. Отбрасывается трафик с тегами на магистрали, получаемый PC1. Этот сценарий отражает плохое проектирование сети по нескольким причинам: он использует концентратор, у него есть хост, подключенный к магистральному каналу, и это означает, что коммутаторы имеют порты доступа, назначенные собственной VLAN. В этом сценарии также показано, что для поддержки устаревших сценариев необходима спецификация IEEE 802.1Q для VLAN с нетегированным трафиком.

Voice over IP



Для голосового трафика требуется:

- Гарантированная пропускная способность
- Высокий приоритет QoS
- Возможность избежать заторов
- Задержка менее 150 мс от источника к месту назначения
- Вся сеть должна быть спроектирована для поддержки голосовой связи

Для поддержки передачи голоса по IP (VoIP) требуется отдельная голосовая сеть VLAN. Это позволяет применять к голосовому трафику политики качества обслуживания (QoS) и безопасности.

Каждый IP-телефон необходимо подключать напрямую к порту коммутатора. IP-хост может подключиться к IP-телефону, чтобы получить сетевое соединение. Для порта доступа, к которому подключен IP-телефон Cisco, можно настроить использование двух отдельных сетей VLAN: Одна VLAN предназначена для голосового трафика, а другая — для передачи данных VLAN для поддержки трафика хоста. Связь между коммутатором и IP-телефоном имитирует магистральную связь для передачи голосового трафика VLAN и трафика VLAN данных.

IP-телефон содержит встроенный 3-портовый коммутатор 10/100. Порты обеспечивают выделенные подключения к следующим устройствам:

- Порт 1 подключается к коммутатору или другому устройству VoIP.
- Порт 2 — это внутренний интерфейс 10/100, через который передается трафик IP-телефона;
- Порт 3 (порт доступа) подключается к ПК или другому устройству.

Порт доступа коммутатора отправляет пакеты CDP, инструктируя подключенный IP-телефон отправлять голосовой трафик одним из трех способов. Используемый метод варьируется в зависимости от типа трафика:

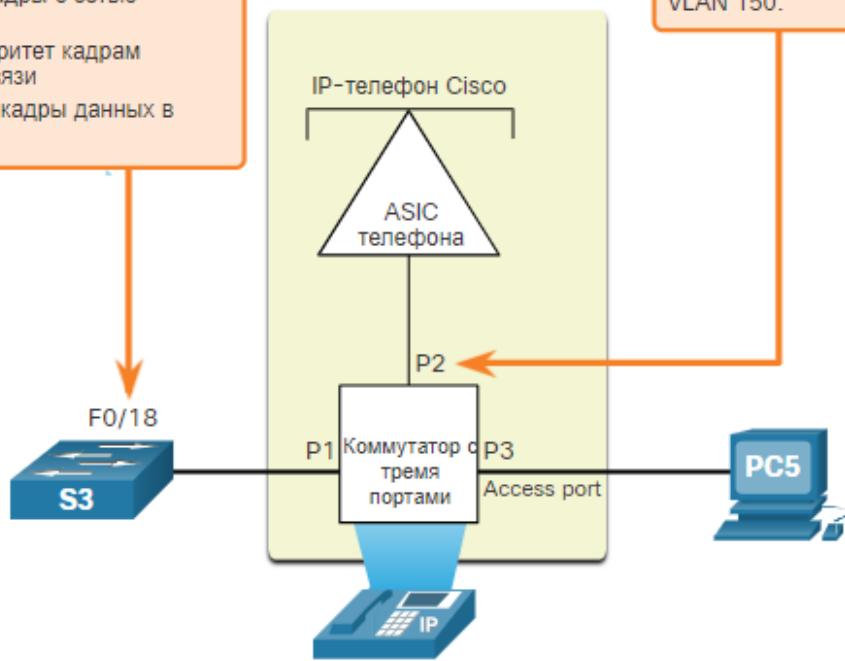
- в голосовой VLAN, тегированной значением приоритета класса обслуживания (CoS) уровня 2;
- в сети VLAN доступа, тегированной значением приоритета CoS уровня 2;
- в нетегированной VLAN доступа (без значения приоритета CoS уровня 2).

На рисунке студенческий компьютер PC5 подключен к IP-телефону Cisco, а телефон подключен к коммутатору S3. VLAN 150 предназначен для передачи голосового трафика, в то время как PC5 находится в VLAN 20, которая используется для данных студентов.

Коммутационный порт настроен для поддержки голосового трафика.

- Указывает телефону тегировать голосовые кадры с сетью VLAN 150.
- Отдает приоритет кадрам голосовой связи
- Пересыпает кадры данных в VLAN 20

Настроен для тегирования голосовых кадров с сетью VLAN 150.



Voice over IP

```
S1# sh interfaces fa0/18 switchport
Name: Fa0/18
Switchport: Enabled
Administrative Mode: static access
Operational Mode: down
Administrative Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 20 (student)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 150 (voice)
<output omitted>
```

с. Диапазоны VLAN: стандартные и расширенные.

Диапазоны VLAN

Switch# show vlan brief			
VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24 Gi0/1, Gi0/2
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fdtnet-default	act/unsup	
1005	trnet-default	act/unsup	

VLAN стандартного диапазона:

- Малые и средние сети
- От 1 до 1005
- От 1002 до 1005 зарезервированы
- Конфигурация хранится в vlan.dat

VLAN расширенного диапазона:

- Для больших сетей
- От 1006 до 4094
- Конфигурация хранится в файле текущей конфигурации (не vlan.dat)

Сети VLAN стандартного диапазона

Ниже приведены характеристики VLAN нормального диапазона:

- Используются в малых и средних сетях предприятий и организаций.
- Сети VLAN обычного диапазона определяются идентификатором VLAN от 1 до 1005.
- (Идентификаторы от 1002 до 1005 резервируются для сетей VLAN типа Token Ring и FDDI.)
- Идентификаторы 1 и идентификаторы от 1002 до 1005 создаются автоматически и не могут быть удалены.
- Конфигурации хранятся в файле базы данных VLAN (vlan.dat), который находится во флеш-памяти.
- При настройке протокол магистрального соединения VLAN (VTP) помогает синхронизировать базу данных VLAN между коммутаторами.

Сети VLAN расширенного диапазона

Ниже приведены характеристики сетей VLAN расширенного диапазона:

- Они используются поставщиками услуг для обслуживания нескольких клиентов и глобальными предприятиями, достаточно крупными для того, чтобы нуждаться в идентификаторах VLAN расширенного диапазона.
- Определяются идентификатором VLAN от 1006 до 4094.
- По умолчанию они сохраняются в файле текущей конфигурации.
- Поддерживают меньше функций VLAN, чем сети VLAN стандартного диапазона.
- Требуется конфигурация прозрачного режима VTP для поддержки сетей VLAN расширенного диапазона.

Примечание: 4096 является верхней границей для числа VLAN, доступных на коммутаторах Catalyst, поскольку в поле идентификатора VLAN заголовка IEEE 802.1Q имеется 12 бит.

d. Управление портами в VLAN: добавление, удаление, изменение принадлежности.

Команды создания VLAN

Настройка VLAN

Создание VLAN:

```
S(config)# vlan vlan_id
S(config-vlan)# name vlan_name
S(config-vlan)# exit
```

Проверка:

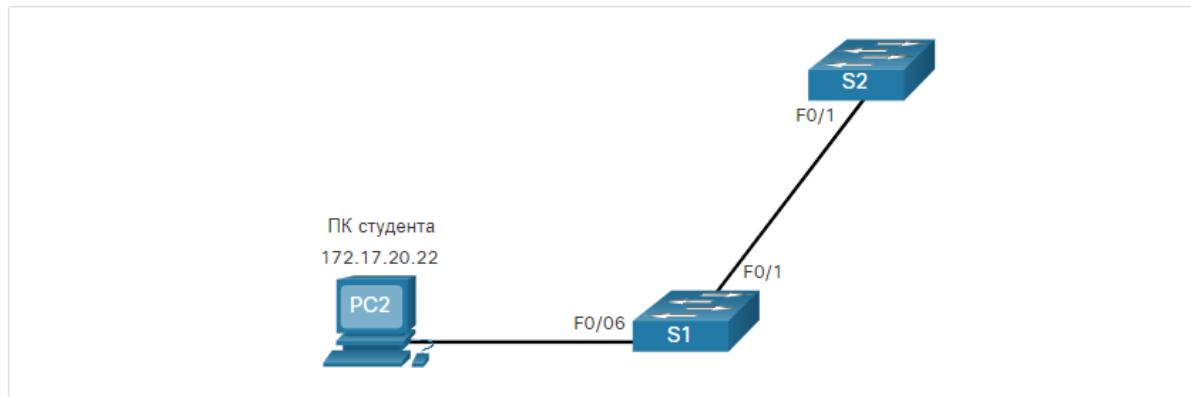
```
S# show vlan brief
```

При настройке сетей VLAN стандартного диапазона сведения о конфигурации хранятся во флеш-памяти коммутатора, в файле *vlan.dat*. Флэш-память является постоянной и не требует команды **copy running-config startup-config**. Однако, поскольку одновременно с созданием сетей VLAN на коммутаторах Cisco часто также настраиваются и другие параметры, рекомендуется сохранять изменения текущей конфигурации в файл загрузочной конфигурации.

В таблице показан синтаксис команды Cisco IOS, используемый для добавления VLAN к коммутатору и указания ему имени. При настройке коммутатора рекомендуется присваивать имя каждой сети VLAN.

Задача	Команда IOS
Войдите в режим глобальной настройки.	Switch# configure terminal
Создайте сеть VLAN с допустимым номером идентификатора.	Switch(config)# vlan <i>vlan-id</i>
Укажите уникальное имя для идентификации сети VLAN.	Switch(config-vlan)# name <i>vlan-name</i>
Вернитесь в привилегированный режим.	Switch(config-vlan)# end

В топологии студенческий компьютер (PC2) еще не связан с VLAN, но имеет IP-адрес 172.17.20.22, который принадлежит VLAN 20.



В примере показано, как настраивается студенческая VLAN (VLAN 20) на коммутаторе S1.

```
S1# configure terminal  
S1(config)# vlan 20  
S1(config-vlan)# name student  
S1(config-vlan)# end
```

Примечание: Помимо одного идентификатора VLAN, можно ввести группу идентификаторов VLAN, разделенных запятыми, или диапазон идентификаторов VLAN, разделенных дефисами, с помощью команды **vlan *vlan-id***. Например, ввод команды глобальной конфигурации **vlan 100,102,105-107** приведет к созданию сетей VLAN 100, 102, 105, 106 и 107.

Добавление портов в VLAN

```
S(config)# interface interface_id  
S(config-if)# switchport mode access  
S(config-if)# switchport access vlan vlan_id
```

Для голосового трафика:

```
S(config-if)# switchport voice vlan vlan_id
```

Следующий шаг после создания сети VLAN — назначение портов сетям VLAN.

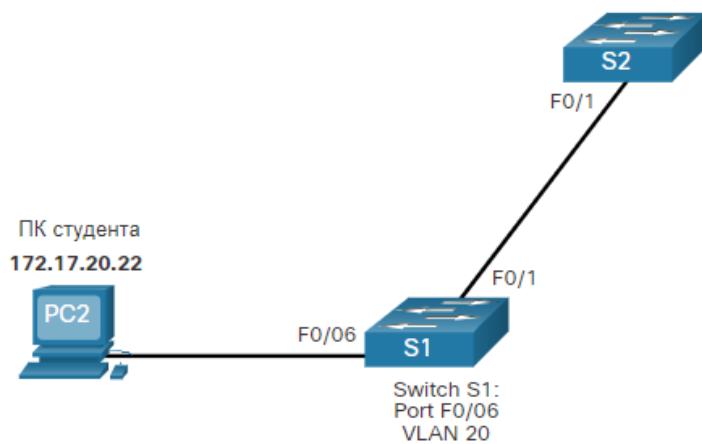
В таблице показан синтаксис для определения порта в качестве порта доступа и назначения его сети VLAN. Выполнять команду **switchport mode access** необязательно, но настоятельно рекомендуется в целях обеспечения безопасности. С помощью этой команды интерфейс переходит в режим постоянного доступа.

Задача	Команда IOS
Войдите в режим глобальной настройки.	Switch# configure terminal

Задача	Команда IOS
Войдите в режим интерфейсной конфигурации.	Switch(config)# interface interface-id
Переведите порт в режим доступа.	Switch(config-if)# switchport mode access
Назначьте порт сети VLAN.	Switch(config-if)# switchport access vlan vlan-id
Вернитесь в привилегированный исполнительский режим.	Switch(config-if)# end

Примечание: Для одновременной настройки нескольких интерфейсов используйте команду **interface range**.

На рисунке порт F0/6 коммутатора S1 настроен в качестве порта доступа и назначается VLAN 20. Любое устройство, подключенное к этому порту, связывается с VLAN 20. Таким образом, в нашем примере компьютер PC2 находится в сети VLAN 20.



В примере показана конфигурация S1 для назначения F0/6 VLAN 20.

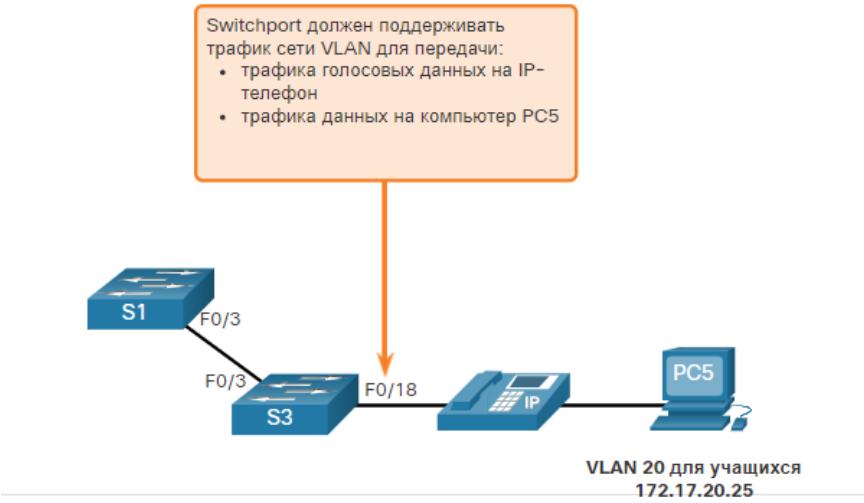
```
S1# configure terminal
S1(config)# interface fa0/6
S1(config-if)# switchport mode access
S1(config-if)# switchport access vlan 20
S1(config-if)# end
```

Сети VLAN настраиваются на коммутационном порте, а не на устройстве. Для компьютера PC2 адрес IPv4 и маска подсети связаны с сетью VLAN, настроенной на коммутационном порте. В данном примере это сеть VLAN 20. При настройке VLAN 20 на других коммутаторах сетевой администратор должен настроить другие компьютеры студентов в той же подсети, в которой находится компьютер PC2 (172.17.20.0/24).

VLAN для передачи данных и голоса

Порт доступа может принадлежать только одной VLAN. Однако, порт также может быть связан с голосовой VLAN. Например, порт, подключенный к IP-телефону и конечному устройству, будет связан с двумя VLAN: одной для голосовой связи и другой для данных.

Рассмотрим топологию на рисунке. В этом примере компьютер PC5 подключен к IP-телефону Cisco, который в свою очередь подключен к интерфейсу FastEthernet 0/18 на коммутаторе S3. Для реализации этой конфигурации создаются сети VLAN 20 и голосовая VLAN 150.



Используйте команду конфигурации интерфейса **switchport voice vlan *vlan-id***, чтобы назначить голосовую VLAN порту

Локальные сети, поддерживающие трафик голосовых данных, обычно также имеют включенную гарантированную полосу пропускания (QoS). Трафик голосовых данных должен быть помечен как доверенный сразу при входе в сеть. Используйте команду настройки интерфейса **mls qos trust [cos | device cisco-phone | dscp | ip-precedence]**, чтобы установить доверенное состояние интерфейса и указать, какие поля пакета используются для классификации трафика.

Конфигурация на рис. 4 создает две сети VLAN (VLAN 20 и VLAN 150), а затем назначает интерфейс F0/18 коммутатора S3 в качестве коммутационного порта в сети VLAN 20. Она также назначает трафик голосовых данных в сеть VLAN 150 и включает классификацию QoS на основе класса обслуживания (CoS), назначенного IP-телефоном.

Команда **switchport access vlan** принудительно создаёт VLAN, если таковая ещё не существует на коммутаторе. Например, сеть VLAN 30 отсутствует в данных, выводимых командой **show vlan brief** на коммутаторе. Если команда **switchport access vlan 30** вводится на любом интерфейсе без предыдущей конфигурации, коммутатор отображает следующее:

```
% Access VLAN does not exist. Creating vlan 30
```

Проверка информации о сетях VLAN

После настройки сети VLAN ее конфигурации можно проверить с помощью команд Cisco IOS категории «**show**».

Команда **show vlan** отображает список всех настроенных сетей VLAN. Команда **show vlan** также может быть использована с параметрами. Полный синтаксис: **show vlan [brief | id *vlan-id* | name *vlan-name* | summary]**.

В таблице описаны параметры команд **show vlan**.

Задача	Вариант команды
Отображает имя, состояние и порты VLAN по одной VLAN на строку.	brief
Отображает информацию об отдельной VLAN, определяемой по номеру идентификатора VLAN. Для <i>vlan-id</i> , диапазон идентификаторов VLAN: от 1 до 4094.	id <i>vlan-id</i>

Задача	Вариант команды
Отображает информацию об имени одной сети VLAN. <i>Имя VLAN</i> — это код ASCII размером от 1 до 32 символов.	name vlan-name
Отображает общую информацию о VLAN.	summary

Команда **show vlan summary** отображает список всех настроенных сетей VLAN.

```
S1# show vlan summary
Number of existing VLANs : 7
Number of existing VTP VLANs : 7
Number of existing extended VLANS : 0
```

Другими полезными командами являются команды **show interfaces interface-id switchport** и **show interfaces vlan vlan-id**. Например, команда **show interfaces fa0/18 switchport** может быть использована для подтверждения того, что порт FastEthernet 0/18 правильно назначен для VLAN данных и голосовых сетей.

```
S1# show interfaces fa0/18 switchport
Name: Fa0/18
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 20 (student)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: 150
Administrative private-vlan host-association: none
(Output omitted)
```

show vlan name

```
S1# show vlan name student
VLAN Name          Status    Ports
----- -----
20  student         active   Fa0/11, Fa0/18

VLAN Type SAID MTU Parent RingNo BridgeNo Stp BrdgMode Trans1 Trans2
----- -----
20  enet 100020 1500 -      -      -      -      0      0

Remote SPAN VLAN
-----
Disabled

Primary Secondary Type          Ports
----- -----
```

Изменение принадлежности порта сети VLAN

Существует несколько способов изменить членство во VLAN.

Если порт доступа коммутатора неправильно назначен VLAN, просто повторно введите команду конфигурации интерфейса **switchport access vlan vlan-id** с правильным идентификатором VLAN. Например, предположим, что Fa0/18 неправильно настроено на VLAN 1 по умолчанию вместо VLAN 20. Чтобы изменить порт на VLAN 20, просто введите **switchport access vlan 20**.

Чтобы изменить членство порта на VLAN 1 по умолчанию, используйте команду **no switchport access vlan** в режиме настройки интерфейса, как показано на иллюстрации.

Например, в выходных данных Fa0/18 настроено на VLAN 1 по умолчанию, что подтверждается командой **show vlan brief**.

```
S1(config)# interface fa0/18
S1(config-if)# no switchport access vlan
S1(config-if)# end
S1#
S1# show vlan brief
VLAN Name          Status    Ports
----- 
1     default       active    Fa0/1, Fa0/2, Fa0/3, Fa0/
                           Fa0/5, Fa0/6, Fa0/7, Fa0/
                           Fa0/9, Fa0/10, Fa0/11, Fa
                           Fa0/13, Fa0/14, Fa0/15, F
                           Fa0/17, Fa0/18, Fa0/19, F
                           Fa0/21, Fa0/22, Fa0/23, F
                           Gi0/1, Gi0/2
20    student        active
1002 fddi-default   act/unsup
1003 token-ring-default act/unsup
```

Обратите внимание, что VLAN 20 по-прежнему активна, несмотря на то, что ей не назначены порты.

Выходные данные **show interfaces f0/18 switchport** также можно использовать для проверки того, что VLAN доступа для интерфейса F0/18 была сброшена на VLAN 1, как показано на выходных данных.

```
S1# show interfaces fa0/18 switchport
Name: Fa0/18
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
```

Удаление VLAN

Удаление настроек с интерфейса:

```
S(config)# interface interface_id
S(config-if)# no switchport mode access
```

Удаление сети VLAN:

```
S(config)# no vlan vlan_id
```

ИЛИ

```
S# delete flash:vlan.dat
```

Команда **no vlan *vlan-id*** режима глобальной конфигурации используется для удаления VLAN из файла коммутатора *vlan.dat*.

Внимание: Перед удалением сети VLAN необходимо сначала переназначить все ее порты другой сети VLAN. Все порты, которые не будут перемещены в активную VLAN, не смогут взаимодействовать с другими станциями после удаления VLAN.

Можно удалить весь файл *vlan.dat* с помощью команды **delete flash:vlan.dat** в привилегированном исполнительском режиме. Сокращенную версию команды (**delete vlan.dat**) можно использовать только в том случае, если файл *vlan.dat* находится в своем расположении по умолчанию. После выполнения этой команды и перезагрузки коммутатора ранее настроенные сети VLAN будут недоступны. Фактически, это позволяет восстановить на коммутаторе его заводские настройки VLAN.

Примечание: Чтобы восстановить заводское состояние коммутатора Catalyst, отключите от коммутатора все кабели, кроме консоли и кабеля питания. Затем введите привилегированную команду режима EXEC **erase startup-config**, за которой следует команда **delete vlan.dat**.

Настройка Trunk порта

```
S(config)# interface interface_id
S(config-if)# switchport mode trunk
S(config-if)# switchport trunk encapsulation dot1q
S(config-if)# switchport trunk native vlan vlan_id
S(config-if)# switchport trunk allowed vlan vlan_list
```

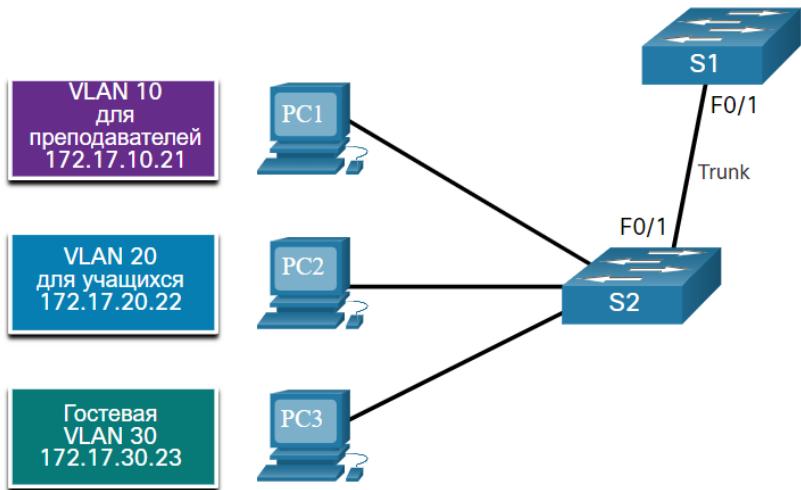
Теперь, когда вы настроили и проверили VLAN, пришло время настроить и проверить транки VLAN. Транк VLAN — это канал уровня 2 между двумя коммутаторами, несущий трафик для всех VLAN (если разрешенный список VLAN не ограничен вручную или динамически).

Чтобы включить магистральные каналы, настройте соединительные порты с помощью набора команд конфигурации интерфейса, показанных в таблице.

Задача	Команда IOS
Войдите в режим глобальной настройки.	Switch# configure terminal
Войдите в режим интерфейсной конфигурации.	Switch(config)# interface <i>interface-id</i>
Установите порт в режим постоянного транка.	Switch(config-if)# switchport mode trunk
Установите в качестве VLAN c нетегированным трафиком сеть, отличную от VLAN 1.	Switch(config-if)# switchport trunk native vlan <i>vlan-id</i>
Укажите список сетей VLAN, которым разрешен доступ в магистральный канал.	Switch(config-if)# switchport trunk allowed vlan <i>vlan-list</i>
Вернитесь в привилегированный исполнительский режим.	Switch(config-if)# end

Пример конфигурации магистрального канала

На риунке сети VLAN 10, 20 и 30 поддерживают компьютеры для инструкторов, учащихся и гостевой компьютер (PC1, PC2 и PC3). Порт F0/1 на коммутаторе S1 настроен в качестве транкового порта и пересыпает трафик для сетей VLAN 10, 20 и 30. Сеть VLAN 99 настроена в качестве native VLAN.



К каждой VLAN относятся следующие подсети:

- VLAN 10 - Faculty/Staff - 172.17.10.0/24
- VLAN 20 - Students - 172.17.20.0/24
- VLAN 30 - Guests - 172.17.30.0/24
- VLAN 99 - Native - 172.17.99.0/24

В примере показана конфигурация порта F0/1 коммутатора S1 в качестве магистрального порта. Сеть VLAN с нетегированым трафиком изменена на VLAN 99, а список разрешенных сетей VLAN включает только сети 10, 20, 30 и 99.

```
S1(config)# interface fastEthernet 0/1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk native vlan 99
S1(config-if)# switchport trunk allowed vlan 10,20,30,99
S1(config-if)# end
```

Примечание: Эта конфигурация предполагает использование коммутаторов Cisco Catalyst 2960, которые автоматически используют инкапсуляцию 802.1Q на магистральных каналах. Для других коммутаторов может потребоваться ручная настройка инкапсуляции. Всегда настраивайте оба конца магистрального канала с одной и той же native VLAN. Если конфигурация магистрали 802.1Q не одинакова на обоих концах, ПО Cisco IOS сообщает об ошибках.

Проверка

```
S# show interfaces interface_id switchport
S# show interfaces trunk
```

Выходной сигнал коммутатора отображает конфигурацию порта коммутатора F0/1 коммутатора S1. Конфигурация проверяется с помощью команды **show interfaces interface-ID switchport**.

```
S1# show interfaces fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (VLAN0099)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: 10,20,30,99
Pruning VLANs Enabled: 2-1001
(output omitted)
```

Верхняя подсвеченная область показывает, что порт F0/1 имеет административный режим **trunk**. Порт находится в режиме транкинга. Следующая подсвеченная область проверяет, является ли собственная VLAN 99. Далее вниз по выходным данным, нижняя подсвеченная область показывает, что VLAN 10, 20, 30 и 99 включены на магистрали.

show interfaces trunk

```
S1# show interfaces trunk
Port Mode Encapsulation Status Native vlan
Fa0/1 on 802.1q trunking 99
Port Vlans allowed on trunk
Fa0/1 10,20,30,99
Port Vlans allowed and active in management domain
Fa0/1 10,20,30,99
Port Vlans in spanning tree forwarding state and not pruned
Fa0/1 10,20,30,99
```

Сброс транка в состояние по умолчанию

Используйте команды **no switchport trunk allowed vlan** и **no switchport trunk native vlan** для удаления разрешенных VLAN и сброса собственной VLAN магистрали. При восстановлении состояния по умолчанию магистраль разрешает все VLAN и использует VLAN 1 в качестве собственной VLAN. В примере показаны команды, используемые для сброса всех характеристик транкинга интерфейса транкинга к настройкам по умолчанию.

```
S1(config)# interface fa0/1
S1(config-if)# no switchport trunk allowed vlan
S1(config-if)# no switchport trunk native vlan
S1(config-if)# end
```

The **show interfaces f0/1 switchport** command reveals that the trunk has been reconfigured to a default state.

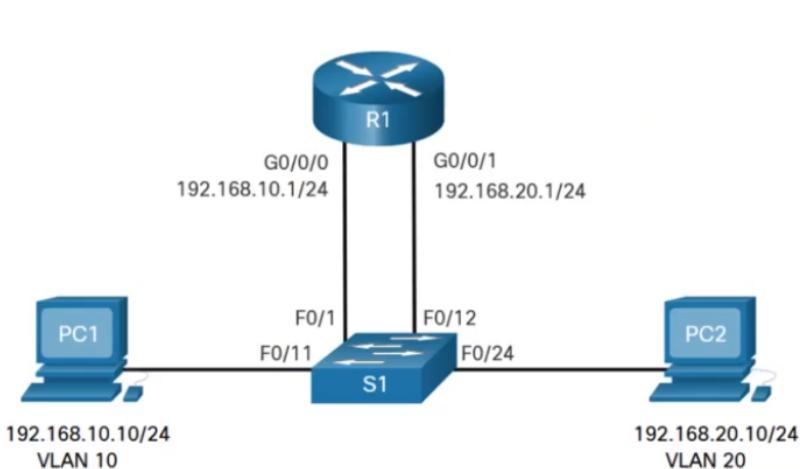
```
S1# show interfaces fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
(output omitted)
```

В этом примере выходных данных показаны команды, используемые для удаления функции магистрали из порта коммутатора F0/1 на коммутаторе S1. Команда **show interfaces f0/1 switchport** показывает, что интерфейс F0/1 теперь находится в статическом режиме доступа.

```
S1(config)# interface fa0/1
S1(config-if)# switchport mode access
S1(config-if)# end
S1# show interfaces fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
(output omitted)
```

e. Маршрутизация между VLAN: способы, описание.

Маршрутизация между VLAN



Три подхода:

- **Старый: 1 порт – 1 VLAN**
- Router-on-a-Stick
- Коммутация 3-его уровня с использованием интерфейса SVI

VLAN используются для сегментации коммутируемых сетей уровня 2 по разным причинам. Независимо от причины, хосты в одной VLAN не могут взаимодействовать с хостами в другой VLAN, если нет маршрутизатора или коммутатора уровня 3 для предоставления услуг маршрутизации.

Маршрутизация между сетями VLAN — это процесс переадресации сетевого трафика из одной сети VLAN в другую с помощью маршрутизатора.

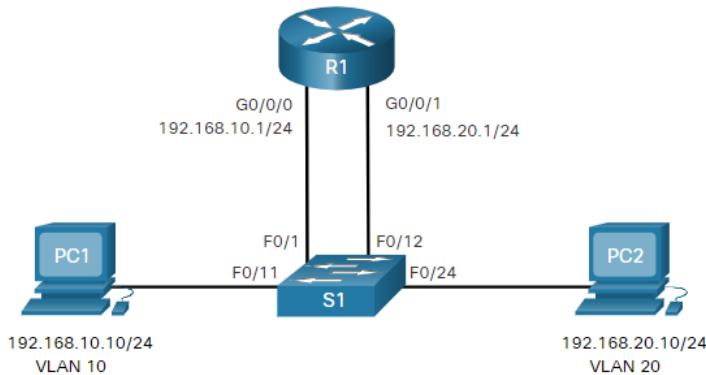
Существуют три варианта маршрутизации между VLAN.

- **Старый метод маршрутизации между VLAN** - это устаревшее решение. Плохо масштабируется
- **Router-on-a-Stick** - это приемлемое решение для сети малых и средних размеров.
- **Коммутатор уровня 3 с использованием коммутируемых виртуальных интерфейсов (SVI)** — это наиболее масштабируемое решение для средних и крупных организаций.

Устаревшие методы маршрутизации между сетями VLAN

Первое решение маршрутизации между VLAN основывалось на использовании маршрутизатора с несколькими интерфейсами Ethernet. Каждый интерфейс маршрутизатора был подключен к порту коммутатора в разных VLAN. Интерфейсы маршрутизатора служат шлюзами по умолчанию для локальных узлов в подсети VLAN.

Например, обратитесь к топологии, где R1 имеет два интерфейса, подключенных к коммутатору S1.



Примечание в примере таблицы MAC-адресов S1 заполняется следующим образом:

- Порт Fa0/1 назначен VLAN 10 и подключен к интерфейсу R1 G0/0/0.
- Порт Fa0/11 назначен VLAN 10 и подключен к PC1.
- Порт Fa0/12 назначен VLAN 20 и подключен к интерфейсу R1 G0/0/1.
- Порт Fa0/11 назначается VLAN 20 и подключен к PC2.

Когда PC1 отправляет пакет PC2 в другой сети, он пересыпает его на шлюз по умолчанию 192.168.10.1. R1 получает пакет через интерфейс G0/0/0 и проверяет адрес назначения пакета. Затем R1 направляет пакет из интерфейса G0/0/1 на порт F0/12 в VLAN 20 на S1. Наконец, S1 перенаправляет кадр на PC2.

Устаревший метод маршрутизации между VLAN, использующий физические интерфейсы, имеет большие ограничения. Он не является достаточно масштабируемым, поскольку маршрутизаторы имеют ограниченное количество физических интерфейсов. По мере возрастания количества VLAN в сети, требующих по одному физическому интерфейсу на каждую VLAN, количество свободных интерфейсов маршрутизатора быстро уменьшается.

В нашем примере R1 требуется два отдельных интерфейса Ethernet для маршрутизации между VLAN 10 и VLAN 20. Что делать, если было шесть (или более) VLAN для связи? Для каждой VLAN потребуется отдельный интерфейс. Очевидно, что это решение не масштабируется.

Примечание: Этот метод маршрутизации между VLAN больше не реализован в коммутируемых сетях и включен только для пояснений.

Настройка по устаревшему методу

Настройка коммутатора:

```
S(config)# vlan vlan_id
S(config-vlan)# name vlan_name

S(config)# interface interface_id
S(config-if)# switchport mode access
S(config-if)# switchport access vlan vlan_id
```

Настройка маршрутизатора:

```
R(config)# interface interface_id
R(config-if)# ip address ip_address subnet_mask
R(config-if)# no shutdown
```

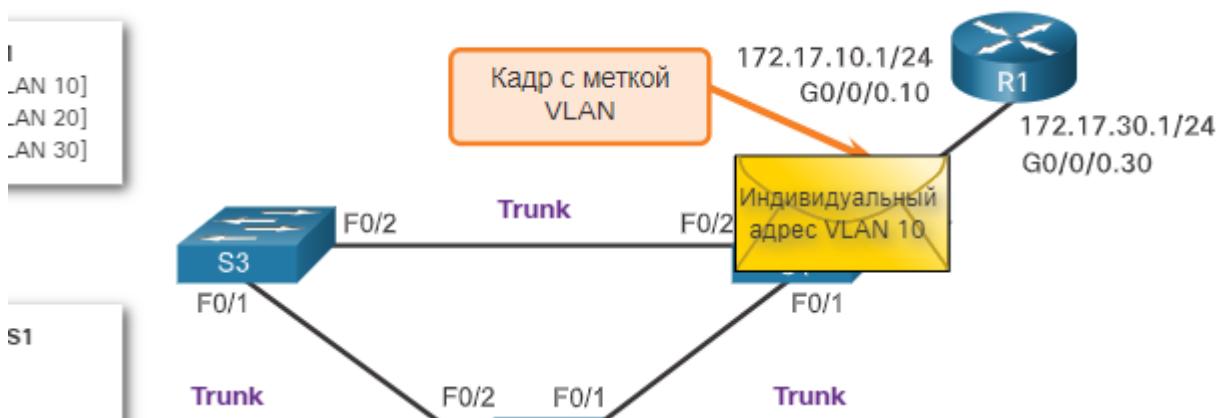
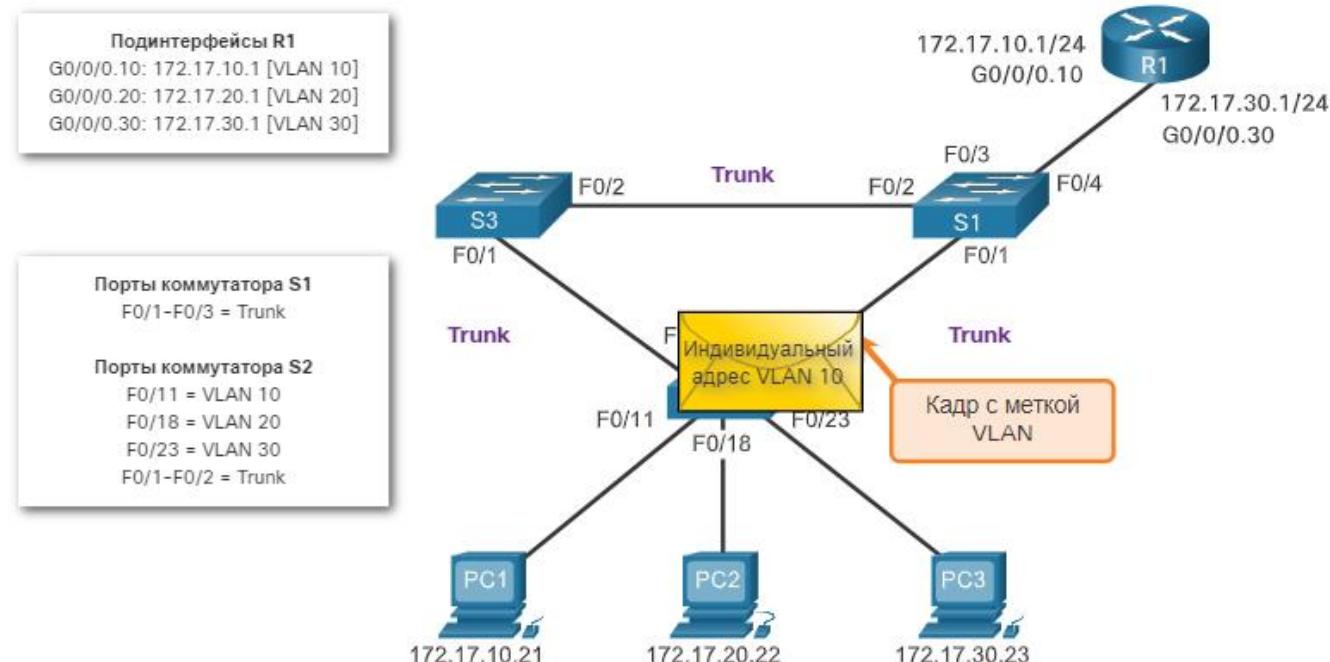
Маршрутизация между сетями VLAN с использованием метода Router-on-a-Stick

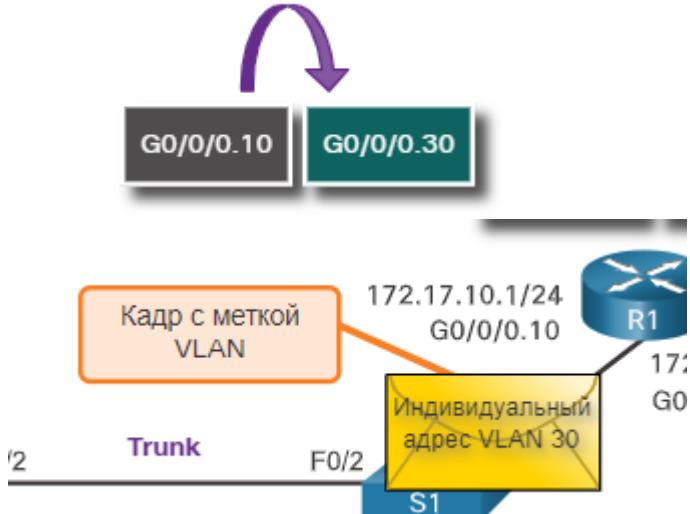
Метод маршрутизации между VLAN «router-on-a-stick» преодолевает ограничение устаревшего метода маршрутизации между VLAN. Для маршрутизации трафика между несколькими сетями VLAN в сети требуется только один физический интерфейс Ethernet.

Интерфейс Ethernet маршрутизатора Cisco IOS настроен как магистральное соединение 802.1Q и подключен к магистральному порту коммутатора уровня 2. В частности, интерфейс маршрутизатора настраивается с использованием подинтерфейсов для идентификации маршрутизуемых VLAN.

Настроенные подинтерфейсы являются программными виртуальными интерфейсами. Каждый из них связан с одним физическим интерфейсом Ethernet. Подинтерфейсы настраиваются в программном обеспечении на маршрутизаторе. Каждому подинтерфейсу отдельно назначаются IP-адрес и длина префикса. Подинтерфейсы настроены для разных подсетей, которые соответствуют назначенным им VLAN. Это облегчает логическую маршрутизацию.

Когда трафик с тегом VLAN входит в интерфейс маршрутизатора, он перенаправляется на подинтерфейс VLAN. После принятия решения о маршрутизации на основе IP-адреса назначения маршрутизатор определяет интерфейс выхода для трафика. Если выходной интерфейс настроен как подинтерфейс 802.1q, Кадры данных помечены новой меткой VLAN отправляются обратно на физический интерфейс.





Как видно из анимации, PC1 на VLAN 10 взаимодействует с PC3 на VLAN 30. Маршрутизатор R1 принимает помеченный одноадресный трафик в сети VLAN 10 и направляет его в сеть VLAN 30 с помощью своих настроенных подинтерфейсов. Коммутатор S2 удаляет из одноадресного кадра метку сети VLAN и пересыпает кадр на порт F0/23 компьютера PC3.

Примечание: Примечание. Маршрутизация между VLAN с использованием метода router-on-a-stick не масштабируется при работе более 50 сетей VLAN.

Настройка по методу router-on-a-stick

Настройка коммутатора:

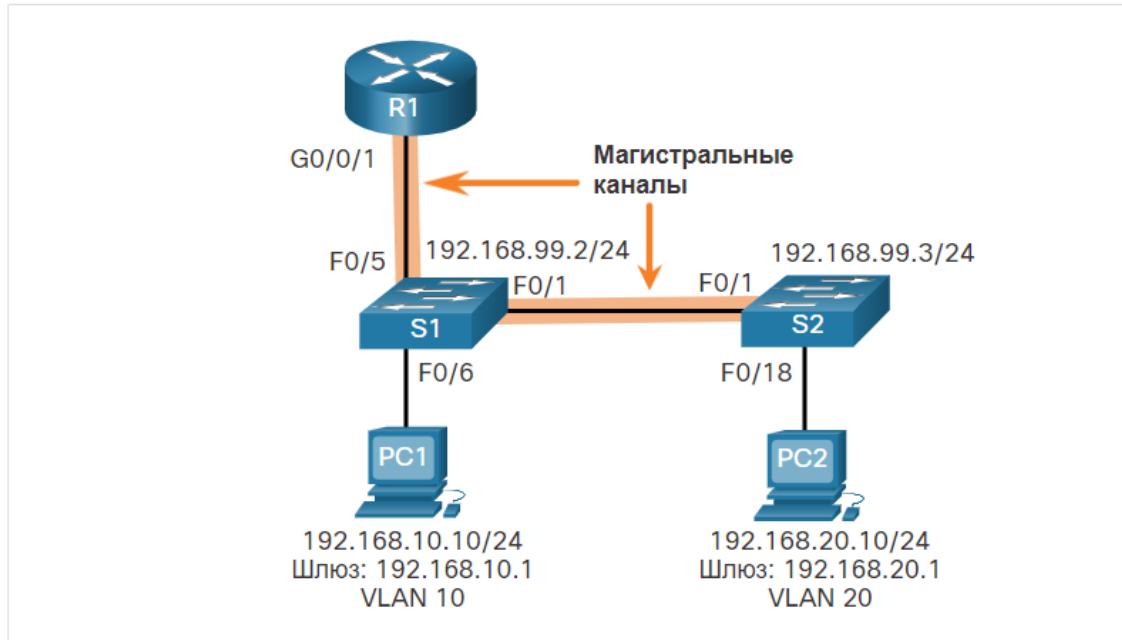
```
S(config)# interface interface_id
S(config-if)# switchport mode trunk
```

Настройка маршрутизатора:

```
R(config)# interface interface_id.subintreface_id
R(config-subif)# encapsulation dot1q vlan_id
R(config-subif)# ip address ip_address subnet_mask
```

В предыдущем разделе были перечислены три различных способа создания маршрутизации между VLAN, а также подробно описана устаревшая маршрутизация между VLAN. В этом разделе подробно описано, как настроить маршрутизацию между виртуальными локальными сетями на маршрутизаторе. Вы можете видеть на рисунке, что маршрутизатор не находится в центре топологии, а вместо этого, кажется, находится на палке рядом с границей, отсюда и название.

На рисунке интерфейс R1 GigabitEthernet 0/0/1 подключен к порту S1 FastEthernet 0/5. Порт S1 FastEthernet 0/1 подключен к порту S2 FastEthernet 0/1. Это магистральные каналы, которые необходимы для пересылки трафика внутри VLAN и между ними.



Для маршрутизации между VLAN интерфейс R1 GigabitEthernet 0/0/1 логически разделен на три подинтерфейса, как показано в таблице. В таблице также показаны три VLAN, которые будут настроены на коммутаторах.

Router R1 Subinterfaces

Подинтерфейс	VLAN	IP-адрес
G0/0/1.10	10	192.168.10.1/24
G0/0/1.20	20	192.168.20.1/24
G0/0/1.99	99	192.168.99.1/24

Предположим, что R1, S1 и S2 имеют начальные базовые конфигурации. В настоящее время PC1 и PC2 не могут **ping** друг друга, поскольку они находятся в отдельных сетях. Только S1 и S2 могут **ping** друг друга, но они, но не доступны PC1 или PC2, потому что они также в разных сетях.

Выполните следующие шаги для настройки S1 с созданием VLAN и транкингом:

Шаг 1. Создайте сети VLAN на коммутаторе S1 и присвойте им имена.

Шаг 2. Создайте интерфейс управления.

Шаг 3. Настройка портов доступа

Шаг 4. Настройте транковые порты.

1. Создайте сети VLAN и присвойте им имена.

Во-первых, создайте и назовите VLAN. VLAN создаются только после выхода из режима подконфигурации VLAN.

```
S1(config)# vlan 10
S1(config-vlan)# name LAN10
S1(config-vlan)# exit
S1(config)# vlan 20
S1(config-vlan)# name LAN20
S1(config-vlan)# exit
S1(config)# vlan 99
S1(config-vlan)# name Management
S1(config-vlan)# exit
S1(config)#
```

2. Создайте интерфейс управления.

Затем создайте интерфейс управления VLAN 99 вместе со шлюзом по умолчанию R1.

```
S1(config)# interface vlan 99
S1(config-if)# ip add 192.168.99.2 255.255.255.0
S1(config-if)# no shut
S1(config-if)# exit
S1(config)# ip default-gateway 192.168.99.1
S1(config)#

```

3. Настройка портов доступа.

Затем порт Fa0/6, подключающийся к PC1, настраивается в качестве порта доступа в VLAN 10. Предположим, что PC1 настроен с правильным IP-адресом и шлюзом по умолчанию.

```
S1(config)# interface fa0/6
S1(config-if)# switchport mode access
S1(config-if)# switchport access vlan 10
S1(config-if)# no shut
S1(config-if)# exit
S1(config)#

```

4. Настройте транковые порты.

Наконец, порты Fa0/1, подключающиеся к S2, и Fa0/5, подключающиеся к R1, настроены в качестве магистральных портов.

```
S1(config)# interface fa0/1
S1(config-if)# switchport mode trunk
S1(config-if)# no shut
S1(config-if)# exit
S1(config)# interface fa0/5
S1(config-if)# switchport mode trunk
S1(config-if)# no shut
S1(config-if)# end

```

Конфигурация S2 аналогична конфигурации S1.

```

S2(config)# vlan 10
S2(config-vlan)# name LAN10
S2(config-vlan)# exit
S2(config)# vlan 20
S2(config-vlan)# name LAN20
S2(config-vlan)# exit
S2(config)# vlan 99
S2(config-vlan)# name Management
S2(config-vlan)# exit
S2(config)#
S2(config)# interface vlan 99
S2(config-if)# ip add 192.168.99.3 255.255.255.0
S2(config-if)# no shut
S2(config-if)# exit
S2(config)# ip default-gateway 192.168.99.1
S2(config)# interface fa0/18
S2(config-if)# switchport mode access
S2(config-if)# switchport access vlan 20
S2(config-if)# no shut
S2(config-if)# exit
S2(config)# interface fa0/1
S2(config-if)# switchport mode trunk
S2(config-if)# no shut
S2(config-if)# exit
S2(config-if)# end
*Mar 1 00:23:52.137: %LINEPROTO-5-UPDOWN: Line prot
up

```

Для использования метода Router-on-a-Stick требуется настроить подинтерфейсы для каждой маршрутизируемой сети VLAN.

Подинтерфейс создается с помощью команды режима глобальной конфигурации **interface interface_id subinterface_id**. Синтаксис для подинтерфейсов следующий: сначала указывается физический интерфейс, в данном случае g0/0, затем точка и номер подинтерфейса. Хотя это не требуется, обычно сопоставляют номер подинтерфейса с номером VLAN.

Затем каждый подинтерфейс настраивается с помощью следующих двух команд:

- **encapsulation dot1q vlan_id [native]** - This command configures the subinterface to respond to 802.1Q encapsulated traffic from the specified *vlan-id*. The **native** ключевого слова добавляется только для установки собственной VLAN отличной от VLAN 1.
- **ip address ip-address subnet-mask** - Эта команда настраивает IPv4-адрес подинтерфейса. Этот адрес обычно служит шлюзом по умолчанию для данной VLAN.

Повторите процесс для каждой маршрутизируемой VLAN. Для осуществления маршрутизации каждому подинтерфейсу маршрутизатора необходимо назначить IP-адрес в своей подсети.

После создания всех подинтерфейсов включите физический интерфейс с помощью команды конфигурации **no shutdown** интерфейса. Если отключить физический интерфейс, то все подчиненные интерфейсы также отключаются.

В следующей конфигурации субинтерфейсы R1 G0/0/1 настроены для VLAN 10, 20 и 99.

```

R1(config)# interface G0/0/1.10
R1(config-subif)# description Default Gateway for VLAN 10
R1(config-subif)# encapsulation dot1Q 10
R1(config-subif)# ip add 192.168.10.1 255.255.255.0
R1(config-subif)# exit
R1(config)#
R1(config)# interface G0/0/1.20
R1(config-subif)# description Default Gateway for VLAN 20
R1(config-subif)# encapsulation dot1Q 20
R1(config-subif)# ip add 192.168.20.1 255.255.255.0
R1(config-subif)# exit
R1(config)#
R1(config)# interface G0/0/1.99
R1(config-subif)# description Default Gateway for VLAN 99
R1(config-subif)# encapsulation dot1Q 99
R1(config-subif)# ip add 192.168.99.1 255.255.255.0
R1(config-subif)# exit
R1(config)#
R1(config)# interface G0/0/1
R1(config-if)# description Trunk link to S1
R1(config-if)# no shut
R1(config-if)# end
R1#

```

Помимо использования **ping** между устройствами, следующие **show** команды могут использоваться для проверки и устранения неполадок конфигурации маршрутизатора на палке.

- **show ip route**
- **show ip interface brief**
- **show interfaces**
- **show interfaces trunk**

Убедитесь, что подинтерфейсы отображаются в таблице маршрутизации R1 с помощью команды **show ip route**. Обратите внимание, что для каждой маршрутизируемой VLAN существует три подключенных маршрута (C) и соответствующие интерфейсы выхода. Вывод подтверждает, что правильные подсети, VLAN и подынтерфейсы активны.

```

R1# show ip route | begin Gateway
Gateway of last resort is not set
    192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.10.0/24 is directly connected, GigabitEthernet0/0/1.10
L 192.168.10.1/32 is directly connected, GigabitEthernet0/0/1.10
    192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.20.0/24 is directly connected, GigabitEthernet0/0/1.20
L 192.168.20.1/32 is directly connected, GigabitEthernet0/0/1.20
    192.168.99.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.99.0/24 is directly connected, GigabitEthernet0/0/1.99
L 192.168.99.1/32 is directly connected, GigabitEthernet0/0/1.99
R1#

```

Другая полезная команда маршрутизатора **show ip interface brief**, , как показано на рисунке, подтверждают, что подинтерфейсы имеют правильный IPv4 адреса и что они работают.

```
R1# show ip interface brief | include up
GigabitEthernet0/0/1 unassigned YES unset up up
Gi0/0/1.10 192.168.10.1 YES manual up up
Gi0/0/1.20 192.168.20.1 YES manual up up
Gi0/0/1.99 192.168.99.1 YES manual up up
R1#
```

Подинтерфейсы можно проверить с помощью команды **show interfaces subinterface-id**, как показано на рисунке.

```
R1# show interfaces g0/0/1.10
GigabitEthernet0/0/1.10 is up, line protocol is up
  Hardware is ISR4221-2x1GE, address is 10b3.d605.0301 (bia 10b3.d605.0301)
  Description Default Gateway for VLAN 10
  Internet address is 192.168.10.1/24
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Hardware is ISR4221-2x1GE, address is 10b3.d605.0301 (bia 10b3.d605.0301)
  ARP type: ARPA, ARP Timeout 04:00:00
  Keepalive not supported
  Last clearing of show interface counters never
R1#
```

Неправильная конфигурация также может быть на порту магистрального коммутатора. Поэтому также полезно проверить активные магистральные каналы на коммутаторе уровня 2 с помощью команды **show interfaces trunk**, как показано в выходных данных. Выходные данные подтверждают, что канал до R1 является магистральным для необходимых VLAN.

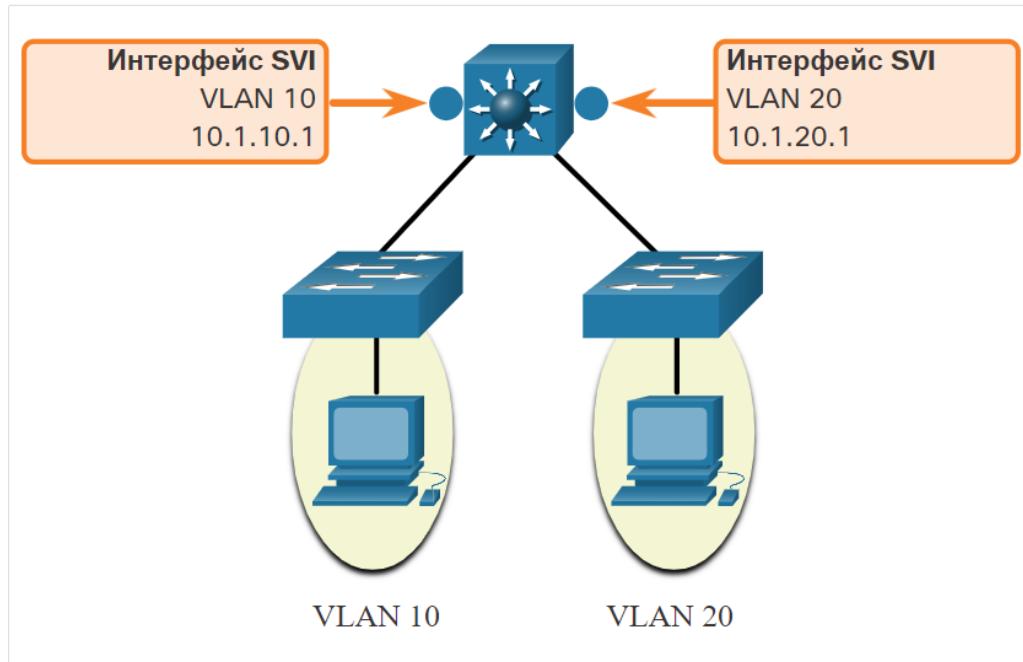
Примечание: Хотя VLAN 1 не была настроена явно, она была включена автоматически, так как управляемый трафик магистральных каналов всегда будет перенаправляться на VLAN 1.

```
S1# show interfaces trunk
Port Mode Encapsulation Status Native vlan
  Fa0/1 on 802.1q trunking 1
  Fa0/5 on 802.1q trunking 1
Port Vlans allowed on trunk
  Fa0/1 1-4094
  Fa0/5 1-4094
Port Vlans allowed and active in management domain
  Fa0/1 1,10,20,99
  Fa0/5 1,10,20,99
Port Vlans in spanning tree forwarding state and not pruned
  Fa0/1 1,10,20,99
  Fa0/5 1,10,20,99
S1#
```

Маршрутизация между VLAN на коммутаторе уровня 3

Современный способ выполнения маршрутизации между VLAN заключается в использовании коммутаторов уровня 3 и коммутируемых виртуальных интерфейсов (SVI). Как показано на рисунке, SVI — это виртуальный интерфейс, настраиваемый в многоуровневом коммутаторе.

Примечание: Коммутатор уровня 3 также называется многоуровневым коммутатором, поскольку он работает на уровнях 2 и 3. Однако в этом курсе мы используем термин «Коммутатор уровня 3».



SVI для маршрутизации между VLAN создаются так же, как и интерфейс VLAN управления. Интерфейс SVI можно создать для любой сети VLAN, существующей на коммутаторе. Несмотря на то, что SVI является виртуальным, он выполняет те же функции для VLAN, что и интерфейс маршрутизатора. Интерфейс SVI для сети VLAN обеспечивает обработку пакетов 3-го уровня в обоих направлениях через порты коммутатора, связанные с этой VLAN.

Ниже приведены преимущества использования коммутаторов уровня 3 для маршрутизации между VLAN:

- Это более быстрая маршрутизация, чем конфигурация router-on-stick, поскольку и коммутация, и маршрутизация выполняются аппаратно;
- для маршрутизации не требуются внешние каналы от коммутатора к маршрутизатору.
- Они не ограничиваются одним каналом, поскольку EtherChannels уровня 2 можно использовать в качестве магистральных каналов между коммутаторами для увеличения пропускной способности.
- Задержка намного короче, поскольку для маршрутизации в другую сеть данным не нужно покидать коммутатор.
- Они чаще развертываются в локальной сети кампуса, чем маршрутизаторы.

Единственным недостатком является то, что коммутаторы уровня 3 дороже.

f. Многоуровневый коммутатор. Маршрутизация на многоуровневом коммутаторе.

Настройка коммутатора 3-его уровня

Создание VLAN:

```
S(config)# vlan vlan_id
S(config-vlan)# name vlan_name
```

Настройка интерфейса SVI:

```
S(config)# interface vlan_id
S(config-if)# ip address ip_address subnet_mask
```

Включение маршрутизации:

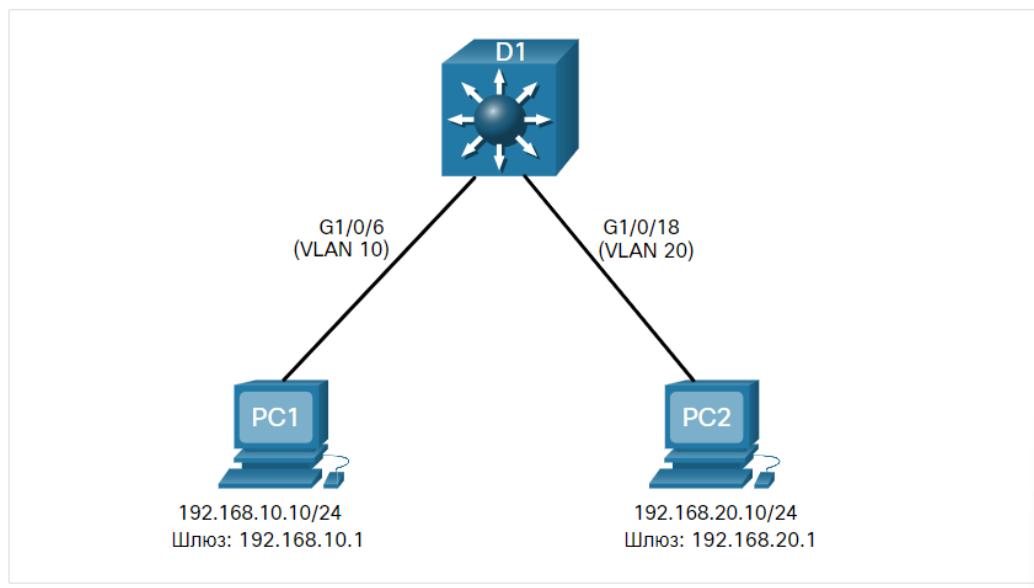
```
S(config)# ip routing
```

Возможности коммутатора уровня 3 включают в себя возможность выполнения следующих действий:

- Маршрутизация от одной VLAN к другой с использованием нескольких коммутируемых виртуальных интерфейсов (SVI).
- Преобразовать порт коммутатора уровня 2 в интерфейс уровня 3 (т.е. маршрутизуемый порт). Маршрутизуемый порт — простой интерфейс 3-го уровня, аналогичный физическому интерфейсу на маршрутизаторе Cisco IOS.

Для обеспечения маршрутизации между VLAN коммутаторы уровня 3 используют SVI. SVI настраиваются с помощью той же команды **interface vlan *vlan-id***, которая используется для создания SVI управления на коммутаторе уровня 2. Для каждой маршрутизуемой сети VLAN необходимо создать SVI уровня 3.

На рисунке коммутатор уровня 3 D1 подключен к двум узлам в разных VLAN. PC1 находится в VLAN 10, а PC2 – в VLAN 20, как показано на рисунке. Коммутатор уровня 3 будет предоставлять услуги маршрутизации между VLAN.



В таблице показаны IP-адреса для каждой VLAN.

D1 VLAN IP Addresses

Интерфейсы VLAN	IP-адрес
10	192.168.10.1/24
20	192.168.20.1/24

Выполните следующие шаги для настройки S1 с созданием VLAN и магистральных каналов:

Шаг 1. Создайте сети VLAN.

Шаг 2. Создание SVI интерфейсов VLAN.

Шаг 3. Настройка портов доступа

Шаг 4. Активация IP-маршрутизации.

1. Создайте сети VLAN.

Сначала создайте две VLAN, как показано на выходных данных.

```
D1(config)# vlan 10
D1(config-vlan)# name LAN10
D1(config-vlan)# vlan 20
D1(config-vlan)# name LAN20
D1(config-vlan)# exit
D1(config)#[/pre>
```

2. Создание SVI интерфейсов VLAN.

Сконфигурируйте SVI для VLAN 10. Настроенные IP-адреса будут служить шлюзами по умолчанию для узлов в соответствующих VLAN. Обратите внимание на информационные сообщения, показывающие линейный протокол на обоих SVI, измененный на вверх.

```
D1(config)# interface vlan 10
D1(config-if)# description Default Gateway SVI for 192.168.10.0/24
D1(config-if)# ip add 192.168.10.1 255.255.255.0
D1(config-if)# no shut
D1(config-if)# exit
D1(config)#
D1(config)# int vlan 20
D1(config-if)# description Default Gateway SVI for 192.168.20.0/24
D1(config-if)# ip add 192.168.20.1 255.255.255.0
D1(config-if)# no shut
D1(config-if)# exit
D1(config)#
*Sep 17 13:52:16.053: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up
*Sep 17 13:52:16.160: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up
```

3. Настройка портов доступа.

Затем настройте порты доступа, подключающиеся к узлам, и назначьте их соответствующим VLAN.

```
D1(config)# interface GigabitEthernet1/0/6
D1(config-if)# description Access port to PC1
D1(config-if)# switchport mode access
D1(config-if)# switchport access vlan 10
D1(config-if)# exit
D1(config)#
D1(config)# interface GigabitEthernet1/0/18
D1(config-if)# description Access port to PC2
D1(config-if)# switchport mode access
D1(config-if)# switchport access vlan 20
D1(config-if)# exit
```

4. Активация IP-маршрутизации.

Наконец, включите маршрутизацию IPv4 с помощью команды глобальной конфигурации **ip routing**, чтобы разрешить обмен трафиком между VLAN 10 и 20. Эта команда должна быть настроена для включения маршрутизации между VAN на коммутаторе уровня 3 для протокола IPv4.

```
D1(config)# ip routing
D1(config)#
```

Проверка маршрутизации между VLAN коммутатором уровня 3

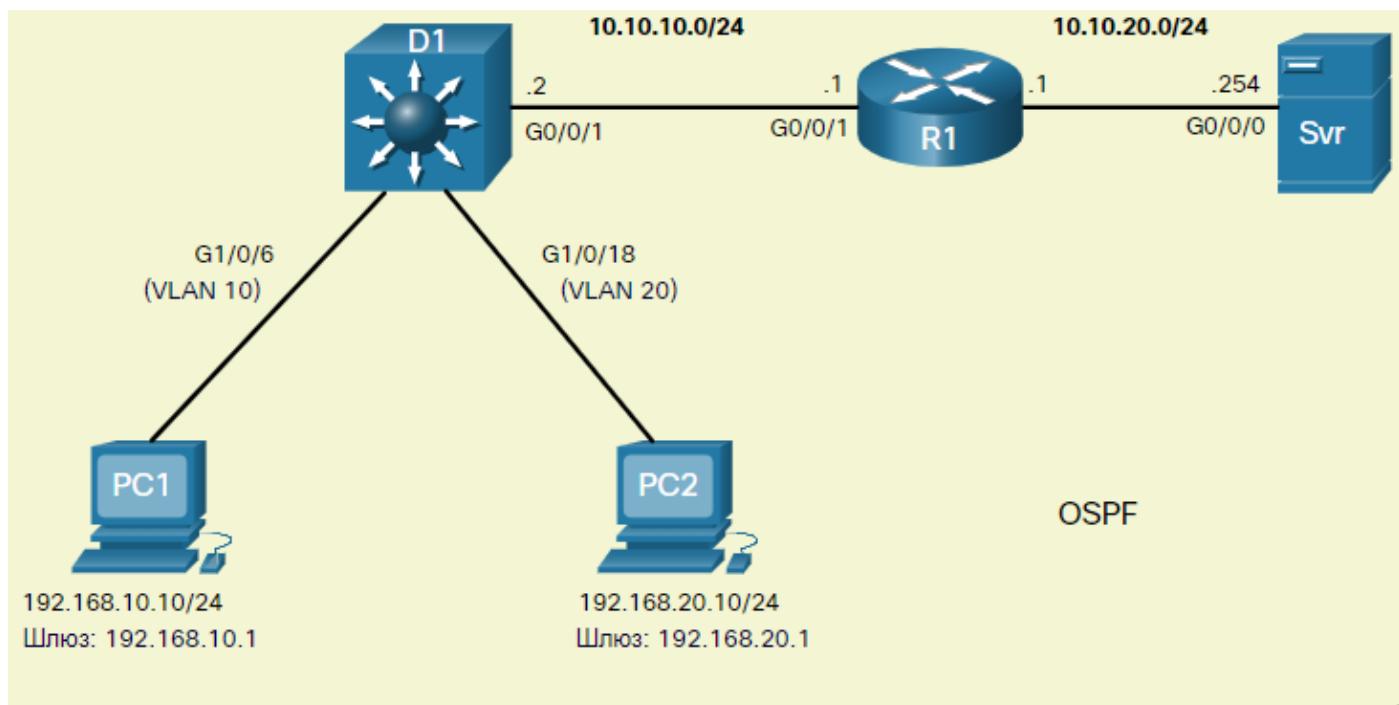
Маршрутизация между VLAN с помощью коммутатора уровня 3 проще в настройке, чем метод ROS. После завершения настройки конфигурация может быть проверена путем тестирования подключения между узлами.

С узла проверьте подключение к узлу в другой VLAN с помощью команды **ping**. Лучше сначала проверить текущую конфигурацию IP хоста с помощью команды Windows **ipconfig**. Выходные данные подтверждают адрес IPv4 и шлюз по умолчанию PC1.

Сценарий маршрутизации на коммутаторе уровня 3

На рисунке ранее настроенный коммутатор D1 уровня 3 теперь подключен к R1. R1 и D1 находятся в домене протокола маршрутизации OSPF. Предположим, что маршрутизация между VLAN успешно реализована на D1. Интерфейс G0/0/1 R1 также был настроен и включен. Кроме того, R1 использует OSPF для объявления своих двух сетей: 10.10.10.0/24 и 10.20.0.0/24.

Примечание: Настройка маршрутизации OSPF рассматривается в другом курсе. В этом модуле команды конфигурации OSPF будут даны вам во всех заданиях и экзаменах. Для включения маршрутизации OSPF на коммутаторе уровня 3 не требуется понимание конфигурации.



Выполните следующие шаги, чтобы настроить D1 для маршрутизации с R1:

Шаг 1. Настройте маршрутизируемый порт.

Шаг 2. Включите маршрутизацию.

Шаг 3. Настройте маршрутизацию.

Шаг 4. Проверка маршрутизации.

Шаг 5. Проверьте подключение.

1. Настройте маршрутизуемый порт.

Настройте G1/0/1 как маршрутизуемый порт, назначьте ему адрес IPv4 и включите его.

```
D1(config)# interface GigabitEthernet1/0/1
D1(config-if)# description routed Port Link to R1
D1(config-if)# no switchport
D1(config-if)# ip address 10.10.10.2 255.255.255.0
D1(config-if)# no shut
D1(config-if)# exit
D1(config)#

```

2. Включите маршрутизацию.

Убедитесь, что маршрутизация IPv4 включена с помощью команды глобальной конфигурации **ip routing**.

```
D1(config)# ip routing
D1(config)#

```

3. Настройте маршрутизацию.

Настройте протокол маршрутизации OSPF для объявления сетей VLAN 10 и VLAN 20 вместе с сетью, подключенной к R1. Обратите внимание на сообщение, информирующее вас о том, что смежность установлена с R1.

```
D1(config)# router ospf 10
D1(config-router)# network 192.168.10.0 0.0.0.255 area 0
D1(config-router)# network 192.168.20.0 0.0.0.255 area 0
D1(config-router)# network 10.10.10.0 0.0.0.3 area 0
D1(config-router)# ^Z
D1#
*Sep 17 13:52:51.163: %OSPF-5-ADJCHG: Process 10, Nbr 10.20.20.1 on GigabitEthernet1/0/1 from LOADING to
FULL, Loading Done
D1#

```

4. Проверка маршрутизации.

Проверка таблицы маршрутизации D1. Обратите внимание, что D1 теперь имеет маршрут к сети 10.20.20.0/24.

```
D1# show ip route | begin Gateway
Gateway of last resort is not set
    10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C 10.10.10.0/30 is directly connected, GigabitEthernet1/0/1
L 10.10.10.2/32 is directly connected, GigabitEthernet1/0/1
O 10.20.20.0/24 [110/2] via 10.10.10.1, 00:00:06, GigabitEthernet1/0/1
    192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.10.0/24 is directly connected, Vlan10
L 192.168.10.1/32 is directly connected, Vlan10
    192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.20.0/24 is directly connected, Vlan20
L 192.168.20.1/32 is directly connected, Vlan20
D1#
```

5. Проверьте подключение.

В это время PC1 и PC2 могут выполнить эхо-запрос сервера, подключенного к R1.

```
C:\Users\PC1> ping 10.20.20.254
Pinging 10.20.20.254 with 32 bytes of data:
Request timed out.
Reply from 10.20.20.254: bytes=32 time<1ms TTL=127
Reply from 10.20.20.254: bytes=32 time<1ms TTL=127
Reply from 10.20.20.254: bytes=32 time<1ms TTL=127
Ping statistics for 10.20.20.254:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss).
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
C:\Пользователи\ PC1>
! =====
C:\Users\PC2> ping 10.20.20.254
Pinging 10.20.20.254 with 32 bytes of data:
Reply from 10.20.20.254: bytes=32 time<1ms TTL=127
Ping statistics for 10.20.20.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
C:\Users\PC2>
```

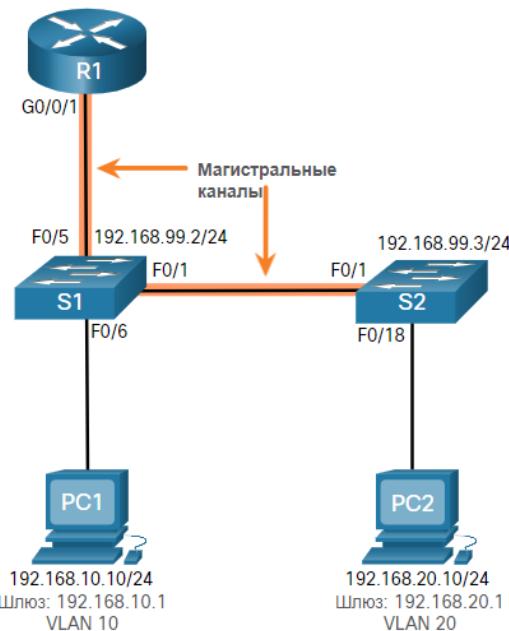
g. Поиск и устранение неисправностей при работе с VLAN.

Существует ряд причин, по которым конфигурация маршрутизации между VLAN может не работать. Все они связаны с проблемами подключения. Сначала проверьте физический уровень, чтобы устранить любые проблемы, при которых кабель может быть подключен к неправильному порту. Если подключения верны, используйте список в таблице по другим общим причинам, по которым может произойти сбой подключения между VLAN.

Тип проблемы	Как исправить	Как проверить
Отсутствующие сети VLAN	<ul style="list-style-type: none">Создайте (или повторно создайте) VLAN, если она не существует.Убедитесь, что порт хоста назначен правильной VLAN.	<code>show vlan [brief]</code> <code>show interfaces switchport ping</code>
Проблемы магистрального порта коммутатора	<ul style="list-style-type: none">Убедитесь, что магистральные соединения настроены правильно.Убедитесь, что порт является магистральным портом и включен.	<code>show interfaces trunk</code> <code>show running-config</code>
Неполадки в работе порта коммутатора	<ul style="list-style-type: none">Назначьте порт соответствующей сети VLAN.Убедитесь, что порт является портом доступа и включен.Неправильно настроен узел в неправильной подсети.	<code>show interfaces switchport</code> <code>show running-config</code> <code>interface</code> <code>ipconfig</code>
Неполадки в настройках маршрутизатора	<ul style="list-style-type: none">IPv4-адрес подинтерфейса маршрутизатора настроен неправильно.Подинтерфейс маршрутизатора назначается с идентификатором VLAN.	<code>show ip interface brief</code> <code>show interfaces</code>

Примеры некоторых из этих проблем маршрутизации между VLAN теперь будут рассмотрены более подробно.

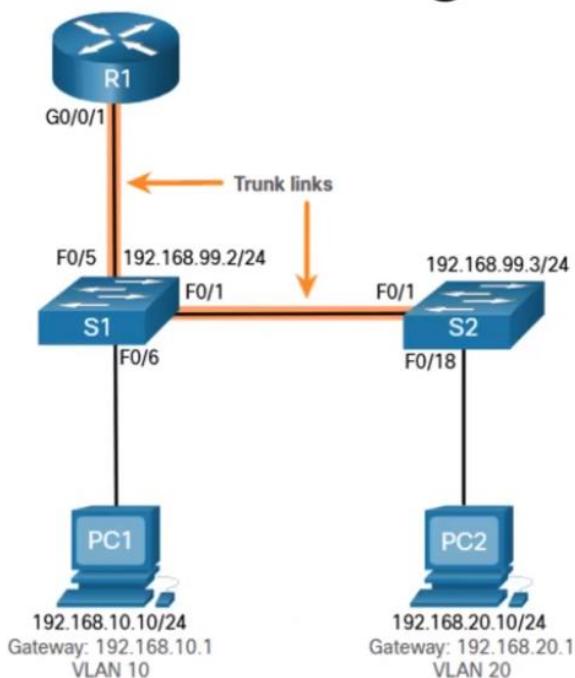
Эта топология будет использоваться для всех этих проблем.



Router R1 Subinterfaces

Подинтерфейс	VLAN	IP-адрес
G0/0/0.10	10	192.168.10.1/24
G0/0/0.20	20	192.168.20.1/24
G0/0/0.30	99	192.168.99.1/24

Troubleshooting



Проблемы:

- Отсутствует или неверно назначена VLAN
- Кабели подключены не к тем портам
- Неверно настроены trunk порты
- Неверно настроены подинтерфейсы
- Неверно настроена маршрутизация

Проблема подключения между VLAN может быть вызвана отсутствием VLAN. VLAN может отсутствовать, если она не была создана, случайно удалена или не разрешена на магистрального канала.

Например, PC1 в настоящее время подключен к VLAN 10, как показано в выходных данных **show vlan brief** команды.

VLAN Name	Status	Ports
1 default	active	Fa0/2, Fa0/3, Fa0/4, Fa0/7 Fa0/8, Fa0/9, Fa0/10, Fa0/11 Fa0/12, Fa0/13, Fa0/14, Fa0/15 Fa0/16, Fa0/17, Fa0/18, Fa0/19 Fa0/20, Fa0/21, Fa0/22, Fa0/23 Fa0/24, Gi0/1, Gi0/2
10 LAN10	active	Fa0/6
20 LAN20	active	
99 Management	active	
1002 fddi-default	act/unsup	
1003 token-ring-default	act/unsup	
1004 fddinet-default	act/unsup	
1005 trnet-default	act/unsup	

Теперь предположим, что VLAN 10 случайно удаляется, как показано в следующем выводе.

VLAN Name	Status	Ports
1 default	active	Fa0/2, Fa0/3, Fa0/4, Fa0/7 Fa0/8, Fa0/9, Fa0/10, Fa0/11 Fa0/12, Fa0/13, Fa0/14, Fa0/15 Fa0/16, Fa0/17, Fa0/18, Fa0/19 Fa0/20, Fa0/21, Fa0/22, Fa0/23 Fa0/24, Gi0/1, Gi0/2
20 LAN20	active	
99 Management	active	
1002 fddi-default	act/unsup	
1003 token-ring-default	act/unsup	
1004 fddinet-default	act/unsup	
1005 trnet-default	act/unsup	

Обратите внимание, что VLAN 10 теперь отсутствует в выходных данных. Также обратите внимание, что порт Fa0/6 не был переназначен во VLAN по умолчанию. При удалении сети VLAN все порты, назначенные этой сети, становятся неактивными. Они остаются связанными с этой сетью VLAN (и, следовательно, неактивными), пока не будут назначены новой сети VLAN.

Используйте команду **show interface interface-id switchport** для проверки членства в VLAN.

S1(config)# do show interface fa0/6 switchport	
Name:	Fa0/6
Switchport:	Enabled
Administrative Mode:	static access
Operational Mode:	static access
Administrative Trunking Encapsulation:	dot1q
Operational Trunking Encapsulation:	native
Negotiation of Trunking:	Off
Access Mode VLAN:	10 (Inactive)
Trunking Native Mode VLAN:	1 (default)
Administrative Native VLAN tagging:	enabled
Voice VLAN:	none
(Output omitted)	

Повторное создание отсутствующей VLAN автоматически переназначает хосты в нее, как показано в следующих выходных данных.

```
S1(config)# vlan 10
S1(config-vlan)# do show vlan brief
VLAN Name          Status    Ports
----- 
1    default        active    Fa0/2, Fa0/3, Fa0/4, Fa0/7
                           Fa0/8, Fa0/9, Fa0/10, Fa0/11
                           Fa0/12, Fa0/13, Fa0/14, Fa0/15
                           Fa0/16, Fa0/17, Fa0/18, Fa0/19
                           Fa0/20, Fa0/21, Fa0/22, Fa0/23
                           Fa0/24, Gi0/1, Gi0/2
20   LAN20          active
99   Management     active
1002 fddi-default   act/unsup
1003 token-ring-default   act/unsup
1004 fddinet-default   act/unsup
1005 trnet-default    act/unsup
S1(config-vlan)#

```

Обратите внимание, что VLAN не была создана должным образом. Причина заключается в том, что для создания VLAN необходимо выйти из режима подконфигурации VLAN, как показано в следующих выходных данных.

```
S1(config-vlan)# exit
S1(config)# vlan 10
S1(config)# do show vlan brief
VLAN Name          Status    Ports
----- 
1    default        active    Fa0/2, Fa0/3, Fa0/4, Fa0/7
                           Fa0/8, Fa0/9, Fa0/10, Fa0/11
                           Fa0/12, Fa0/13, Fa0/14, Fa0/15
                           Fa0/16, Fa0/17, Fa0/18, Fa0/19
                           Fa0/20, Fa0/21, Fa0/22, Fa0/23
                           Fa0/24, Gi0/1, Gi0/2
10   VLAN0010       active    Fa0/6

```

Проблемы магистрального порта коммутатора

Другая проблема маршрутизации между VLAN связана с неправильной конфигурацией портов коммутатора. В устаревшем решении маршрутизации между VLAN это может быть вызвано, когда порт доступа маршрутизатора не назначен правильной VLAN.

Однако при решении ROS наиболее распространенной причиной является неправильная настройка магистрального порта.

Например, предположим, что до недавнего времени PC1 мог подключаться к узлам в других VLAN. Краткий обзор журналов обслуживания показал, что коммутатор S1 уровня 2 был недавно доступен для текущего обслуживания. Таким образом вы подозреваете, что проблема может быть связана с этим коммутатором.

На S1 убедитесь, что порт, подключаемый к R1 (например, F0/5), правильно настроен в качестве магистрального канала с помощью команды **show interfaces trunk**, как показано на рисунке.

```
S1# show interfaces trunk
Port      Mode           Encapsulation  Status      Native vlan
Fa0/1     on            802.1q         trunking    1
Port      Vlans allowed on trunk
Fa0/1     1-4094
Port      Vlans allowed and active in management domain
Fa0/1     1,10,20,99
Port      Vlans in spanning tree forwarding state and not pruned
Fa0/1     1,10,20,99
S1#

```

Порт Fa0/5, подключающийся к R1, таинственно отсутствует на выходе. Проверьте конфигурацию интерфейса с помощью команды **show running-config interface fa0/5**, как показано на рисунке.

```
S1# show running-config | include interface fa0/5
Building configuration...
Current configuration : 96 bytes
!
interface FastEthernet0/5
  description Trunk link to R1
  switchport mode trunk
  shutdown
end
S1#
```

Как вы можете видеть, порт был случайно отключен. Чтобы устранить проблему, повторно включите порт и проверьте состояние транка, как показано на выходных данных.

```
S1(config)# interface fa0/5
S1(config-if)# no shut
S1(config-if)#
*Mar  1 04:46:44.153: %LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to up
S1(config-if)#
*Mar  1 04:46:47.962: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed state to up
S1(config-if)# do show interface trunk
Port      Mode          Encapsulation  Status       Native vlan
Fa0/1    on           802.1q        trunking    1
Fa0/5    on           802.1q        trunking    1
Port      Vlans allowed on trunk
Fa0/1    1-4094
Fa0/5    1-4094
Port      Vlans allowed and active in management domain
Fa0/1    1,10,20,99
Fa0/5    1,10,20,99
Port      Vlans in spanning tree forwarding state and not pruned
Fa0/1    1,10,20,99
Fa0/1    1,10,20,99
S1(config-if)#

```

Чтобы уменьшить риск того, что нарушение канала связи между коммутаторами нарушит и маршрутизацию между сетями VLAN, в проекте сети должны быть предусмотрены резервные каналы связи и альтернативные пути маршрутизации.

Неполадки в работе порта коммутатора

В настройках коммутатора могут присутствовать проблемы, поэтому рекомендуется использовать специальные команды для проверки конфигурации и определения неполадок.

Предположим, PC1 имеет правильный IPv4 адрес и шлюз по умолчанию, но не может использовать **ping** до собственного шлюза по умолчанию. PC1 должен быть подключен к порту VLAN 10.

Проверьте конфигурацию порта на S1 с помощью команды **show interfaces interface-id switchport**.

```
S1# show interface fa0/6 switchport
Name: Fa0/6
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
```

Порт Fa0/6 настроен как порт доступа, как указано в выводе «статический доступ». (“static access”) Тем не менее, что он не был настроен для работы во VLAN 10. Проверьте настройку интерфейса:

```
S1# show running-config interface fa0/6
Building configuration...
Current configuration : 87 bytes
!
interface FastEthernet0/6
  description PC-A access port
  switchport mode access
end
S1#
```

Назначьте порт Fa0/6 во VLAN 10 и проверьте назначение порта.

```
S1# configure terminal
S1(config)# interface fa0/6
S1(config-if)# switchport access vlan 10
S1(config-if)#
S1(config-if)# do show interface fa0/6 switchport
Name: Fa0/6
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 10 (VLAN0010)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
(Output omitted)
```

PC1 теперь может взаимодействовать с хостами в других VLAN.

Неполадки в настройках маршрутизатора

Проблемы конфигурации маршрутизатора ROS обычно связаны с неправильными конфигурациями подинтерфейса. Например, был настроен неверный IP-адрес или неправильный идентификатор VLAN был присвоен подинтерфейсу.

Например, R1 должен предоставлять маршрутизацию между VLAN для пользователей в VLAN 10, 20 и 99. Однако пользователи VLAN 10 не могут связаться с любой другой VLAN.

Вы проверили магистральный канал коммутатора и все выглядит в порядке. Проверьте состояние подинтерфейса с помощью команды **show ip interface brief**.

```
R1# show ip interface brief
Interface          IP-Address      OK? Method Status        Protocol
GigabitEthernet0/0/0 unassigned     YES unset administratively down down
GigabitEthernet0/0/1 unassigned     YES unset up          up
Gi0/0/1.10         192.168.10.1   YES manual up          up
Gi0/0/1.20         192.168.20.1   YES manual up          up
Gi0/0/1.99         192.168.99.1   YES manual up          up
Serial0/1/0         unassigned     YES unset administratively down down
Serial0/1/1         unassigned     YES unset administratively down down
R1#
```

Подинтерфейсам были назначены правильные адреса IPv4, и они работают.

Проверьте, какие VLAN включен каждый из подинтерфейсов. Для этого полезна команда **show interfaces**, но она генерирует много дополнительных выходных данных. Вывод команды можно уменьшить с помощью фильтров команд IOS, как показано на выходных данных.

```
R1# show interfaces | include Gig|802.1Q
GigabitEthernet0/0/0 is administratively down, line protocol is down
GigabitEthernet0/0/1 is up, line protocol is up
  Encapsulation 802.1Q Virtual LAN, Vlan ID  1., loopback not set
GigabitEthernet0/0/1.10 is up, line protocol is up
  Encapsulation 802.1Q Virtual LAN, Vlan ID  100.
GigabitEthernet0/0/1.20 is up, line protocol is up
  Encapsulation 802.1Q Virtual LAN, Vlan ID  20.
GigabitEthernet0/0/1.99 is up, line protocol is up
  Encapsulation 802.1Q Virtual LAN, Vlan ID  99.
R1#
```

Символ (**|**) вместе с некоторыми ключевыми словами **select** является полезным методом для вывода команды фильтрации. В этом примере ключевое слово **include** было использовано для идентификации того, что будут отображаться только строки, содержащие буквы «**Gig**» или «**802.1Q**». Из-за того, как **show interface** выходные данные перечислены естественным образом, с помощью этих фильтров создается сжатый список интерфейсов и назначенных им VLAN.

Обратите внимание, что интерфейс G0/0/1.10 неправильно назначен VLAN 100 вместо VLAN 10. Это подтверждается при просмотре конфигурации подинтерфейса R1 GigabitEthernet 0/0/1.10, как показано на рисунке.

```
R1# show running-config interface g0/0/1.10
Building configuration...
Current configuration : 146 bytes
!
interface GigabitEthernet0/0/1.10
  description Default Gateway for VLAN 10
  encapsulation dot1Q 100
  ip address 192.168.10.1 255.255.255.0
end
R1#
```

Чтобы исправить эту проблему, настройте подчиненный интерфейс G0/0.10 на правильную сеть VLAN с помощью команды режима глобальной конфигурации подинтерфейса **encapsulation dot1q 10****encapsulation dot1q 10****encapsulation dot1q 10**.

```
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface gigabitEthernet 0/0/1.10
R1(config-subif)# encapsulation dot1Q 10
R1(config-subif)# end
R1#
R1# show interfaces | include Gig|802.1Q
GigabitEthernet0/0/0 is administratively down, line protocol is down
GigabitEthernet0/0/1 is up, line protocol is up
  Encapsulation 802.1Q Virtual LAN, Vlan ID 1., loopback not set
GigabitEthernet0/0/1.10 is up, line protocol is up
  Encapsulation 802.1Q Virtual LAN, Vlan ID 10.
GigabitEthernet0/0/1.20 is up, line protocol is up
  Encapsulation 802.1Q Virtual LAN, Vlan ID 20.
GigabitEthernet0/0/1.99 is up, line protocol is up
R1#
```

При назначении подинтерфейса верной сети VLAN он становится доступным для устройств в этой VLAN, а маршрутизатор может осуществлять маршрутизацию между VLAN.

До

!!! Необходимо знать команды настройки и отладки.