**1 Понятие системного администрирования**

*Системное администрирование* призвано решать широкий круг задач, связанных с созданием и поддержкой в работоспособном состоянии сложной информационной системы, включающей различные аппаратные и программные средства. В настоящее время практически невозможно отделить сетевое администрирование от администрирования вообще. Системным администрированием занимается *системный администратор*

Существуют две базовые стратегии администрирования:

1. *Распределенное* -- нет единого центра, регламентирующего политику администрирования

2. *Централизованное* -- политика администрирования регламентируется единым центром

Так как основная часть повседневной работы системного администратора связана с ПО, нужно кратко оговорить ряд моментов, касающихся этого самого ПО. В первую очередь речь идет об установке и настройке (плюс удалении) ПО.

Установка ПО может происходить по -разному:

1. Просто копирование исполняемых и вспомогательных файлов «вручную» безо всяких проверок.

2. Компиляция исходных текстов, проверка зависимостей и копирование файлов с помощью стандартного набора специальных консольных команд.

3. Автоматическое, но контролируемое, выполнение проверок, копирование подготовленных файлов и осуществление других действий с помощью специальной программы, внешней по отношению к устанавливаемому ПО, -- обычно называемой пакетным менеджером (packet manager).

4. Выполнение аналогичных действий с помощью специальной программы, частично или полностью интегрированной в устанавливаемое ПО, -- называемой установщиком (installer).

Настройка, заключается в конфигурировании, то есть в изменении значений обязательных и опциональных параметров со значений по умолчанию на нужные значения. Часто есть возможность выполнять конфигурирование в режиме диалога -- с помощью визарда или, по-другому, мастера (wizard).

Термин «администрирование» был заимствован с сохранением семантики. И его лучше отделять от терминов «контроль» и «управление». В русском языке, при использовании термина «контроль», в первую очередь, подразумевают, что первично отслеживание состояния объекта. При использовании же термина «управление» – что первично воздействие на объект.

**2 Выбор программного обеспечения администратором**

Базовая классификация ПО заключается в его разделении на:

1. Системное -- реализует функционал различных подсистем ОС и позволяет контролировать ОС (само по себе «никому не нужно»).

2. Прикладное -- позволяет решать конкретные прикладные задачи («интересно» пользователям).

3. Инструментальное -- позволяет разрабатывать и тестировать другое ПО.

4. Встраиваемое (embedded) -- позволяет управлять некоторым устройством («неотделимо» от устройства для которого предназначено).

Пять основных критериев выбора ПО:

1. Степень соответствия требованиям (сугубо техническим и другим).

2. Стоимость (приобретения, освоения, использования).

3. Доступность (сложность приобретения и освоения).

4. Эргономичность (сложность использования).

5. Качество технической поддержки (при возникновении проблем).

В отношении ОС, ПО может быть:

1. Native -- оригинальным (от того же разработчика).

2. Third party -- сторонним (от других разработчиков).

При выборе сетевого программного обеспечения надо в первую очередь учитывать следующие факторы:

* какую сеть оно поддерживает: одноранговую сеть, сеть на основе сервера или оба этих типа;
* какое максимальное количество пользователей допускается (лучше брать с запасом не менее 20%);
* какое количество серверов можно включить и какие типы серверов возможны;
* какова совместимость с разными операционными системами и разными компьютерами, а также с другими сетевыми средствами;
* каков уровень производительности программных средств в различных режимах работы;
* какова степень надежности работы, каковы разрешенные режимы доступа и степень защиты данных;
* и, возможно, главное - какова стоимость программного обеспечения.

Еще до установки сети необходимо решить вопрос об управлении сетью.

**3 Установка программного обеспечения администратором**

Установка ПО может происходить по-разному:

1. Просто копирование исполняемых и вспомогательных файлов «вручную» без всяких проверок.

2. Компиляция исходных текстов, проверка зависимостей и копирование файлов с помощью стандартного набора специальных консольных команд.

3. Автоматическое, но контролируемое, выполнение проверок, копирование подготовленных файлов и осуществление других действий с помощью пакетного менеджера (packet manager).

4. Выполнение аналогичных действий с помощью специальной программы, частично или полностью интегрированной в устанавливаемое ПО, -- называемой установщиком (installer).

Некоторые компании разрабатывают более сложные программные средства для автоматизации масштабной установки ПО на большое количество компьютеров (automated software deployment). Примером может служить IBM Tivoli.

**4 Сопровождение программного обеспечения администратором**

Сопровождение ПО - это одна из фаз жизненного цикла программного обеспечения, следующая за фазой передачи ПО в эксплуатацию. В ходе сопровождения в программу вносятся изменения, с тем, чтобы исправить обнаруженные в процессе использования дефекты и недоработки, а также для добавления новой функциональности, с целью повысить удобство использования и применимость ПО.

Системный администратор сам должен владеть навыками тестирования ПО, аппаратного обеспечения и КС. В том числе оценивать их производительность. Правда таковое тестирование во многом отличается от тестирования, выполняемого разработчиками.

Три основные стратегии поиска и устранения неисправностей (troubleshooting):

1. Сверху вниз (top-down) -- начинать с прикладного уровня и постепенно «спускаться» на физический.

2. Снизу-вверх (bottom-up) -- начинать с физического уровня и постепенно «подниматься» на прикладной.

3. «Разделяй и властвуй» (divide-and-conquer) -- начинать с наиболее вероятного уровня и «расширяться» в двух направлениях.

Эти стратегии можно применять не только к КС, а к любым информационным системам.

**5 Маршрутизаторы Cisco как специализированное сетевое**

**Оборудование**

Маршрутизаторы Cisco делят на шесть основных целевых категорий (Branch, WAN aggregation, Service provider, Industrial, Virtual, Small business) и множество серий.

Branch (Филиал): Филиалы могут иметь свою собственную локальную сеть и требовать подключения к центральной сети для доступа к общим ресурсам.

WAN Aggregation (Агрегация широкой области сети): Это процесс объединения нескольких соединений широкой области сети (Wide Area Network, WAN) для увеличения пропускной способности и надежности.

Service Provider (Поставщик услуг): В контексте Cisco это может относиться к провайдерам интернет-соединений или другим операторам связи.

Industrial (Промышленный): Относится к сетевым решениям, предназначенным для использования в промышленной среде.

Virtual (Виртуальный): В контексте сетей Cisco, это может относиться к виртуализации сетей или виртуальным сетевым ресурсам, которые могут быть созданы и управляться программно, без привязки к физическим устройствам.

Small Business (Малый бизнес): Это относится к малым предприятиям, которые имеют ограниченные ресурсы и требуют простых, но эффективных сетевых решений.

В названиях многих производившихся и производимых серий фигурируют аббревиатуры.

Маршрутизационные платформы от Cisco можно разделить на менее производительные и более производительные, плюс появившиеся совсем недавно специализированные платформы.

Высокопроизводительные серии позиционируют как основу для наиболее

требовательных к сетевым ресурсам сегментов рынка, а именно: компьютерных систем с большим числом сервисов, инфраструктуры провайдеров, центров обработки данных.

Многие серии представляют собой гибриды с коммутаторами. Примером может служить формально относящаяся к коммутаторам линейка Nexus.

В качестве лабораторной базы для первоначального обучения обычно используют относительно недорогие серии ISRs различных поколений

ПРИВЕСТИ ПРИМЕРЫ

**6 Модули маршрутизаторов Cisco**

Маршрутизаторы Cisco (исключая самые дешевые) изначально разрабатывают как модульные (modular). На так называемое шасси (chassis) с уже установленным базовым набором сетевых интерфейсов существует возможность доустанавливать различные количественно и качественно различающиеся модули.

При этом, в первом приближении, выделяют пять групп модулей:

1) Интерфейсные карты.

2) Интерфейсные модули.

3) Внутренние модули.

4) DSP-сопроцессоры.

5) Порт-адаптеры и другие модули для высокопроизводительных плат-

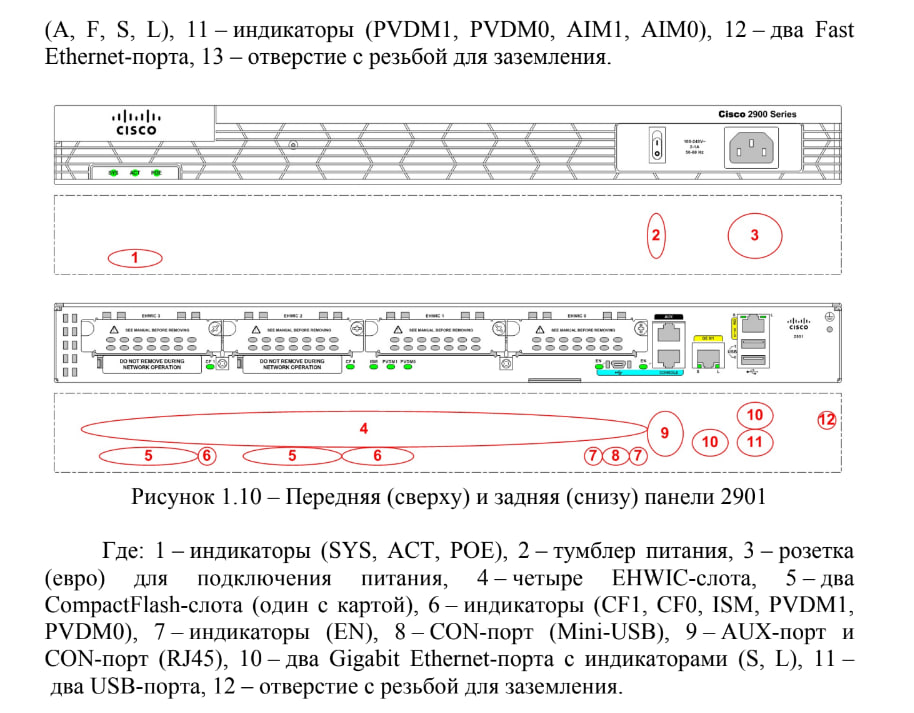
форм.

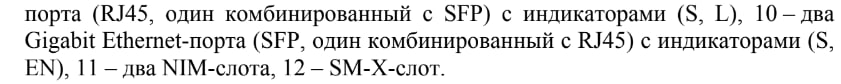
В названии модуля отражено его наполнение (например, HWIC-2FE позволяет добавить два сетевых интерфейса Fast Ethernet).

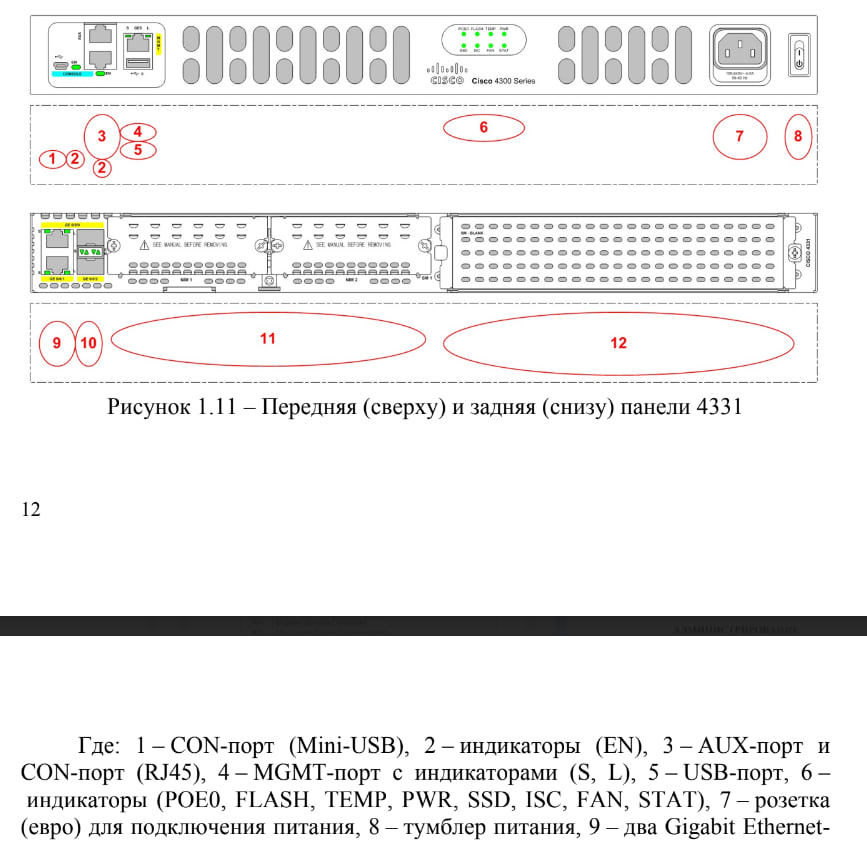
Разрабатывают и полностью программные модули, но маршрутизаторы

(коммутаторы) это не затрагивает.

**7 Учебные маршрутизаторы Cisco**







+ ПЛАНЫ

**8 Структура маршрутизаторов Cisco**

**В этом вопросе можно писать в принципе все что связано с маршрутизаторами – модули, структуру, планы, файловую систему**

Как и в любой микро-ЭВМ, в структуре маршрутизатора произвольной сложности, в конечном счете, можно выделить три «строительных» блока: процессор, память и устройства ввода-вывода.

В маршрутизаторах Cisco используют **процессоры** от ряда производителей с различными архитектурами – в большинстве случаев RISC.

В маршрутизаторах (и коммутаторах) Cisco задействованы четыре традиционные подсистемы **памяти**:

1 BootROM – загрузочное ПЗУ, в котором хранится собственно первичный загрузчик

2 NVRAM (Non-Volatile RAM) – энергонезависимое ОЗУ

в котором хранится загрузочная конфигурация, глобальный конфигурационный регистр.

3 Flash – ПЗУ-накопитель, основным назначением которого является хранение образов ОС IOS

4 DRAM – обычное ОЗУ, в котором «удерживается» исполняющаяся ОС IOS со всеми своими подсистемами, рабочая конфигурация, таблица маршрутизации, буферы пакетов, таблицы адресов

Устройства ввода-вывода, в первую очередь, реализуют различные сетевые интерфейсы. Безусловно, особо следует выделить flash-устройства.

При рассмотрении любого сетевого устройства с точки зрения его функциональной организации можно выделить три так называемых плана (planes):

1 Management – административный – включает весь инструментарий, не-

обходимый администратору для того чтобы он мог управлять сетевым устрой-

ством и отслеживать его состояние (например, протокол SSH).

2 Data – данных – включает все необходимое для выполнения сетевым устройством полезной нагрузки, то есть непосредственной пересылки пользовательского трафика (например, классическую таблицу маршрутизации ).

3 Control – управляющий – представляет собой служебную надстройку

над планом данных, с помощью которой сетевое устройство «разговаривает» с

другими сетевыми устройствами и тем самым адаптирует структуры плана

данных (например, протокол OSPF).

Могут быть выделены и другие планы (например, связанные с виртуализацией).

Маршрутизатор может быть разделен на планы программно и аппаратно (например, планы могут иметь собственные процессоры или, как вариант, отдельные ядра одного процессора).

Аппаратное разделение свойственно высокопроизводительным платформам.

**9 Cisco IOS как встраиваемая операционная система**

Cisco **Internetwork Operating System** (IOS) относят к специализированным

встраиваемым ОС. Основное назначение IOS заключается в предоставлении возможности конфигурирования маршрутизаторов и коммутаторов производства Cisco.

Одна из важных особенностей Cisco IOS - это его модульная архитектура. Она позволяет добавлять и удалять модули функциональности, чтобы адаптировать операционную систему под конкретные требования сети. Это обеспечивает гибкость и масштабируемость при настройке и развертывании сетей Cisco.

Наряду с собственно IOS, существуют еще отдельные линейки для некоторых высокопроизводительных платформ: IOS XE, IOS XR и NX-OS. Сейчас видна тенденция постепенного перехода к использованию IOS XE в качестве основной ОС. IOS XE базируется на ядре Linux (специальный демон IOSd эмулирует интерфейс IOS).

При изучении IOS можно использовать два основных эмулятора: Cisco Packet Tracer и GNS3 .

Cisco имеет собственный интерфейс командной строки – Cisco Command Line Interface (CLI).

Для первоначальной работы с CLI необходимо осуществить физическое подключение консоли, например ноутбука, через RS-232. А также запустить и настроить терминальное ПО, например PuTTY

В настоящее время наиболее актуальны следующие версии IOS для маршрутизаторов:

12.4 – для 2811 и других «современников» (по-прежнему),

15.X – для 2811, 2901 и других.

И IOS XE для маршрутизаторов:

16.X – для 4331 и других

**10 Загрузка IOS**

Последовательность загрузки:

1 После включения питания в первую очередь отрабатывает загрузчик bootstrap, который инициализирует аппаратные структуры загрузочной среды, выполняет POST, инициализирует аппаратные подсистемы, инициализирует программные структуры загрузочной среды.

2 Загрузчик bootstrap пытается найти бинарный образ IOS в загрузочной конфигурации. Если образа не найдено, то bootstrap пытается найти образ в памяти Flash. Если образов несколько, то выбирается первый обнаруженный.

Если образов нет вообще, то загрузчик bootstrap запускает интерпретатор командной строки, который можно использовать для копирования образа в Flash (например, с внешнего TFTP-сервера)

После этого bootstrap загружает его в DRAM и передает ему управление

3 Образ IOS распаковывается и загружается, при этом инициализируя все необходимые программные и аппаратные структуры (необходимое для функционирования)

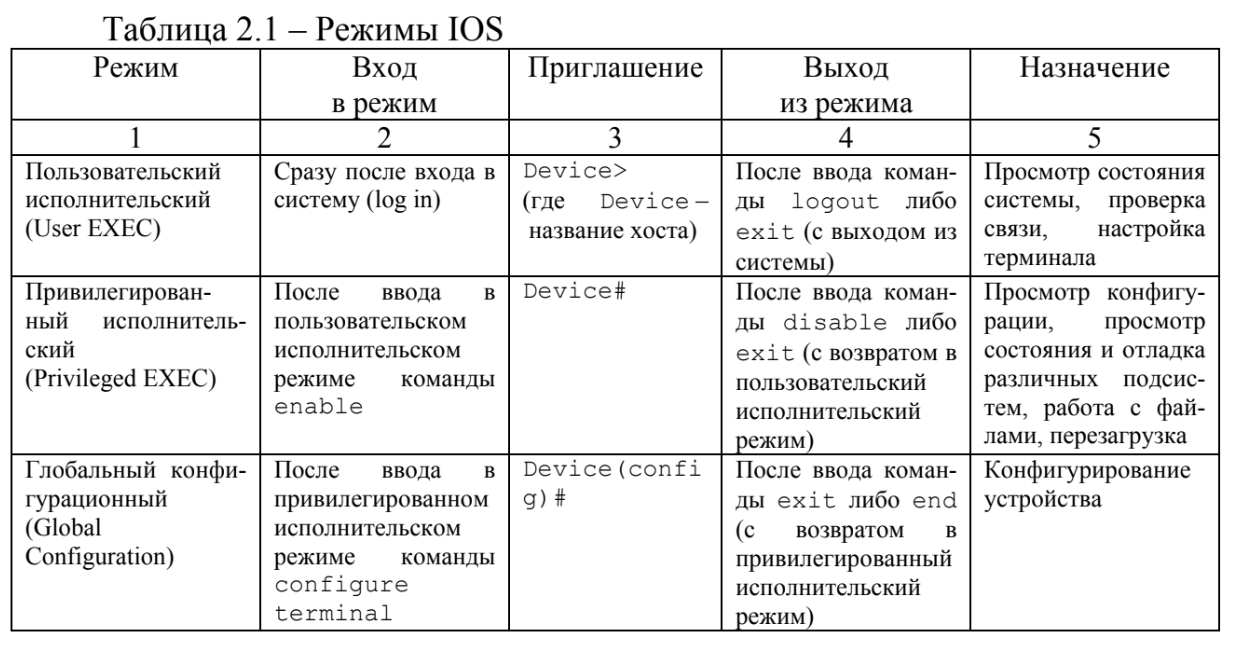
4 Выводится сообщение о нажатии клавиши Enter для начала работы.

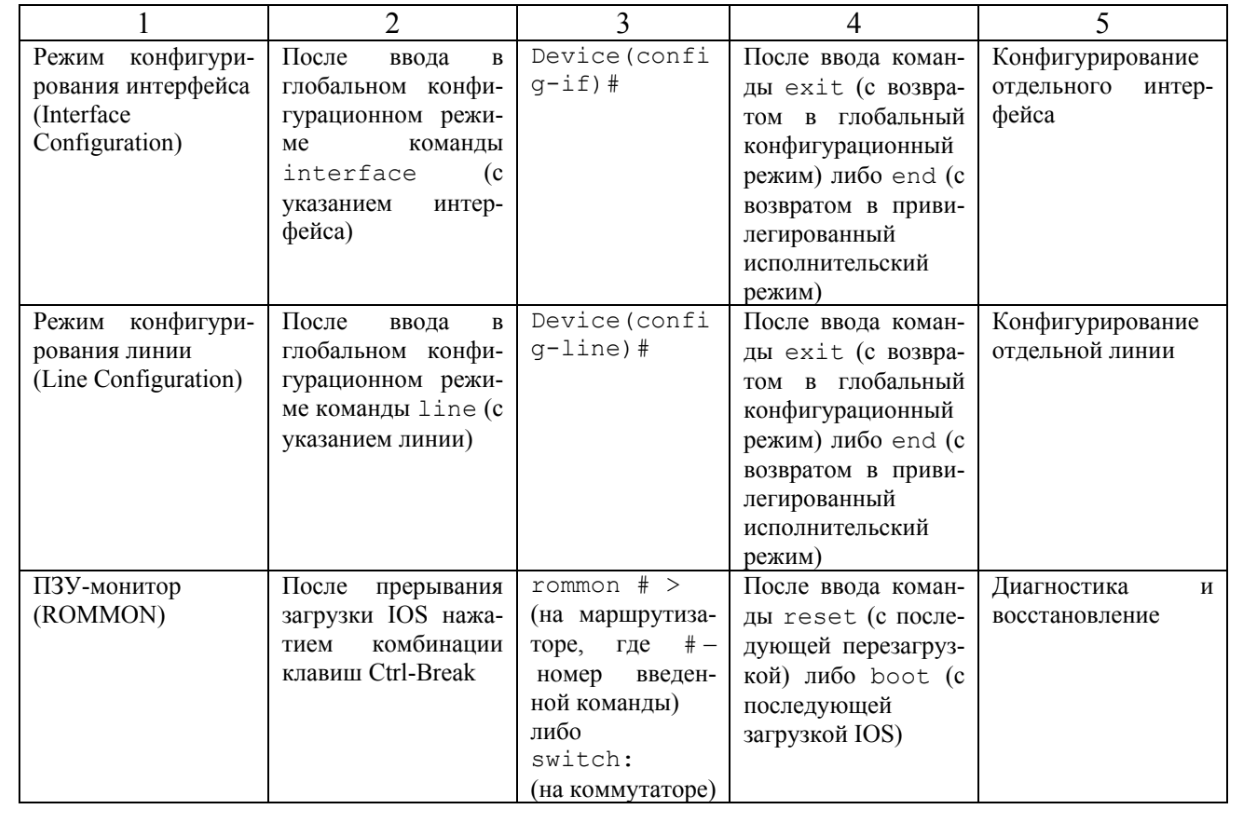
При нажатии клавиши Enter система считается загруженной.

**11 Режимы IOS**

CLI может функционировать в одном из нескольких режимов, отличающихся назначением.

В начале каждого из руководств по IOS напечатана таблица режимов. Ниже приведен адаптированный перевод (таблица 2.1).



****

**12 Система команд IOS и ее особенности**

Команды IOS в большинстве своем комплексные, а значит требуют наличия аргументов при их вводе.

IOS не различает строчные и прописные буквы при вводе команд, но это правило не распространяется на значения некоторых аргументов (например, паролей).

Каждая команда предназначена для определенного режима (режимов), поэтому понимание смысла режимов позволяет легко соотносить с ними команды.

Некоторые команды в разных режимах имеют разные наборы аргументов.

В любом из режимов командой help можно запросить помощь.

С помощью ? можно запросить список всех доступных команд.

Также в CLI заложено несколько вариантов получения подсказок при

вводе команд:

1 Подсказка о неоднозначной команде. (Ambiguous command: "команда")

2 Подсказка о неполной команде. (Incomplete command)

3 Подсказка о неправильной команде. (Invalid input detected at '^' marker.)

Одним из востребованных удобств CLI является возможность сокращения

команд при их вводе.

Префикс “no” позволяет придать «инверсный» смысл некоторой

команде в соответствующей ситуации (shutdown/no shutdown).

Для просмотра состояния различных подсистем IOS используют команду show.

Для программной перезагрузки маршрутизатора либо коммутатора используют команду reload.

**13 Файлы конфигурации и файловая система IOS**

Для обеспечения возможности работы IOS с файловыми ресурсами разработана собственная файловая система Cisco IOS File System (IFS), включающая три подсистемы: network file systems, special file systems, storage file systems.

В основу современных IFSes положены FAT16 (IOS) и ext2 (IOS XE).

Для обращения к локальным или удаленным файловым ресурсам используют специальные префиксы.

Основные команды для работы с файлами:

- cd – сменить каталог;

- copy – скопировать файл либо каталог;

- delete – удалить файл;

- dir – вывести на экран содержимое текущего каталога;

- erase – удалить все файлы и каталоги из файловой системы;

- format – отформатировать файловую систему;

- mkdir – создать каталог;

- more – вывести на экран содержимое файла;

- pwd – вывести на экран название текущего каталога;

- rename – переименовать файл либо каталог;

- rmdir – удалить каталог.

Во время работы необходимо сохранять рабочую конфигурацию. Предварительно

сохраненную на ПК конфигурацию можно вносить и методом «copy-paste».

Конфигурация зависит от версии. Следовательно, файлы конфигурации совместимы «с точностью» до версии.

**14 Основные команды IOS для базовой настройки маршрутизатора и просмотра его состояния**

*При базовой настройке маршрутизатора* используются следующие команды:

enable - переключает режим командной строки в привилегированный

configure terminal (или conf t) - входит в режим конфигурации маршрутизатора

hostname - устанавливает имя хоста для маршрутизатора.

Interface (int) - позволяет настроить параметры интерфейса маршрутизатора

ip route - устанавливает статический маршрут, указывая IP-адрес назначения и следующий хоп

exit - выходит из текущего режима конфигурации

*Для просмотра состояния* различных подсистем IOS используют ком-

плексную команду show.

- show running-config, show startup-config – вывести на эк-

ран конфигурацию;

- show interfaces – вывести на экран подробное состояние всех сете-

вых интерфейсов (без аргументов) либо отдельно взятого интерфейса (если он

указан);

- show line – вывести на экран состояние всех линий (без аргументов)

либо подробное состояние отдельно взятой линии (если она указана);

- show version – вывести на экран общую информацию об IOS и мар-

шрутизаторе либо коммутаторе;

- show processes – вывести на экран подробную информацию о про-

цессах;

- show diag, show platform – вывести на экран подробную инфор-

мацию об оборудовании;

**15 Сетевые интерфейсы в IOS**

Все сетевые интерфейсы (физические и логические, аппаратные и программные, реальные и виртуальные), применительно к которым возможно конфигурирование, Cisco разделяет на два типа: L2 и L3.

Одной из самых важных особенностей оборудования Cisco (даже относи-

тельно дешевого) является возможность преобразования L2 и L3-интерфейсов

друг в друга.

Сетевые интерфейсы коммутаторов по умолчанию являются L2-

интерфейсами и по умолчанию административно включены (administratively

up), а сетевые интерфейсы маршрутизаторов по умолчанию являются L3-

интерфейсами и по умолчанию административно выключены (administratively

down).

Для конфигурирования L2-интерфейсов, точнее, всего что относится ко

второму уровню в L2-интерфейсах, предназначена лишь одна команда, но

очень «развесистая» – switchport.

Как L2-, так и L3-интерфейсы, кроме всего прочего, обладают еще физи-

ческими параметрами, которые конфигурируют отдельным набором команд.

Основные типы: Ethernet, FastEthernet, GigabitEthernet, TenGigabitEthernet, Serial.

Остальные типы Ethernet: TwoGigabitEthernet (2,5 Gb/s), FiveGigabitEthernet, TwentyFiveGigE, FortyGigabitEthernet, HundredGigE

Cisco Loopback – это сугубо программный L3-интерфейс, как правило

используемый для отладки. Создается автоматически при первом «обращении»

(например, interface lo0) и может быть удален. После создания сразу ад-

министративно включается, хотя может быть и административно выключен.

Cisco Null – это так же сугубо программный L3-интерфейс, как правило

используемый для устранения маршрутизационных циклов. Никогда не прини-

мает и не передает пакеты.

Названия интерфейсов при вводе обычно сокращают (по общим прави-

лам), причем можно не вводить и разделяющие пробелы (например, GigabitEthernet 0/0 равно gi0/0).

**16 Сетевые интерфейсы и подсети**

Физически Internet состоит из огромного количества самых разнообразных сегментов. Логическая структуризация Internet заключается в разбиении на подсети. Подсетью (subnet) называют определенное адресное пространство, предполагающее наличие некоторого количества станций. Логическая структура может «накладываться» на физическую по-разному. Но минимальная подсеть должна соответствовать сегменту.

Cisco предлагает три основных критерия объединения станций в подсети:

1. Расположение.

2. Назначение.

3. Принадлежность.

Хотя, в конечном счете, все «завязано» на маршрутизацию.

С точки зрения IP-адресации выделяют два основных типа станций:

1. Пользовательские станции -- User Nodes(UNs) -- за ними работают рядовые пользователи сети.

2. Шлюзовые станции или просто шлюзы -- GateWays(GWs) -- предназначены для объединения подсетей (объединить подсети можно только объединив сегменты).

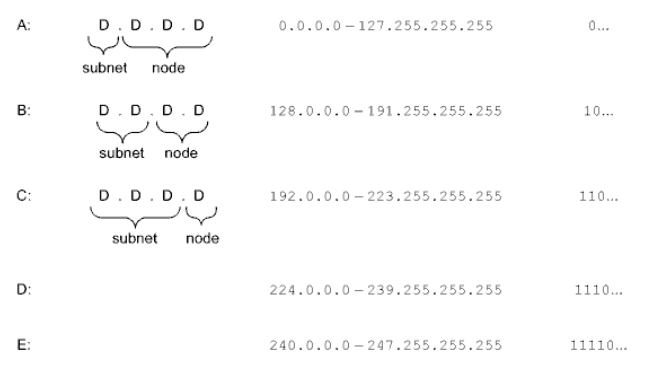
Сетевой интерфейс (network interface) -- это минимально адресуемый в СПД компонент, входящий в состав какой-либо станции. Применительно к компьютерам, как правило, сетевой интерфейс физически выражен в виде сетевого адаптера -- Network Interface Card (NIC).

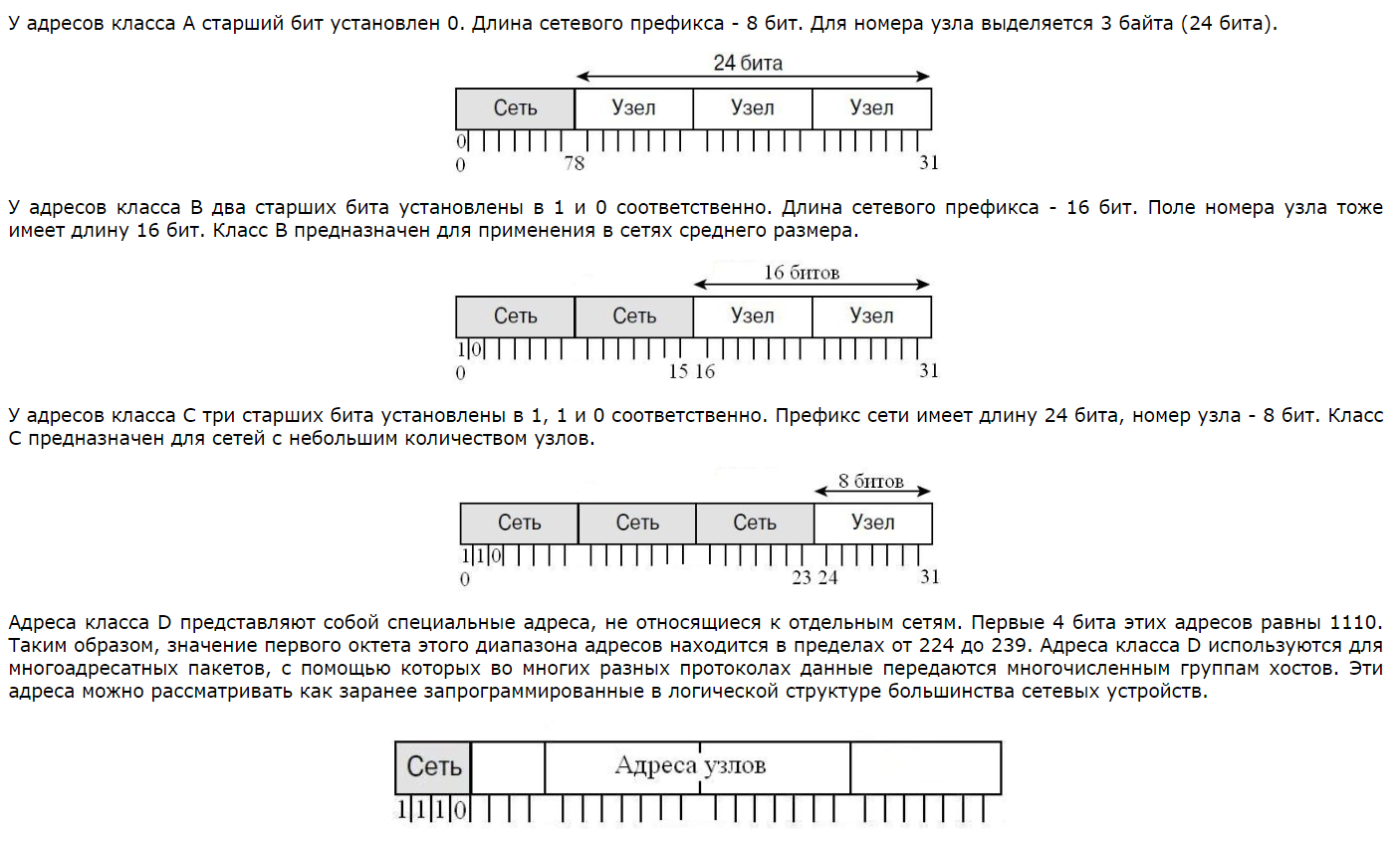
Cтанция может содержать произвольное количество сетевых интерфейсов (пользовательская -- обычно один, шлюзовая -- минимум два). В одном сетевом адаптере обычно содержится один сетевой интерфейс, но может быть интегрировано и несколько. Каждый сетевой интерфейс обычно имеет одну точку подключения к СрПД, то есть физический порт.

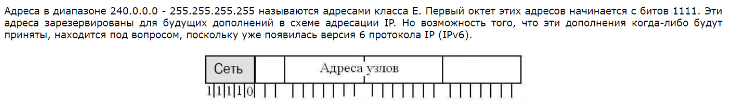
Каждый сетевой интерфейс должен иметь собственный IP-адрес. Если некоторая станция содержит два либо более сетевых интерфейсов, то среди них выделяется главный, ассоциированный с самой станцией. Обычно главный интерфейс «смотрит» в сторону Internet.

**17 Классы IPv4-адресов**

Формально выделяют пять классов IP-адресов. Классы A, B и C являются основными, а классы D и E -- дополнительными. Класс D используется для адресации мультикаст -групп. Класс E зарезервирован для будущего использования. Есть и другие зарезервированные диапазоны.







**18 Явные IPv4-параметры сетевого интерфейса**

Для каждого сетевого интерфейса существует возможность задать 4 так называемых явных IP-параметра:

1. IP Address (IP)

2. Subnet Mask (SM)

3. Default Gateway (DG)

4. DNS Server (DNS)

Собственно, IP-адрес предназначен для адресации некоторой станции посредством соответствующего сетевого интерфейса. Должен быть уникален по крайней мере в пределах подсети. Если станция содержит несколько сетевых интерфейсов, то им нельзя присваивать адреса из перекрывающихся подсетей.

Маска подсети предназначена для выделения подсети исходя из IP-адреса. Маска подсети в двоичном виде представляет собой непрерывную последовательность единиц и следующую за ней непрерывную последовательность нулей согласно общей длине IP-адреса. Принято, что нули соответствуют станционной части, единицы -- подсетевой:

Маски подсетей для стандартных классов:

1) 255.0.0.0 2) 255.255.0.0 3) 255.255.255.0

Маска подсети всегда четная. Маска одинакова для всех сетевых интерфейсов в пределах подсети.

Число адресов в диапазоне подсети всегда равно степени двойки (минимум 4).

Шлюз по умолчанию -- адрес сетевого интерфейса из подсети, на который нужно направлять пакеты, которые предназначены станциям не из текущей подсети (пути к этим станциям неизвестны).

Принято в качестве шлюза по умолчанию назначать адрес первого сетевого интерфейса в подсети и использовать один шлюз по умолчанию. Кроме того, принято в пределах подсети использовать один шлюз по умолчанию

Адрес DNS-сервера необходим для обращения к службе DNS, позволяющей восстановить цифровое значение адреса станции-абонента, с которым работают компьютеры, исходя из символьного, с которым работают люди.

Минимально должны быть известны IP-адрес и маска подсети. Подсеть выделяется из IP-адреса всегда автоматически согласно введенной маске. Если маска подсети не указана, то используется стандартная

**19 Неявные IPv4-параметры сетевого интерфейса**

Определение явных IP-параметров подразумевает задание еще двух неявных:

5. Subnet Address (SA).

6. Broadcast Address (BA).

Адрес подсети используется для «поочередной» адресации всех возможных станций подсети. Адресом подсети является самый нижний адрес из диапазона адресов подсети, и он всегда четный.

Широковещательный адрес используется для одновременной адресации всех возможных станций подсети. Широковещательным адресом является самый верхний адрес из диапазона адресов подсети, и он всегда нечетный.

Более точно таковые широковещательные адреса называют directed broadcasts. Согласно последним рекомендациям RFCs (с целью повышения безопасности), если соответствующая подсеть занимает больше сегмента, то, по умолчанию, пакеты с такими адресами назначения все равно должны «подавляться» на границах сегментов, то есть на шлюзах. Но должна существовать возможность опционального отключения «подавления»

Количество адресов из диапазона подсети, которые можно присвоить сетевым интерфейсам, меньше общего количества адресов на два (минус адрес подсети и широковещательный адрес).

**20 Классификация IPv4-адресов**

С точки зрения «видимости» все IP-адреса делятся на:

1. Реальные (public).

2. Внутренние (private).

В отличие от станции с реальным адресом, cтанция с внутренним адресом «видна» только во внутренней сети предприятия или организации.

В каждом из классов существуют диапазоны адресов, специально зарезервированные для внутренних подсетей.

Диапазоны адресов, зарезервированные для внутренних подсетей:

A: 10.X.X.X

B: 172.16.0.0 - 172.31.255.255

C: 192.168.X.X

С точки зрения временного постоянства все IP-адреса делятся на:

1. Статические

2. Динамические

Статический адрес закрепляется за станцией администратором на более или менее продолжительное время.

Динамический адрес присваивается станции в процессе загрузки по некоторому критерию и действителен только в течение сеанса работы.

Динамический адрес может присваиваться по-разному:

1. Передаваться с сервера по определенному протоколу (например, DHCP) после

выборки из:

-- статического пула

-- динамического пула

2. Случайно генерироваться - адреса Link Local: 169.254.X.X.

Имеется несколько специальных соглашений в области IP-адресации:

1. 0.0.0.0 -- формально адрес всей глобальной сети Internet, но имеет и другие смыслы.

2. 255.255.255.255 -- формально глобальный широковещательный адрес, но поскольку представляет большу'ю «опасность» уже давно интерпретируется как Limited Broadcast, то есть пакеты с такими адресами назначения должны «безоговорочно» подавляться шлюзами.

3. 127.0.0.1 (как и любой адрес из диапазона 127.X.X.X) -- ассоциирован со специальным сетевым интерфейсом-заглушкой (loopback), необходимым для обеспечения переносимости ПО, то есть пакеты с такими адресами назначения, переданные приложениями, тут же программно возвращаются на прикладной уровень.

**21 Использование адресного пространства IPv4 и «правила хорошего**

**тона»**

В настоящее время широко применяется практика последовательного деления адресного пространства. При этом возможны стратегии:

1. Новая подсеть включается в существующую бо'льшую подсеть.
2. Новая подсеть добавляется к существующей как смежная.

Основная разница заключается в маршрутизации. Первая стратегия целесообразна для разноранговых подсетей, вторая -- одноранговых.

С учетом абстракции, типовая оконечная подсеть физически выражена как совокупность станций, подключенных к одной СрПД. В пределах подсети, переданный одной станцией пакет принимается всеми остальными. Чтобы попасть в другие подсети, пакет должен пройти соответствующие шлюзы. В крайнем случае, подсеть может состоять, как только из станций, так и только из шлюзов.

“Правила хорошего тона” подразумевают использование только одного шлюза по умолчанию на одном маршрутизаторе.

Шлюз по умолчанию — [сетевой шлюз](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D1%88%D0%BB%D1%8E%D0%B7), на который пакет отправляется в том случае, если маршрут к сети назначения пакета не известен (не задан явным образом в таблице маршрутизации хоста)[[1]](https://ru.wikipedia.org/wiki/%D0%A8%D0%BB%D1%8E%D0%B7_%D0%BF%D0%BE_%D1%83%D0%BC%D0%BE%D0%BB%D1%87%D0%B0%D0%BD%D0%B8%D1%8E#cite_note-_01d813f825110eac-1). Применяется в сетях с хорошо выраженными центральными [маршрутизаторами](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80), в малых сетях, в клиентских сегментах сетей. Шлюз по умолчанию задаётся записью в таблице маршрутизации вида «сеть 0.0.0.0 с маской сети 0.0.0.0».

**22 Статическая IPv4-адресация в Windows**

Назначение и просмотр статических ipv4 адресов в Windows возможен 3 методами:

1. Через панель управления

- Открыть настройки сети и Интернета.

- Связанные настройки -> Изменить параметры адаптера

- Откроется отдельное окно «Сетевые подключения» панели управления.

- сетевое соединение, для которого нужно установить статический IP-адрес, и выберите параметр Свойства.

- выберите Протокол Интернета версии 4 (TCP/IPv4) на вкладке Сеть и нажмите кнопку Свойства.

- Переключите селектор на «Использовать следующий IP-адрес».

- Теперь введите данные в следующие поля, соответствующие настройкам вашей сети: Ipv4 “адрес” “маска подсети” “Шлюз по умолчанию”

2. Через настройки

- Нажмите значок “Настройки” и выберите вкладку Сеть и Интернет.

- Выберите Wi-Fi> Текущее соединение, т.е. Сеть, к которой вы подключены.

- Прокрутите страницу вниз до раздела настроек IP и нажмите кнопку Изменить.

- Затем выберите параметр Вручную.

- Включите тумблер IPv4.

- Установите статический IP-адрес и маску подсети

3. Через PowerShell

Откройте Powershell от имени администратора и введите следующую команду, чтобы просмотреть текущую конфигурацию сети:

Get-NetIPConfiguration

После этого введите следующую команду, чтобы установить статический IP-адрес, и нажмите Enter.

New-NetIPAddress

-InterfaceIndex 15 -IPAddress 192.168.29.34 -PrefixLength 24 -DefaultGateway

192.168.29.1

Set-DnsClientServerAddress

-InterfaceIndex 4 -ServerAddresses 10.1.2.1

Для просмотра текущих параметров сетевых интерфейсов в Windows используется команда ipconfig.

**23 Статическая IPv4-адресация в Linux**

Обычно, стандартное ядро Linux распознает основные виды сетевых адаптеров. Если такого не происходит, то требуется установка драйвера от производителя, либо «ручная» настройка или перекомпиляция ядра.

В Linux, на примере Ethernet, специальные файлы устройств -- сетевых интерфейсов -- это eth0, eth1 и так далее согласно их количеству.

IP-параметры каждого сетевого интерфейса хранятся в соответствующем файле в каталоге /etc/sysconfig/network-scripts (ветви Red Hat и SuSE) либо в файле /etc/network/interfaces (ветвь Debian).

Еще один важный файл -- это /etc/sysconfig/network.

Список DNS-серверов хранится в файле - /etc/resolv.conf.

Для просмотра текущих параметров сетевых интерфейсов в Linux -- ifconfig (позволяет менять параметры «на лету», но изменения хранятся до ближайшей перезагрузки).

Для проверки связи, как в Windows, так и в Linux, применяется команда ping.

Для отслеживания пакетов в Linux широко применяется команда tcpdump.

В Linux статическая IPv4-адресация может быть настроена с использованием утилиты ifconfig или команды ip.

Вот примеры настройки статического IPv4-адреса в Linux:

С использованием ifconfig:

sudo ifconfig eth0 192.168.1.100 netmask 255.255.255.0

В этом примере eth0 - имя сетевого интерфейса, 192.168.1.100 - желаемый IPv4-адрес, 255.255.255.0 - маска подсети.

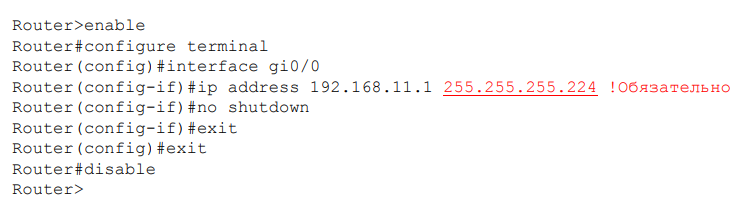
С использованием команды ip:

sudo ip addr add 192.168.1.100/24 dev eth0

Здесь eth0 - имя сетевого интерфейса, 192.168.1.100/24 - IPv4-адрес и маска подсети.

**24 Статическая IPv4-адресация в IOS**

Для назначения IP-адреса сетевому интерфейсу используют команду ip address. IOS поддерживает подинтерфейсы, но на уровне сетевого интерфейса может быть только один IP-адрес. При попытке ввода второго IP-адреса первый вытесняется. Для административного включения сетевого интерфейса используют команду no shutdown, для выключения -- соответственно shutdown.

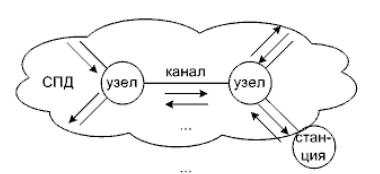
****

Для вывода на экран IP-информации о сетевом интерфейсе либо сетевых интерфейсах используют команду show ip interface.

Для указания адреса DNS-сервера используют команду ip name-server. Для запрещения обращений к DNS-серверу используют команду no ip domain lookup

Для проверки связи используют команды ping и traceroute. Эти команды в СПД с маршрутизаторами Cisco начинают «срабатывать» постепенно. Если команду ping либо команду traceroute ввести без аргументов, то ее можно «настроить» перед запуском.

**25 Структура сети передачи данных**

****

Канал представляет собой СрПД, через которую передаются пакеты. Узел представляет собой некоторое устройство, выполняющее прием, передачу или ретрансляцию пакетов. Узлами и станциями могут быть самые разные устройства.

Все узлы делят на два типа:

1. Пассивные

2. Активные

Пассивность узла означает, что он не выполняет анализ или обработку пакетов. Активность подразумевает, что пакеты анализируются или обрабатываются

Если соотносить узлы с моделью OSI, то можно выделить:

1. Повторители (repeaters) -- аппаратно «сращивают» СПД на физическом уровне, типичными представителями являются оконечные концентраторы (hubs) (уже не производят).

2. Мосты (bridges) -- аппаратно (но есть и «интеллектуальные») «сращивают» СПД на канальном уровне, типичными современными представителями являются коммутаторы (switches).

3. Шлюзы (gateways) -- аппаратно и программно «сращивают» СПД на сетевом уровне, типичными представителями являются маршрутизаторы (routers).

Функция маршрутизации выполняется собственно маршрутизаторами. Но, нужно учитывать, что и все оконечные устройства должны иметь подсистему маршрутизации.

Физические порты маршрутизаторов ограничивают широковещательные домены. Физические порты коммутаторов ограничивают домены коллизий.

Оконечные концентраторы входят в домены коллизий и широковещательные домены.

**26 Понятие маршрута и классификация маршрутов***Маршрут* -- это путь, по которому пакет передается от станции-отправителя к станции-получателю или составная часть этого пути.

Выделяются три вида маршрутов:

1. Маршрут к станции (сетевого интерфейса).

2. Маршрут к подсети.

3. Маршрут по умолчанию.  
Каждый маршрутизатор принимает решения о направлении пересылки пакетов на основании таблицы маршрутизации. Таблица маршрутизации содержит набор правил. Каждое правило в наборе описывает шлюз или интерфейс, используемый маршрутизатором для доступа к определенной сети.

В IP-сетях реализованы два типа маршрутизации:

1. *Статическая*

2. *Динамическая*

При статической маршрутизации таблицы формируются «вручную» или автоматически на основе указанных IP-параметров и хранятся до их «ручной» модификации. При динамической маршрутизации таблицы и формируются, и модифицируются автоматически с задействованием специальных служебных протоколов, что не отменяет возможность вмешательства администратора.

В отношении протоколов динамической маршрутизации, все сетевые интерфейсы делятся на:

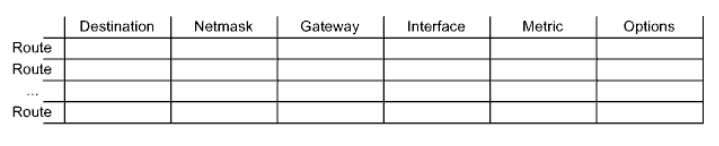
1. *Активные* -- могут использоваться при обмене маршрутной информацией.

2. *Пассивные* -- не могут использоваться при обмене маршрутной информацией.

В некоторых реализациях аналогично делятся маршруты. *Пассивные* маршруты, в отличие от *активных*, не могут «затрагиваться» (считываться и замещаться) протоколами динамической маршрутизации.

В некоторых реализациях (например, Windows) особо выделяются ***персистентные*** маршруты, которые должны сохраняться после перезагрузки.

**27 Обобщенная структура таблицы маршрутизации**Маршруты хранятся в специальной таблице, называемой *таблицей маршрутизации*. В обобщенном виде, с теми или иными вариациями, таблицу можно представить следующим образом.



Назначение полей:

1. Destination -- адрес назначения.

2. Netmask -- маска подсети -- дополняет адрес назначения с целью его правильной интерпретации.

3. Gateway -- шлюз -- IP-адрес шлюза-соседа, которому нужно передать пакет.

4. Interface -- интерфейс -- IP-адрес или другой параметр, однозначно определяющий сетевой интерфейс, который должен физически «выдать» пакет в канал.

5. Metric -- метрика -- определяет приоритетность маршрута (основное назначение), часто рассматривают в совокупности с так называемой административной дистанцией.

6. Options -- опции -- специфические опции данной реализации.

Специальные соглашения в области IP-маршрутизации:

1. Адрес назначения 0.0.0.0 -- маршрут по умолчанию.

2. Mаска подсети 255.255.255.255 -- маршрут к одному сетевому интерфейсу.

**28 Алгоритм применения таблицы маршрутизации для передачи пакета**Таблица маршрутизации определяет, что делать с уже принятым пакетом, подлежащим ретрансляции, или имеющимся пакетом, сформированным для передачи на вышестоящих уровнях. При наличии такого пакета, работа с таблицей маршрутизации протекает в две фазы:

1. Поиск маршрутной информации.

2. Применение маршрутной информации.

В настоящее время, как де-факто стандартный, применяется подход согласно принципу *наиболее точного соответствия* (best match, longest match), заключающийся в следующем:

1. Маршрут ищется путем последовательного сравнения IP-адреса назначения с диапазонами, считываемыми из строк таблицы маршрутизации.

2. При попадании (hit) маршрут считается подходящим.

3. Просматривается вся таблица маршрутизации. Конечно, этот процесс разными способами оптимизируется.

4. При наличии нескольких попаданий выбирается наиболее точный маршрут. Точность попадания определяется «размером мишени». Самым точным является маршрут к станции.

5. При одинаковой точности попадания маршрут выбирается исходя из дополнительного критерия -- метрики.

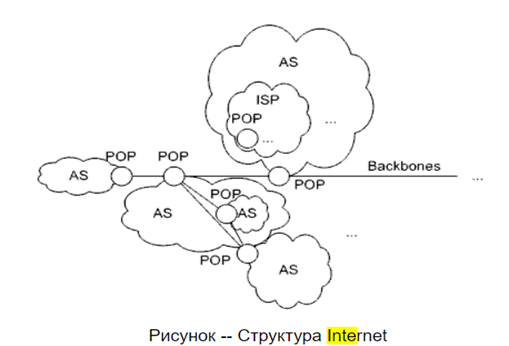
6. Маршрут по умолчанию выбирается если не найдено ни одного более точного маршрута. «Промахнуться» невозможно.

7. При отсутствии попаданий пакет уничтожается (drop).

8. Маршрут ищется для того, чтобы его применить. Применение маршрута заключается в отправке по нему пакета. Пакет передается один раз.

9. На вопросы о том, куда и чем передавать, отвечают соответствующие поля в маршруте.

При наличии нескольких альтернативных маршрутов могут совпасть и их метрики, то есть маршруты оказываются абсолютно равноправными (надо отметить, что такое происходит довольно часто). В некоторых реализациях это считается недопустимым, а в некоторых возникает так называемая балансировка нагрузки, точнее, *эквивалентная балансировка нагрузки* -- соответствующие пакеты поочередно передаются в разных направлениях. Существует еще и *неэквивалентная балансировка нагрузки* -- отличается тем, что трафик распределяется пропорционально согласно метрикам.

**29 Структура Internet**  


Основными структурными единицами Internet являются *автономные системы* - (AS). Каждая AS выделяется исходя из наличия собственной системы маршрутизации (возможно оригинальной), то есть состояние AS не должно зависеть от состояния других ASes.

Все ASes имеют уникальные 16-битные номера. Номера ASes поделены на:

1. Public: 1 -- 64511.

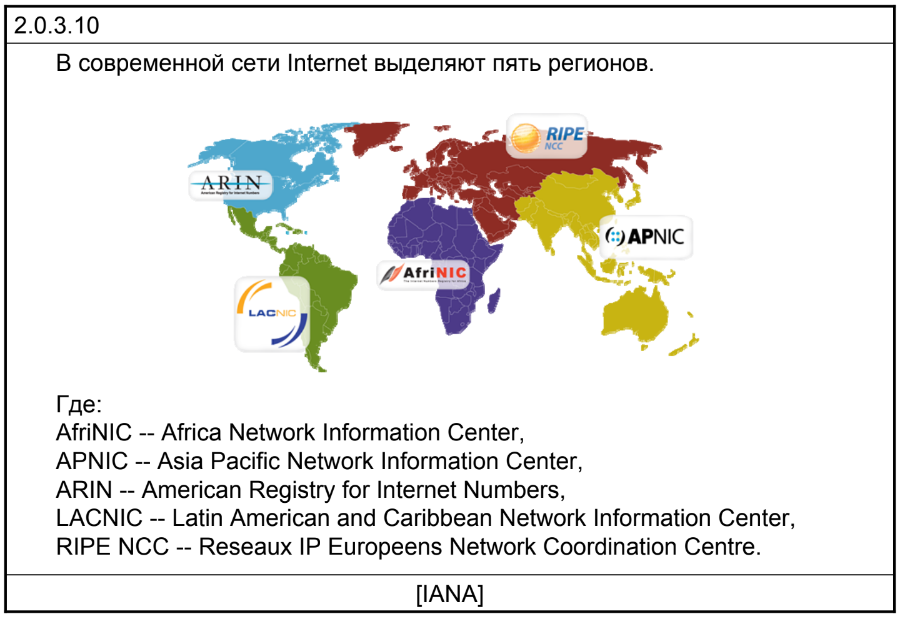
2. Private: 64512 -- 65535.

В связи с исчерпанием, не так давно были введены дополнительные 32-битные номера.

ASes связаны между собой посредством *базовых магистралей (backbones)*. Изначально в структуре Internet была задумана и реализована одна базовая магистраль, но сейчас это лишь условность. Поскольку на практике далеко не всегда удавалось осуществить непосредственное примыкание той или иной AS к базовой магистрали, к настоящему времени возникла очень сильная фрагментация. Реально, ASes соединены друг с другом через так называемые *пиринговые точки* или, по-другому, *точки присутствия* -- Points-Of-Presence (POPs).

Внутри ASes работают *провайдеры* -- Internet Service Providers (ISPs). Касательно POPs, следует уточнить, что терминологически это, в первую очередь, точки предоставления коммуникационных услуг пользователям Internet.

Также следует отметить, что крупные телекоммуникационные компании могут обладать несколькими ASes, а их СПД могут иметь межконтинентальную протяженность.



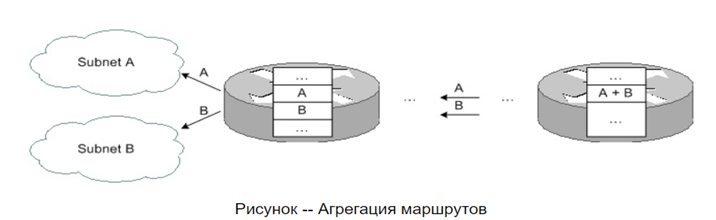
КАРТИНКА ИЗ ЛЕКЦИИ

**30 Назначение и классификация протоколов динамической маршрутизации**

Суть всех протоколов динамической маршрутизации заключается в реализации тех или иных алгоритмов обмена маршрутами к подсетям, с целями как оптимизации трафика, так и вообще нахождения абонентов.

Обмен происходит именно маршрутами к подсетям. Основной смысл разбиения на подсети состоит в упрощении таблиц маршрутизации. Вместо того чтобы отслеживать станции и направлять пакет каждой из них «персонально», пакет направляется сразу в подсеть.

Также упрощение достигается за счет *агрегации маршрутов* -- получение более общего маршрута из отдельных маршрутов к нескольким подсетям, если направления к этим подсетям совпадают. Реально агрегация происходит путем суммирования маршрутов.



Суммирование может быть:

1. «Ручным» (manual) -- выполняется администратором

2. Автоматическим (auto) -- выполняется реализацией протокола

динамической маршрутизации

Как на уровне базовых магистралей, так и в пределах AS, допускается одновременное применение нескольких протоколов динамической маршрутизации. Шлюзы в пределах ASes называют внутренними, а шлюзы, через которые ASes подключены к базовым магистралям -- внешними. Соответственно, протоколы для внутренних шлюзов называют IGPs (Interior Gateway Protocols), а для внешних -- EGPs (Exterior Gateway Protocols).

Почти все используемые в IP-сетях протоколы динамической маршрутизации относят к группе адаптивных двух типов:

1. Distance Vector Algorithms -- алгоритмы, основанные на анализе векторов расстояний.

2. Link State Algorithms -- алгоритмы, основанные на анализе состояния связей.

DVAs при выборе маршрутов оценивают расстояние до подсетей. Касательно пересылки пакетов, расстояние в КС принято измерять в хопах. Один *хоп* (hop) -- это изначальная передача либо одна последующая ретрансляция пакета.

LSAs при выборе маршрутов оценивают состояние связей, то есть каналов. Классическим примером состояния канала является его пропускная способность.

///////////////////

В отношении протоколов динамической маршрутизации, все сетевые интерфейсы делят на:

1. Активные (могут использоваться при обмене маршрутной информацией).

2. Пассивные (не могут использоваться при обмене маршрутной информацией).



**31 Последовательность действий при передаче пакета в подсети и пересылка транзитных пакетов**

Последовательность действий при передаче пакета в некоторой подсети заключается в следующем:

1. Пакет с известным IP-адресом назначения в заголовке передается на уровень MAC (например, Ethernet) и выполняется инкапсуляция.

2. В нормальной ситуации ядро сетевой ОС хранит таблицу соответствия MAC и IP-адресов. Если MAC-адрес назначения станции-абонента либо шлюза не известен, то для его восстановления используется протокол ARP.

3. Если пакет (теперь уже кадр) предназначен станции из текущей подсети, то, после передачи сетевым интерфейсом станции-передатчика, он будет сразу принят всеми станциями подсети.

4. Причем только на станции-абоненте, на основании анализа MAC-адреса назначения, кадр будет распознан как свой и его содержимое будет передано на уровень IP для дальнейшей обработки. Остальными станциями кадр будет отброшен.

5. Если пакет предназначен станции из другой подсети, то он будет передан, согласно таблице маршрутизации, соответствующему шлюзу с использованием MAC-адреса этого шлюза.

Если по каким-либо причинам необходимо принимать и обрабатывать все кадры, то включается специальный режим работы сетевого интерфейса – promiscuous.

Для того чтобы обеспечить передачу транзитных пакетов между подсетями через шлюз на нем должен быть разрешен IP Forwarding.

После включения IP Forwarding, каждый пакет, принятый одним из сетевых интерфейсов, может быть ретранслирован другими, то есть станция работает собственно, как шлюзовая.

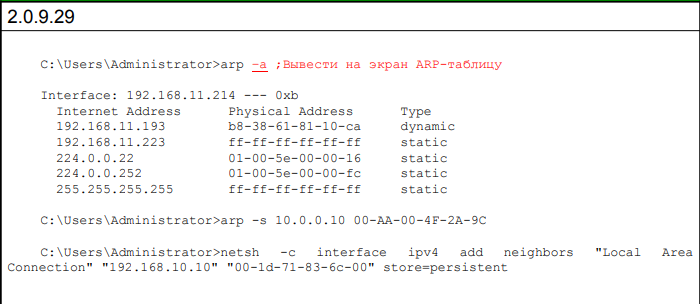
**32 Структура таблицы ARP**

Для просмотра ARP-таблицы используют команду show ip arp. Строки ARP-таблицы могут быть:

1. Статическими -- вносятся администратором и, как правило, хранятся до перезагрузки или «ручного» удаления.

2. Динамическими -- вносятся ОС автоматически и, как правило, удаляются по таймеру.

Строки с постоянными соответствиями сохраняются после перезагрузки.



Структура таблицы ARP крайне простая. В первой колонке указывается IP адрес устройства, во второй соответствующий ему MAC адрес. В третей колонке указывается тип строки static/dynamic.

**33 Использование протокола ARP**

ARP — протокол разрешения адресов (Address Resolution Protocol) является протоколом третьего (сетевого) уровня модели OSI, используется для преобразования IP-адресов в MAC-адреса, играет важную функцию в множественном доступе сетей.

Непосредственно связь между IP адресом и MAC адресом осуществляется с помощью так называемых ARP-таблиц, где в каждой строке указывается соответствие IP адреса MAC адресу.

ARP-сообщения инкапсулируются в Ethernet-кадры. Существуют следующие типы сообщений ARP: запрос и ответ.

Request - широковещательный (адресован всем станциям в домене). Суть запроса: «компьютер с IP-адресом …, сообщите свой MAC-адрес компьютеру с МАС-адресом …

В дополнение к запросу и ответу, предусмотрен еще один вид ARP-сообщений -- ARP probe. Что позволяет, например, при загрузке ОС, обнаружить конфликты IP-адресов и параллельно оповестить все станции в подсети о «возникновении» у сетевого интерфейса нового IP-адреса. «Исчезновение» IP-адреса не анонсируется.

ARP proxy в связке с directed broadcast forwarding позволяет организовать прозрачный шлюз. Включение ARP proxy разрешает шлюзу отвечать на ARP-запрос из одной своей подсети в отношении IP-адреса из другой своей подсети (подставлять свой MAC-адрес). Такой запрос может возникнуть только если запрашивающая станция считает, что запрашиваемая станция находится в той же подсети.

ARP proxy при правильном конфигурировании никогда не требуется

В Windows вообще нет способов включения ARP proxy.

**34 Практические особенности IPv4-маршрутизации**

Таблица маршрутизации определяет что делать с уже принятым пакетом, подлежащим ретрансляции, или имеющимся пакетом, сформированным для передачи на вышестоящих уровнях. При наличии такого пакета, работа с таблицей маршрутизации протекает в две фазы:

1. Поиск маршрутной информации.

2. Применение маршрутной информации

В настоящее время, как де факто стандартный, применяется подход согласно принципу наиболее точного соответствия (best match, longest match), заключающийся в следующем:

1. Маршрут ищется путем последовательного сравнения IP-адреса назначения, считанного из заголовка пакета, с диапазонами, задаваемыми адресами назначения в связке с масками подсетей, считываемыми из строк таблицы маршрутизации.

2. При попадании (hit) маршрут считается подходящим.

3. Просматривается вся таблица маршрутизации. Конечно, этот процесс разными способами оптимизируется.

4. При наличии нескольких попаданий выбирается наиболее точный маршрут. Точность попадания определяется «размером мишени». Самым точным является маршрут к станции.

5. При одинаковой точности попадания маршрут выбирается исходя из дополнительного критерия -- метрики.

6. Маршрут по умолчанию выбирается если не найдено ни одного более точного маршрута. «Промахнуться» невозможно.

7. При отсутствии попаданий пакет уничтожается (drop).

8. Маршрут ищется для того, чтобы его применить. Применение маршрута заключается в отправке по нему пакета. Пакет передается один раз.

9. На вопросы о том, куда и чем передавать, отвечают соответствующие поля в маршруте

При наличии нескольких альтернативных маршрутов могут совпасть и их метрики, то есть маршруты оказываются абсолютно равноправными (надо отметить, что такое происходит довольно часто). В некоторых реализациях это считается недопустимым, а в некоторых возникает так называемая балансировка нагрузки, точнее, эквивалентная балансировка нагрузки (equal load balancing) -- соответствующие пакеты поочередно передаются в разных направлениях. Существует еще и неэквивалентная балансировка нагрузки (unequal load balancing) -- отличается тем, что трафик распределяется пропорционально согласно метрикам.

В IP-сетях реализованы два типа маршрутизации:

1. Статическая (static).

2. Динамическая (dynamic).

При статической маршрутизации таблицы формируются «вручную» или автоматически на основе указанных IP-параметров и хранятся до их «ручной» модификации. При динамической маршрутизации таблицы и формируются, и модифицируются автоматически с задействованием специальных служебных протоколов, что не отменяет возможность вмешательства администратора.

В отношении протоколов динамической маршрутизации, все сетевые интерфейсы делят на:

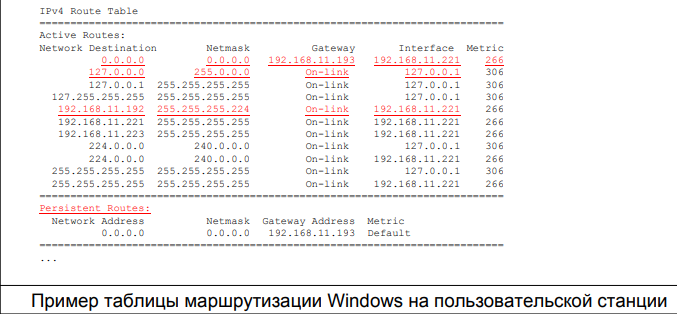
1. Активные (active) -- могут использоваться при обмене маршрутной информацией.

2. Пассивные (passive) -- не могут использоваться при обмене маршрутной информацией.

В некоторых реализациях (например, UNIX routed) аналогично делят маршруты. Пассивные маршруты, в отличие от активных, не могут быть «затронуты» (считаны или замещены) протоколами динамической маршрутизации. В реализациях особо выделяют постоянные или, по -другому, персистентные (persistent) маршруты, которые должны сохраняться после перезагрузки.

**35 Структура таблицы IPv4-маршрутизации в Windows**

Чтобы просмотреть текущую таблицу маршрутизации в Windows используют команду route с аргументом print. В первой колонке таблицы маршрутизации Windows указывается адрес подсети назначения, во второй - её маска, в третьей - шлюз, через который достижима эта подсеть, четвёртый - интерфейс, который должен физически «выдать» пакет в канал, пятый - метрика.



Постоянные маршруты: настраиваемые вручную маршруты статического лечени

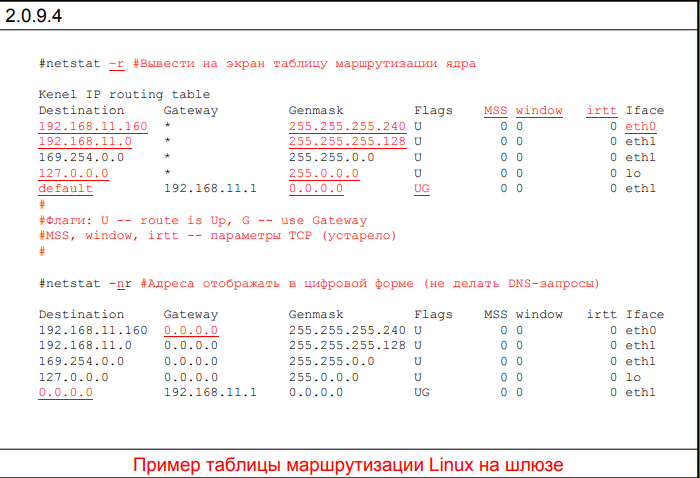
/////////////////

3) Шлюз: При отправке пакетов данных IP шлюз определяет сервер следующего перехода, на который отправляются пакеты данных для определенного сетевого адреса назначения.

4) Интерфейс:. Интерфейс определяет конкретный сетевой адрес назначения, сетевой интерфейс, используемый локальным компьютером для отправки пакетов данных.

5) Метрика: количество переходов, счетчик переходов используется для обозначения стоимости маршрутизации, обычно представляет собой количество переходов, которые необходимо пройти, чтобы достичь адреса назначения, а счетчик переходов представляет маршрутизатор. Чем меньше количество переходов, тем ниже стоимость маршрутизации и выше приоритет.

**36 Структура таблицы IPv4-маршрутизации в Linux**



Традиционная команда для просмотра таблицы маршрутизации в Linux: netstat -r (-nr). Также можно воспользоваться командой route. Сама таблица состоит из 8 столбцов:

* Destination - адрес подсети назначения,
* Gateway - шлюз,
* Genmask - маска подсети назначения,
* Flags - флаг, определяющий характеристики маршрута,
* Iface - название интерфейса.
* MSS - размер пакета
* Window - TCP-окно
* irtt - время отклика для TCP-соединений
* Всё остальное - устаревшие параметры TCP

Обратить внимание на картинке:

Флаги:

U -- route is Up,

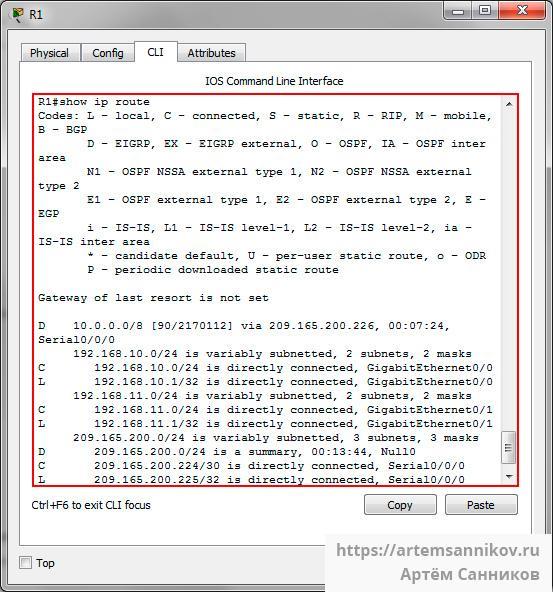
G -- use Gateway

MSS,

window, -- параметры TCP (устарело)

irtt

**37 Структура таблицы IPv4-маршрутизации в IOS**



Просмотреть содержимое таблицы маршрутизации в Cisco IOS можно с помощью команды show ip route.

Вначале вывода команды мы видим легенду, где показано, какая буква что обозначает.

Ниже легенды идёт список известных устройству маршрутов.

Первоначально в строке таблицы пишется буква, которая обозначает тип маршрута.

Затем идёт адрес подсети, в которую нужно попасть и адрес сетевого интерфейса, который может обеспечить достижимость подсети.

Дополнительно после адреса желаемой подсети в квадратных скобках может указываться административная дистанция и метрика или доп. информация (”is directly connected”).

В конце дополнительно может указываться название интерфейса.

В первую очередь маршруты делят на:

1. Directly connected равно Connected (код C) -- маршруты к своим подсетям

2. Static (код S) -- статические (маршруты, которые вносят «вручную»).

3. Dynamic (коды R, B, O и другие) -- динамические (автоматическивносятся процессами динамической маршрутизации).

Новшество таблиц маршрутизации IOS 15.X (а также таблиц маршрутизации IPv6):

+4. Local (код L) -- локальные или, в данном контексте, маршруты к своим сетевым интерфейсам

В иерархии маршрутов выделяют два уровня:

1. L1 -- маршруты к стандартным подсетям и подсетям, большим чем стандартные.

2. L2 -- маршруты к подсетям, меньшим чем стандартные, и к сетевым интерфейсам.

С другой стороны, маршруты в иерархии можно рассматривать как:

1. Parent -- родительские.

2. Child -- дочерние.

Иерархия необходима для ускорения обработки таблицы маршрутизации. Сначала просматриваются маршруты первого уровня. В случае попадания происходит переход к просмотру соответствующих маршрутов второго уровня.

**38 Статическая IPv4-маршрутизация в Windows, Linux и IOS**

Чтобы добавить статический маршрут в таблицу маршрутизации ядра, и в Windows, и в Linux, используют команду route с аргументом add.

Удалить в Windows: route delete.

Удалить в Linux: route del.



Постоянство вводимого статического маршрута в Windows достигают за счет аргумента -p.

Постоянство статических маршрутов в Linux обеспечивают несколькими способами с возможностью комбинирования этих способов.

Маршруты могут «привязываться» к конкретным сетевым интерфейсам (но необязательно использовать их).

При необходимости введения сравнительно большого количества статических маршрутов или при переходе к простейшей динамической маршрутизации (RIP) в Linux можно задействовать демон routed (Де́мон — компьютерная программа в UNIX-подобных системах, запускаемая самой системой и работающая в фоновом режиме без прямого взаимодействия с пользователем. Демоны обычно запускаются во время загрузки системы.). При этом **статические маршруты помещают в стандартный конфигурационный файл /etc/gateways.**

После настройки и запуска сервиса в течение некоторого времени сформируется таблица маршрутизации, которая затем может изменяться.

IOS поддерживает эквивалентную и неэквивалентную балансировку нагрузки при маршрутизации (и в отношении маршрутов по умолчанию).

Для внесения статического маршрута в таблицу маршрутизации IOS используют команду **ip route.**

*Router(config)#ip route 192.168.11.160 255.255.255.240 192.168.11.50*

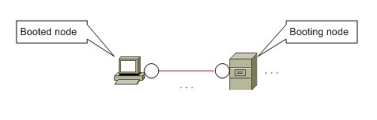
Отключение бесклассового выбора маршрутов, то есть включение полноклассового, осуществляют командой **no ip classless.**

Функционал IP forwarding по умолчанию включен и может быть выключен командой **no ip routing.**

Функционал ICMP redirects по умолчанию включен и может быть выключен командой **no ip redirects.**

**39 Структура системы удаленной загрузки**

Термин удаленная загрузка означает, что по крайней мере ядро ОС некоторой станции загружается не с локальных накопителей, а по сети -- с удаленной станции. Удаленная загрузка как правило используется для бездисковых пользовательских станций, не предназначенных для хранения информации. Каждый раз загружается «заготовка» ОС. Таким образом, в состав сети с удаленной загрузкой входит как минимум две станции, которые обычно расположены в одном сегменте.



Для решения проблемы эмуляции системного диска используют два подхода:

1. Поддержка виртуального диска в памяти (RAM Drive).

2. Поддержка сетевого виртуального диска.

Плюсы удаленной загрузки:  
  
+ Централизованное управление (работа с большим кол-вом компьютеров не по-одиночке)

+ Экономия времени и средств  
+ Безопасность (загрузка происходит с проверенного источника)

**40 Технологии удаленной загрузки**

В настоящее время существуют несколько семейств технологий, связанных с удаленной загрузкой (используется клиент-серверная модель, включая поддержку со стороны BIOS/UEFI и загрузчиков Linux, в первую очередь выражены в соответствующих протоколах):

1. Для IPX: RPL плюс ПО от Novell, Microsoft и другое.

2. Для IPv4: **BOOTP -> DHCP -> PXE** плюс ПО от 3COM, Intel, Citrix, Microsoft, IBM, НР и другое.

3. Для IPv6: **DHCPv6 -> Netboot6** (PXE на базе IPv6) плюс ПО от Citrix и Microsoft.

4. Для IPv4/IPv6: iSCSI (internet SCSI) Boot и FCoE (Fibre Channel Over Ethernet) Boot и HTTP Boot плюс ПО от Cisco, Microsoft, Intel, IBM и другое.

5. Для IPv4/IPv6: HTTP Boot плюс ПО от HPE (для некоторых серверов),

IBM (для некоторых серверов).

6. Для IPv4/IPv6: Прочие протоколы плюс как правило свободно

распространяемое ПО, например, *gPXE -> iPXE* (развитие EtherBoot)

(альтернатива PXE, но поддерживает PXE плюс другие протоколы).

*Я бы выделил тут те протоколы которые вынесены на отдельные вопросы и описал бы как они работают + для примера ipv4/ipv6 HTTP Boot*

**41 Поддержка удаленной загрузки в BIOS**

BIOS работает в реальном режиме с 16-ти разрядной адресацией и имеет совсем немного реализаций с разными модификациями и «обертками». В BIOS, еще при изначальной разработке, была заложена возможность включать сторонние дополнения -- add-on BIOSes. Для обеспечения удаленной загрузки на стороне клиентской станции в состав add-on BIOSes необходимо включить boot ROM -- специальное загрузочное ПЗУ.

Касательно IP, boot ROMs ориентированы только на IPv4, IPv6 не поддерживают.

После включения загружаемой станции выполняется так называемый POST (Power On Self Test).

При этом BIOS сканирует память в диапазоне C0000h -- EE000h (куда отображаются add-on BIOSes) с инкрементом, равным 2 kByte, в поисках сигнатуры 55AAh, которая свидетельствует о наличии add-on BIOS.

Если сигнатура найдена, то третий байт, содержащий размер add-on BIOS в 512-ти байтовых страницах, используется для проверки контрольной суммы.

Если контрольная сумма равна нулю, то осуществляется вызов подпрограммы по адресу, расположенному со смещением +3 (четвертый байт).

В случае с boot ROM, вызванный код используется для подмены обработчика прерывания 18h (ROM BASIC).

После просмотра всего диапазона, BIOS выполняет инструкцию INT 18h (перезагрузка).

Затем, вернув управление, новый обработчик копирует основное содержимое boot ROM (loader) в оперативную память и передает ему управление.

Затем, загрузчик loader с помощью подпрограмм boot ROM загружает простейший сетевой протокол, получает код загрузчика bootstrap от загружающей станции и передает ему управление.

Дальнейшие действия зависят от реализации.

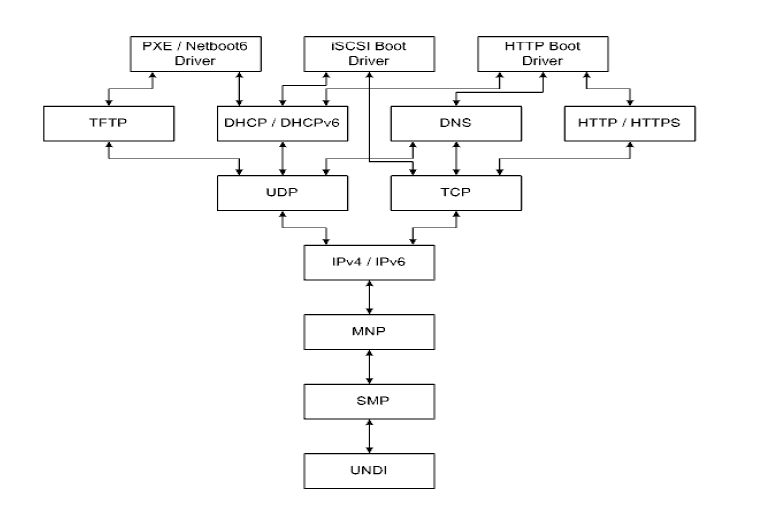
**42 Поддержка удаленной загрузки в UEFI**

UEFI имеет более сложную структуру и намного больше «оберток» всравнении с BIOS. При этом подразумевается поддержка даже сложныхсетевых протоколов, в том числе необходимых для удаленной загрузки.UEFI переходит в защищенный режим с 32-ух- либо 64-ех разряднойадресацией.Сложность требует наличия ПЗУ соответствующего объема, что вовремена BIOS было «роскошью».Место add-on BIOSes заняли специальные UEFI-драйверы.

Для обеспечения удаленной загрузки от производителей сетевыхконтроллеров требуется только написание драйверов. Как правило это UNDI-драйверы, совместимые с UEFI API.

Драйвер может быть, как «прошит» в ПЗУ на плате сетевого адаптера, так и интегрирован в UEFI.

Типичные UEFI ориентированы на IPv4/IPv6 и поддерживает комплекс протоколов: **PXE, Netboot6, iSCSI Boot, FCoE Boot, HTTP Boot**, а также фильтрацию и аутентификацию.

****

Где: MNP -- Managed Network Protocol, SNP -- Simple Network Protocol.

**43 Взаимодействие по протоколу PXE**

PXE представляет собой своеобразную надстройку над DHCP, формализующую три основные вещи:

1. Протокол взаимодействия клиентской станции с сервисами адресации и загрузочными сервисами.

2. Набор APIs, которые образуют «продвинутую» загрузочную среду на клиентской станции.

3. Структуру boot ROM.



1. DHCPDISCOVER -- DHCP-клиент в составе PXE-клиента посылает

броадкаст-запрос с целями анонсирования своего«возникновения» и поиска

сервиса адресации, коим является DHCP-сервер.

2. DHCPOFFER -- DHCP-сервер сразу выдает DHCP-клиенту IP-адрес и в

юникаст- либо броадкаст-форме отвечает о своей готовности.

3. DHCPREQUEST -- DHCP-клиент, по-прежнему в бродкаст-форме,

подтверждает, что он выбрал определенный DHCP-сервер, и параллельно

собственно запрашивает IP-адрес и требующиеся ему конфигурационные

параметры.

*\*вопрос с лекции: почему DHCPREQUEST именно в бродкаст-форме?*

*- отвечаю своими словами!!!*

*Т.к. может быть несколько DHCP-серверов которые предлагают разные IP-адреса. В таком случае получается что реквест отправляется нескольким серверам с одним и тем же выбранным IP-адресом. Вследствие чего сервера сверяют этот адрес с предложенным и отправляют клиенту или ACK (тот где совпал) или NACK (где не совпал). После NACK общение прекращается и остаётся только один сервер.*

4. DHCPACK -- DHCP-сервер в юникаст- либо броадкаст-форме

подтверждает подтверждение

5. Boot Service Discover -- PXE-клиент посредством DHCP посылает

запрос о предоставлении загрузочного сервиса.

6. Boot Service Ack -- DHCP-сервер, находящийся на стороне загрузочного

сервиса, подтверждает предоставление услуг.

7. Bootstrap Request -- TFTP-клиент в составе PXE-клиента посылает

запрос о предоставлении файла -- загрузчика bootstrap

8. Bootstrap Download -- TFTP-клиент скачивает файл -- загрузчик

bootstrap.

9. Bootstrap Exec -- загрузчик bootstrap исполняется.

**44 Протоколы BOOTP, DHCP, TFTP и их использование** \*DHCP есть видео у Созыкина

BOOTP, DHCP и TFTP используют транспорт UDP.

Первым протоколом, который массово использовали для динамического назначения IP-адресов, является BOOTP.

DHCP – аналог BOOTP, но более усовершенствованный:

- В отличии от BOOTP (не способен работать с клиентами DHCP) способен работать с клиентами BOOTP.

- DHCP поддерживает автоконфигурацию (BOOTP поддерживает только ручную настройку)  
 - в DHCP адреса предоставляются на определенный промежуток времени

1. DISCOVER -- DHCP-клиент в составе PXE-клиента посылает

броадкаст-запрос с целями анонсирования своего«возникновения» и поиска

сервиса адресации, коим является DHCP-сервер.

2. OFFER -- DHCP-сервер сразу выдает DHCP-клиенту IP-адрес и в

юникаст- либо броадкаст-форме отвечает о своей готовности.

3. REQUEST -- DHCP-клиент, по-прежнему в бродкаст-форме,

подтверждает, что он выбрал определенный DHCP-сервер, и параллельно

собственно запрашивает IP-адрес и требующиеся ему конфигурационные

параметры.

*\*вопрос с лекции: почему DHCPREQUEST именно в бродкаст-форме?*

*- отвечаю своими словами!!!*

*Т.к. может быть несколько DHCP-серверов которые предлагают разные IP-адреса. В таком случае получается что реквест отправляется нескольким серверам с одним и тем же выбранным IP-адресом. Вследствие чего сервера сверяют этот адрес с предложенным и отправляют клиенту или ACK (тот где совпал) или NACK (где не совпал). После NACK общение прекращается и остаётся только один сервер.*

4. ACK -- DHCP-сервер в юникаст- либо броадкаст-форме

подтверждает подтверждение

Кроме уже упомянутых DHCP\_DISCOVER, DHCP\_OFFER, DHCP\_REQUEST и DHCP\_ACK, есть еще:

DHCP\_DECLINE -- отказ со стороны клиента от IP-адреса, если клиент выявил, что этот IP-адрес уже используется.

DHCP\_NAK -- отказ со стороны сервера, если запрос DHCP\_REQUEST неправильный.

DHCP\_RELEASE -- сообщение от клиента к серверу об освобождении выделенных до этого DHCP-ресурсов, если эти ресурсы больше не нужны.

DHCP\_INFORM -- запрос от клиента к серверу о некоторых конфигурационных параметрах, если собственно IP-адрес назначен «вручную».

DHCP\_FORCERENEW -- сообщение от сервера к клиенту о принудительном начале повторного взаимодействия по DHCP.

Для пересылки файлов используется упрощенный и менее надежный вариант протокола FTP, называемый TFTP. Клиент посылает запрос о предоставлении файла, далее идёт процесс загрузки файлов.

**45 Поддержка удаленной загрузки в Windows и Linux**

DHCP-клиент в **Windows** запускается автоматически и активизируется при отсутствии статического IP-адреса. Интерфейс для явного конфигурирования не предусмотрен. Если назначить IP-адрес с помощью протокола DHCP не удалось, то назначается случайный IP-адрес -- Link Local.

DHCP-клиент в **Linux** представлен демоном *dhclient*. Для активизации dhclient необходимо отредактировать соответствующую строку в соответствующем конфигурационном файле.

Для «тонкой» настройки дополнительно редактируют стандартный конфигурационный файл etc/dhclient.conf.

На загружающей станции должны быть установлены и настроены как минимум два сервиса:

1. DHCP либо ему подобный.

2. TFTP либо ему подобный.

В серверных редакциях Windows имеется возможность установить собственный сервис DHCP. Для конфигурирования используют оснастку DHCP.

Однако, сервис TFTP как отдельный полноценный компонент не поддерживается. Поэтому для организации удаленной загрузки обычно используют стороннее, более «полноценное», ПО.

В Linux используют демоны bootpd и dhcpd со стандартными конфигурационными файлами /tftpboot/bootptab и /etc/dhcpd.conf соответственно. А также демон tftpd, который не имеет конфигурационного файла.

**46 Динамическая IPv4-адресация в IOS**

Запуск DHCP-клиента в Cisco IOS происходит с помощью соответствующего аргумента команды ip address. Перед запуском DHCP-клиента можно настроить.

На маршрутизаторах Cisco поддерживается сервис DHCP. По умолчанию этот сервис запущен, для останова используют команду no service dhcp. Для просмотра состояния сервиса DHCP используют команды группы show ip dhcp: show ip dhcp binding, show ip dhcp conflict, show ip dhcp pool, show ip dhcp server statistics и другие.

Для просмотра состояния сервиса DHCP используют команды группы show ip dhcp; show ip dhcp binding; show ip dhcp conflict, show ip dhcp pool, show ip dhcp server statistics и другие.

ЭТО ИЗ ЛЕКЦИИ

\_\_\_\_\_\_\_\_\_\_ НО я бы почти полностью переписал ответ \_\_\_\_\_\_\_\_\_\_\_\_\_

На маршрутизаторах Cisco поддерживается сервис DHCP

По умолчанию этот сервис запущен, для останова используют команду **no service dhcp.**  
  
Для перехода к настройке DHCP на маршрутизаторе необходимо в режиме глобальной конфигурации написать команду *ip dhcp pool “название пула”*.

В открывшемся dhcp-config необходимо определить default router командой *default-router “адрес интерфейса роутера”*.

Адреса в пул добавляются командой *network “адрес” “маска”.*

Для определения dns-сервера можно воспользоваться командой *dns-server “адрес”*.

Выход из конфигурирования dhcp выполняется командой *exit.*

Для исключения адресов из раздачи используется команда **ip dhcp excluded-address** “адрес” [или диапазон адресов двумя аргументами]

Команда ***show ip dhcp binding*** показывает список всех IP адресов которые были выданы DHCP-сервером.

команда ***show ip dhcp server statistics*** покажет статистику DHCP сервера

***show ip dhcp pool***используется для вывода пула адресов

НАСТРОЙКА РОУТЕРА КАК DHCP КЛИЕНТА

Иногда роутер сам должен получить IP адрес по DHCP, например от интернет-провайдера. Для этого нужно в режиме конфигурации интерфейса ввести команду **ip address dhcp**, после чего интерфейс будет пытаться получить адрес от DHCP сервера.

**47 Специальные соглашения при IPv4-адресации и IPv4-Маршрутизации**

Имеется несколько специальных соглашений в области IP-адресации:

1. 0.0.0.0 -- так называемый Unspecified IPv4-адрес, формально адрес всей глобальной сети Internet, но имеет и другие смыслы, которые будут описаны в дальнейшем.

2. 255.255.255.255 -- формально глобальный широковещательный адрес, но поскольку представляет большу'ю «опасность» уже давно интерпретируется как Limited Broadcast, то есть пакеты с такими адресами назначения должны «безоговорочно» подавляться шлюзами.

3. 127.0.0.1 (как и любой адрес из диапазона 127.X.X.X) -- ассоциирован со специальным сетевым интерфейсом-заглушкой (loopback), необходимым для обеспечения переносимости ПО, то есть пакеты с такими адресами назначения, переданные приложениями, тут же программно возвращаются на прикладной уровень.

Специальные соглашения в области IP-маршрутизации:

1. Адрес назначения 0.0.0.0 -- маршрут по умолчанию.

2. Mаска подсети 255.255.255.255 -- маршрут к одному сетевому интерфейсу

**48 Правила записи IPv6-адресов**

Наряду с общим сохранением преемственности, технологии IPv6 все -таки существенно отличаются от технологий IPv4.

Изменены как длина, так и формат адреса.

**Формат представления и примеры записи одного и того же адреса IPv6:**

**X:X:X:X:X:X:X:X**

**1234:abcd:CDEF:0000:abEF:0000:0000:09aF 1234:abcd:cdef:0:abef::9af**

где Х -- шестнадцатеричное (любой регистр) шестнадцатибитное число. То есть общая длина адреса составляет 128 битов. Поскольку часто встречаются длинные последовательности нулей, одно либо более рядом стоящих нулевых чисел можно сокращать как два двоеточия. Но нужно помнить об однозначности интерпретации адреса. Также можно не писать лидирующие нули в тетрадах.

* · В записи используются латинские буквы только нижнего регистра. Несмотря на то, что шестнадцатеричные цифры сравниваются по коду символа без учёта регистра, IETF предлагает использовать только строчные буквы.
* · Незначащие нули в каждом поле можно опустить, однако каждая группа должна иметь хотя бы один знак, даже если она состоит из одних нулей.
* · Самая длинная последовательность нулевых полей заменяется двумя двоеточиями ("::"). Если таких последовательностей несколько, для предотвращения неоднозначности сжимается крайняя левая. Кроме того, "::" может использоваться для сокращения последовательности из только одного нулевого поля.
* · Двоеточие традиционно используется для завершения пути к хосту перед номером порта, потому IPv6-адрес ограничивается квадратными скобками

**49 IPv6-терминология в сравнении с IPv4-терминологией**

Изменен формат заголовка пакета. Вместо заголовка фиксированной длины с фиксированными полями применяется гибкий базовый заголовок плюс набор необязательных заголовков различного формата.

Изменена иерархия адресного пространства. Применительно к IPv6-адресации механизм классов упразднен. Вместо классов широко применяется механизм адресных префиксов.

Имеются три базовых типа адресов:

1. Юникаст.

2. Мультикаст.

3. Эникаст.

Бродкаст -адресов нет вообще.

Базовые типы, как таковые, не используются. Их делят на виды согласно специфике применения. Принадлежность к тому либо иному виду определяется по адресному префиксу -- фиксированным начальным битам адреса.

Изменен подход к назначению адресов сетевым интерфейсам. Одному и тому же сетевому интерфейсу могут быть назначены несколько адресов различных типов. Допускается даже назначение более одного адреса одного типа и это вполне нормально.

Модифицированы понятия сети и подсети. Если в случае с IPv4 предусматривалась только одна глобальная сеть, то на базе IPv6 предполагается возможность построения нескольких независимых глобальных сетей. Понятие подсети расширено.Особо следует выделить линк -- подсеть размером в один сегмент.

Введены новые правила задания размера подсети. [*Маска подсети, как таковая, аннулирована*] ??? на 51м вопросе понял почему. Размер подсети определяется по префиксу подсети -- фиксированным начальным битам адресов из диапазона описываемой подсети.

Модифицировано понятие станции (узла). Для ссылки на любой из видов пользовательских станций в основном используют обобщенный термин хост. Вместо термина «шлюз» используют обобщенный термин маршрутизатор.

Введено понятие зоны.

Под зоной (zone) понимают некоторую условно выделенную, с целью обеспечения удобства администрирования, виртуальную подобласть в пределах области, очерченной подсетью определенного вида. Зоны не могут пересекаться.

Пример указания зоны IPv6: fd00:0:0:1::11%5 -- число либо строка -- идентификатор зоны

Широко применяют туннелирование.

Фактически туннели выражаются в виде отдельных специализированных сетевых интерфейсов на взаимодействующих станциях.

Туннельный интерфейс должен быть привязан к физическому.

Туннели могут конфигурироваться как полностью «вручную», так и частично автоматически.

**50 Локальные IPv6-адреса типа юникаст**

При IPv6-адресации внутренние адреса, как таковые, не выделяются. Обобщенно их заменяют локальные адреса.

Адрес вида Link-local Unicast (FE80::/10) предназначен для использования в пределах линка. Выход пакетов с адресами Link-local Unicast за пределы линков должен подавляться маршрутизаторами.

Так как пакеты из таких адресов не проходят через маршрутизаторы, то в адресе подсети установлены нули.

Последние 64 бита это идентификатор интерфейса

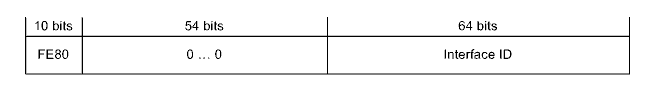


Рисунок -- Формат адреса вида Link-local Unicast

В отличии от глобальных адресов, для формирования локального адреса IPV6, организации не обязательно обращаться в агентства IANA, такие адреса не распространяются в интернет, и их можно формировать самостоятельно.

Как и в других юникаст-адресах, имеется четкое разделение на топологическую и интерфейсную части. Адрес подсети составляет 64 бита, и 64 бита выделено на адрес интерфейса

Адреса Link-local Unicast автоматически генерируются на базе MAC-адресов следующим образом.

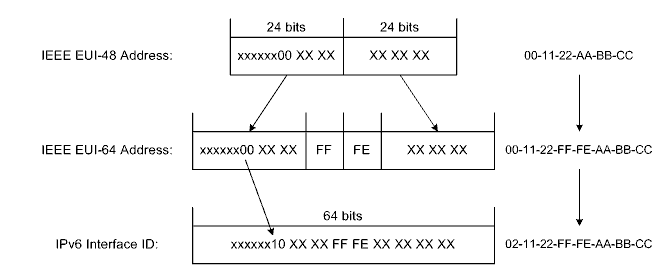


Рисунок -- Алгоритм и пример вычисления значения поля IPv6 Interface ID

В результате, интерфейсная часть соответствует нотации EUI-64 (точнее, модифицированной нотации EUI-64).

Для всех организаций, имеющих более или менее иерархическую подсетевую структуру и не испытывающих потребность во внешнем трафике, в качестве основной замены внутренних адресов IPv4 позиционируются адреса вида Unique Local Unicast (FC00::/7) или (FD00::/8). Пакеты с адресами Unique Local Unicast должны подавляться всеми маршрутизаторами кроме внутренних.

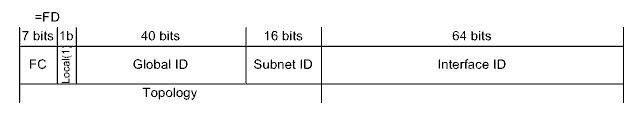
**

Рисунок -- Формат адреса вида Unique Local Unicast

Для всех юникаст-адресов, в том числе Unique Local Unicast, приемлема (но не всегда удобна) EUI-64-нотация интерфейсной части.

**51 Глобальные IPv6-адреса типа юникаст**

Если в случае с IPv4 предусматривалась только одна глобальная сеть, то на базе IPv6 предполагается возможность построения нескольких независимых глобальных сетей.

В качестве основной замены реальных адресов IPv4 предлагаются адреса вида Global Unicast (2000::/3).

Глобальный адрес IPv6, состоит из двух частей, адрес сети и адрес интерфейса.

Длина адреса сети и идентификаторов интерфейса — 64 бита

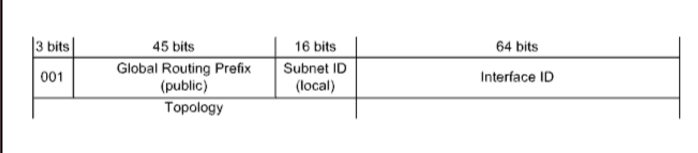
Адрес сети в свою очередь делится на две части, глобальный префикс маршрутизации, длина которого 48 бит, глобальные префиксы должны быть уникальными и они распределяются IANA, среди организаций. Следующие 16 бит могут использоваться организацией, для того чтобы разбить сеть на отдельные подсети 

Рисунок -- Формат адреса вида Global Unicast

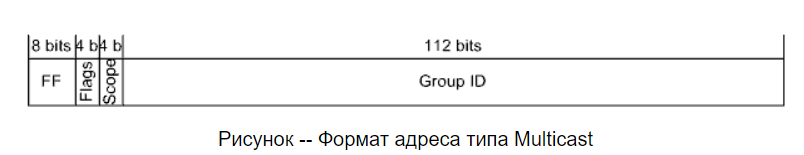
Префикс глобальной маршрутизации: наиболее значимые 48-разрядные обозначаются как префикс глобальной маршрутизации, который назначается конкретной автономной системе. Три наиболее значимых бита префикса глобальной маршрутизации всегда установлены на 001.

Соглашения в области IPv6-адресации:

1. Unspecified (::/128) -- адрес всех глобальных сетей.

2. Loopback (::1/128) -- адрес сетевого интерфейса -- заглушки.

**52 IPv6-адреса типа мультикаст и стандартные подсети**

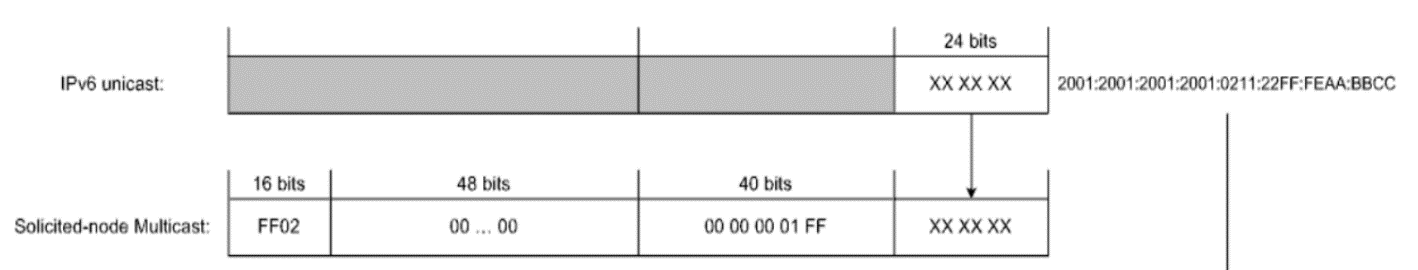
Адрес типа Multicast (FF00::/8) предназначен для использования в пределах подсети определенного вида и представляет собой уникальный в пределах таковой подсети групповой идентификатор.  


Примеры стандартных видов:

Link-local All Nodes Multicast (FF02::1/128),

Link-local All Routers Multicast (FF02::2/128),

Site-local All Routers Multicast (FF05::2/128) и так далее.

Кроме того, при автоконфигурировании в пределах линка используются специальные адреса вида Solicited-node Multicast (FF02::1:FF00/104), строящиеся на основе адресов Link-local Unicast, из которых переносятся последние 24 бита.

**Новая картинка**

Мультикаст-адреса могут присутствовать в пакетах только в поле Destination Address.

Понятие подсети расширено. Стандартизированы следующие виды подсетей, что, в частности, отражается в значениях специального для этого введенных четырехбитных полей Scope в форматах адресов некоторых видов: 0, F -- Reserved.

1 -- Interface-local.

2 -- Link-local.

3 -- Realm-local.

4 -- Admin-local.

5 -- Site-local.

6, 7, 9, A, B, C, D -- Unassigned (по своему усмотрению).

8 -- Organization-local.

E -- Global.

Таким образом «очерчивается круг», в пределах которого адрес валиден и применяется.

Особо следует выделить линк (link) -- подсеть размером в сегмент.

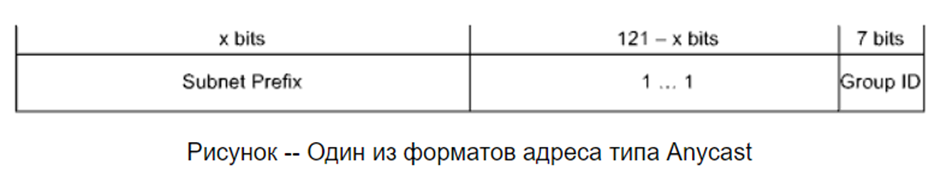
**53 IPv6-адреса типа эникаст**

Применительно к IPv6 эникаст-адреса обладают двумя специфическими свойствами (так задумывалось).

Во-первых, если юникаст-адрес присвоить более чем одному сетевому интерфейсу в подсети, то он превращается в эникаст-адрес.

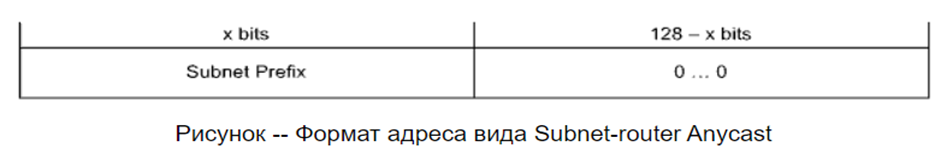
Во-вторых, критерием выбора эникаст-адреса является кратчайшесть расстояния при маршрутизации.

Адрес типа Anycast предназначен для использования в пределах подсети и получается **на основе префикса подсети**.



Соответствующие приведенному выше формату, одному из двух форматов Reserved Subnet Anycast, виды пока применения не нашли.

Единственным используемым на практике видом является Subnet-router Anycast.



Такие адреса разрешено назначать только сетевым интерфейсам маршрутизаторов и они могут присутствовать только в соответствующих служебных пакетах, причем только в поле Destination Address.

**54 Нотация EUI-64 и инкапсуляция IPv6-адресов типа мультикаст**

Для всех юникаст-адресов, в том числе Unique Local Unicast, приемлема (но не всегда удобна) EUI-64-нотация интерфейсной части.

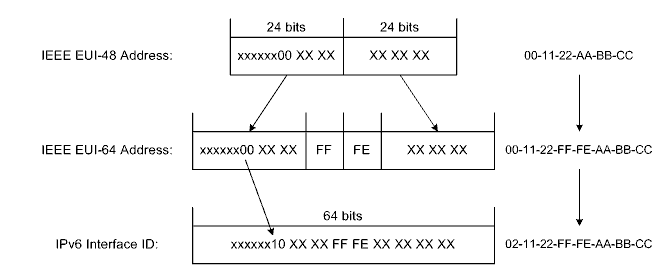


Рисунок -- Алгоритм и пример вычисления значения поля IPv6 Interface ID

Инкапсуляция IPv6-адресов типа мультикаст (Multicast) происходит путем использования специального группового адреса IPv6, который используется для доставки пакетов одновременно нескольким узлам в рамках одной сетевой подсети. При инкапсуляции IPv6-адреса мультикаста в заголовок пакета добавляется соответствующий групповой адрес.

Формат групповых адресов IPv6 для мультикаста начинается с префикса FF00::/8, и оставшиеся биты определяют конкретную группу мультикаста. Этот адрес может быть использован источником для отправки пакетов мультикаста и получателем, который является частью этой группы мультикаста.

Примеры групповых адресов мультикаста в IPv6:

FF02::1: Адрес "Все узлы на локальной подсети"

FF02::2: Адрес "Все маршрутизаторы на локальной подсети"

FF02::9: Адрес "Все RIP маршрутизаторы"

FF02::16: Адрес "Все OSPFv3 маршрутизаторы"

FF02::1:FF00:0/104: Адрес "Соответствие IPv6-адреса узла и MAC-адреса"

При инкапсуляции пакета IPv6 с адресом мультикаста, пакет будет отправлен на все узлы, которые присоединены к этой группе мультикаста в пределах сети. Получающие узлы, которые являются частью этой группы, смогут распознать и обработать пакет.

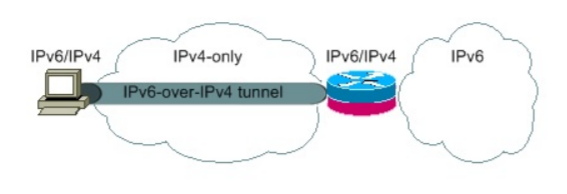
Использование инкапсуляции IPv6-адресов типа мультикаст позволяет эффективно реализовывать групповую коммуникацию в рамках IPv6-сетей и обеспечивает распространение информации сразу на несколько узлов одновременно.

**55 Совместимость IPv6 с IPv4**

В контексте совместимости IPv4 и IPv6, практический интерес представляет лишь возможность передавать трафик IPv6 посредством трафика IPv4, то есть организовывать туннели IPv6-over-IPv4.

| **Флаг** | **Описание** |
| --- | --- |
| Маршрутизатор-маршрутизатор | Пакеты **IPv6** могут передаваться через сеть **IPv4** от одного маршрутизатора **IPv6** или **IPv4** к другому такому же маршрутизатору. В этом случае туннель - это один из транзитных участков на пути пакета **IPv6**. |
| Хост-маршрутизатор | Хост **IPv6** может передавать пакеты **IPv6** маршрутизатору **IPv6** через сеть **IPv4**. Такой туннель представляет собой первый транзитный участок пути пакета. |
| Хост-хост | Пакеты **IPv6** могут передаваться через сеть **IPv4** от одного хоста **IPv6** или **IPv4** к другому такому же хосту. В этом случае туннель - это весь путь пакета. |
| Маршрутизатор-хост | Маршрутизаторы **IPv6** и **IPv4** могут передавать пакеты **IPv6** целевым хостам **IPv6** или **IPv4**. Такой туннель представляет собой последний транзитный участок пути пакета. |

 Существует несколько типов туннелирования:



В Cisco для настройки туннеля необходимо создать интерфейс. Это делается автоматически после написания команды int tunnel0 (не обязательно 0) в режиме конфигурации девайса. После чего необходимо установить следующие настройки: адрес интерфейса туннеля, тип туннеля, адрес назначения, интерфейс, через который проходит туннель. Выполняется это следующими командами: (в данном случае R2 наш девайс, на котором настраиваем туннель)

R2(config-if)#ipv6 address fd3e:1e67:e6a2:2::5/64

R2(config-if)#tunnel source FastEthernet0/0

R2(config-if)#tunnel mode ipv6ip

R2(config-if)#tunnel destination 154.166.0.2

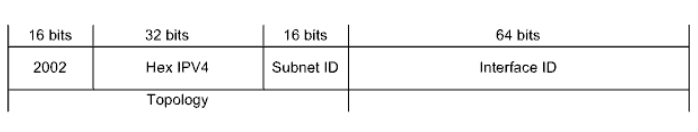
В данном случае установлен туннель типа manual.

Также можно выделить туннели 6to4 (адрес начинается с 2002), 6in4, 6rd, Teredo, ISATAP

Для обеспечения совместимости с IPv4 стандартизированы следующие виды адресов IPv6. Адрес вида IPv4-compatible (::D.D.D.D/128). Включает публичный адрес IPv4. В настоящее время использование этих адресов не рекомендуется.

Адрес вида IPv4-mapped (::FFFF:D.D.D.D/128). Предназначен для использования при работе с виртуальной станцией IPv4 внутри станции IPv6. В физических пакетах эти адреса недопустимы и в основных реализациях не поддерживаются.

Адрес вида 6to4 Unicast



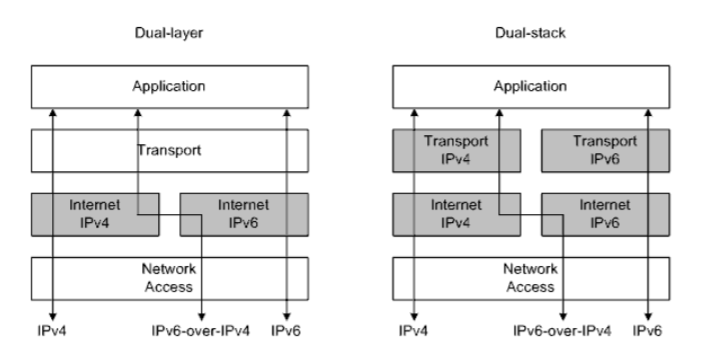
Включает шестнадцатеричное представление публичного адреса IPv4 и предназначен для формирования автоматических туннелей. Это один из видов туннельных адресов, поддерживаемый всеми основными реализациями.

Вопросы совместимости IPv4 и IPv6 затрагивают работу всего семейства протоколов TCP/IPv6. Выделяют две архитектуры:

1. Dual-layer -- IPv4 и IPv6 разделены только на Internet-уровне модели TCP/IP.

2. Dual-stack -- стеки TCP/IPv4 и TCP/IPv6 полноценны и независимы

В оптимальном случае трафик IPv6 полностью отделен от трафика IPv4.



**56 Модели и задачи IPv6-автоконфигурирования**

В сравнении с IPv4, возможности динамической IPv6-адресации значительно расширены и усовершенствованы, вплоть до полного автоконфигурирования. Предусмотрены две базовых модели:

1. Stateless -- распределенное управление, адреса и другие параметры конфигурируют с помощью служебных сообщений, базируется на ICMPv6. (SLAAC)

2. Stateful -- централизованное управление, адреса и другие параметры передаются по специальному протоколу, базируется на DHCPv6. Причем, в качестве приоритетной модели рассматривают первую, а не вторую.

ICMPv6, кроме всего прочего, включает в себя два мощных функционала:

1. Neighbor Discovery (ND) -- граничное обнаружение.

2. Multicast Listener Discovery (MLD) -- обнаружение мультикаст-станции-потребителя.

При разработке ND были четко сформулированы девять задач для решения в границах линка:

1. Обнаружение соседних маршрутизаторов. (вопрос 57)

2. Восстановление значений префиксов подсетей. (58)

3. Восстановление значений некоторых других параметров. (58)

4. Автоконфигурирование адресов. (59)

5. Восстановление MAC-адресов соседних станций. (как ARP в IPv4) (60)

6. Обнаружение маршрутизаторов следующего звена. (57)

7. Проверка достижимости соседних станций. (61)

8. Проверка конфликтов адресов. (60)

9. Оптимизация маршрутов. (57)

Важно, что задачи ND решают именно **в пределах линка**.

Для обеспечения ND предусмотрены пять типов ICMPv6-сообщений:

133. RS (Router Solicitation).

134. RA (Router Advertisement).

135. NS (Neighbor Solicitation).

136. NA (Neighbor Advertisement).

137. Redirect.

**57 Обнаружение маршрутизаторов и оптимизация маршрутов при IPv6-автоконфигурировании**

ICMPv6 включает в себя два мощных функционала:

1. Neighbor Discovery (ND) -- граничное обнаружение.

2. Multicast Listener Discovery (MLD) -- обнаружение мультикаст-станции-потребителя.

Обнаружение маршрутизаторов и оптимизация маршрутов при IPv6-автоконфигурировании являются двумя из девяти задач ND

При разработке ND были четко сформулированы девять задач для решения в границах линка:

1. Обнаружение соседних маршрутизаторов.

2. Восстановление значений префиксов подсетей.

3. Восстановление значений некоторых других параметров.

4. Автоконфигурирование адресов.

5. Восстановление MAC-адресов соседних станций. (как ARP в IPv4)

6. Обнаружение маршрутизаторов следующего звена.

7. Проверка достижимости соседних станций.

8. Проверка конфликтов адресов.

9. Оптимизация маршрутов.

Для обеспечения ND предусмотрены пять типов ICMPv6-сообщений:

133. RS (Router Solicitation).

134. RA (Router Advertisement).

135. NS (Neighbor Solicitation).

136. NA (Neighbor Advertisement).

137. Redirect.

Для решения первой задачи используется связка RS и RA. Согласно стандарту ND, маршрутизаторы должны не только отвечать на RSes, а и периодически передавать RAs «на упреждение», анонсируя свое присутствие в линке.

Для решения последней задачи используется специальное сообщение

Redirect.

Сообщение Redirect содержит следующие ключевые поля:

- Target Address -- адрес Link-local Unicast соседа, которому в дальнейшем нужно напрямую передавать пакеты с указанным адресом назначения;

- Destination Address -- адрес назначения

и две ND-опции:

- Target Link-layer Address -- MAC-адрес соседа, которому в дальнейшем нужно напрямую передавать пакеты;

- Redirected Header -- фрагмент предварительно принятого пакета

Любой маршрутизатор, сам по себе, уже является потенциальным маршрутизатором следующего звена для соседних хостов (маршрутизаторов) -- если требуется послать пакет за пределы соответствующего линка.

При автоконфигурировании все соседние маршрутизаторы автоматически рассматриваются как кандидаты в маршрутизаторы по умолчанию -- создается специальный список

Текущий маршрутизатор по умолчанию рекомендуется выбирать исходя

из состояния ND-кэша (вопрос 61). А также исходя из значения специального

Router Lifetime -- время жизни маршрутизатора.

Если же в линке оказывается несколько равноценных маршрутизаторов, то очевидно возникает проблема выбора. В базовой редакции стандарта ND четко не оговорено поведение в таких случаях. В реализациях, как правило, выбирается первый «попавшийся» маршрутизатор.

**58 Восстановление параметров при IPv6-автоконфигурировании**

*\*можно продублировать всю информацию про ND (56 вопрос)*

Можно описать две из 9 задач ND:

2. Восстановление значений префиксов подсетей.

3. Восстановление значений некоторых других параметров.

Для обеспечения ND предусмотрены пять типов ICMPv6-сообщений:

133. RS (Router Solicitation).

134. RA (Router Advertisement).

135. NS (Neighbor Solicitation).

136. NA (Neighbor Advertisement).

137. Redirect.

*Задача восстановления значения префикса подсети:*

Хост (маршрутизатор) восстанавливает значения префиксов подсетей путем анализа RA.

Префикс подсети анонсируется маршрутизатором в виде ND-опции Prefix Information со следующими ключевыми полями:

Prefix Length -- длина префикса;

Valid Lifetime -- общее время жизни;

Preferred Lifetime -- интервал времени, в течение которого адрес, сгенерированный на основе данного префикса подсети, будет считаться предпочтительным;

Prefix -- собственно префикс подсети;

В RA вкладывается столько ND-опций, сколько нужно.

В результате анализа RA, маршруты ко всем соответствующим подсетям автоматически вносятся в таблицу маршрутизации -- как маршруты к своим подсетям.

*Задача восстановления других параметров:*

Хост (маршрутизатор) восстанавливает значение еще двух важных параметров, опять же, путем анализа RA.

Первым таковым параметром является Cur Hop Count.

Вторым параметром является MTU (максимальный размер пакета, который может быть передан по сети).

**59 Автоконфигурирование адресов и их жизненный цикл при IPv6-автоконфигурировании**

Автоконфигурирование адресов является одной из задач ND.

Для обеспечения ND предусмотрены пять типов ICMPv6-сообщений:

133. RS (Router Solicitation).

134. RA (Router Advertisement).

135. NS (Neighbor Solicitation).

136. NA (Neighbor Advertisement).

137. Redirect.

В контексте SLAAC (Stateless Address Autoconfiguration) под автоконфигурированием адресов понимают автоматическое назначение сетевому интерфейсу юникаст-адресов, не затрагивая адреса Link-local Unicast. Адреса Link-local Unicast также назначаются автоматически, но вне рамок автоконфигурирования.

Топологическая часть адреса берется из ND-опции Prefix Information в RA от маршрутизатора, а для интерфейсной части используется нотация EUI-64.

Автоконфигурирование адресов также подразумевает и автоматическое нахождение маршрутизатора по умолчанию.

SLAAC и DHCPv6 (Stateful из вопроса 56) вполне совместимы друг с другом. «Разделение обязанностей» контролируется флагами в RA.

Состояния адреса, полученного в результате автоконфигурирования:

1. **Tentative** -- уникальность адреса еще не проверена.

2. Preferred -- адрес является предпочтительным.

3. **Deprecated** -- использование адреса нежелательно.

4. Valid -- адрес находится в состоянии Preferred либо Deprecated.

5. Invalid -- время жизни адреса истекло.

Валидность сгенерированных адресов контролируется двумя таймерами:

Preferred Lifetime -- интервал времени, в течение которого адрес является предпочтительным;

Valid Lifetime -- интервал времени, равный собственно времени жизни адреса. Таймеры инициализируются исходя из значений соответствующих полей в сообщениях ND либо DHCPv6.

**60 Восстановление адресов при IPv6-автоконфигурировании и проверка конфликтов адресов**

Восстановление MAC-адресов и проверка конфликтов адресов являются двумя из девяти задач ND

Для обеспечения ND предусмотрены пять типов ICMPv6-сообщений:

133. RS (Router Solicitation).

134. RA (Router Advertisement).

135. NS (Neighbor Solicitation).

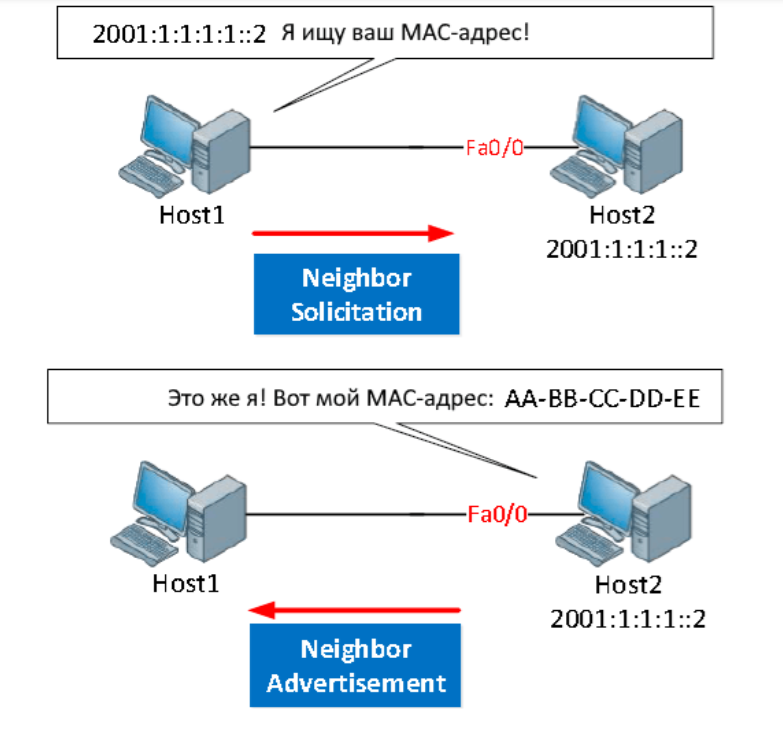
136. NA (Neighbor Advertisement).

137. Redirect.

Для решения задачи восстановления адреса используется связка NS

и NA.

Это аналог ARP-протокола в IPv4



NA содержит три флага:

R (Router) -- данное NA передано маршрутизатором (не хостом),

S (Solicited) -- данное NA передано в ответ на NS,

O (Override) -- данное NA содержит новый MAC-адрес.

Не смотря на всю гибкость IPv6 конфликты адресов никто не отменял

Задача решается передачей специальным образом наполненного NS (с нулевым IPv6-адресом источника) и проверкой есть ли ответ.

**61 Проверка достижимости при IPv6-автоконфигурировании**

Проверка достижимости является одной из девяти задач ND.

В отличие от ARP, проверка достижимости соседа проводится, причем по мере надобности -- упор сделан на то, что сетевые интерфейсы способны сообщать о своем состоянии.

Одна проверка достижимости соседа подразумевает несколько попыток (по умолчанию три).

Алгоритм проверки достижимости опирается на два основных таймера:

-Reachable Time -- интервал времени после приема последнего сообщения NА от соседа, в течение которого этот сосед считается достижимым (согласно стандарту по умолчанию генерируется случайно в диапазоне 15-45 s).

-Retrans Timer -- интервал между передачей NSes при переходе к следующей попытке (согласно стандарту по умолчанию 1 s).

Каждый сетевой интерфейс IPv6 должен иметь свой ND-кэш. ND-кэш напоминает ARP-таблицу.

Каждому из соседей в ND-кэше соответствует строка и одно из состояний:

1. **INCOMPLETE** -- сосед неизвестен, возникла необходимость передать ему пакет, идет восстановление его MAC-адреса.

2. REACHABLE -- сосед известен и считается достижимым.

3. **STALE** -- сосед известен, уже считается недостижимым, но нет необходимости передать ему пакет.

4. DELAY -- сосед известен, считается недостижимым, возникла необходимость передать ему пакет, пакет передан, ожидается подтверждение от протоколов вышестоящих уровней

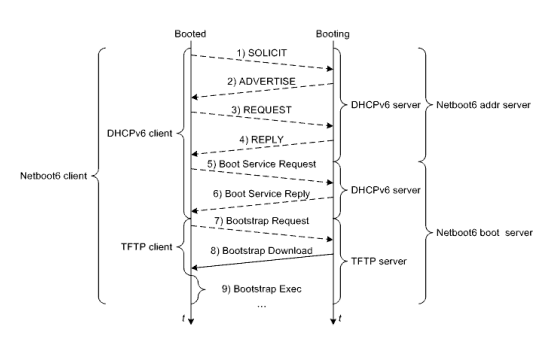
5. PROBE -- идет собственно проверка достижимости соседа.

Reachable Time задает длительность состояния REACHABLE.

Есть еще таймер, который задает длительность состояния DELAY (согласно стандарту по умолчанию 5 s).

**62 Взаимодействие по протоколу Netboot6**

Netboot6 – аналог PXE для IPv6. Solicit – аналог DHCPDISCOVER, advertise – аналог dhcpoffer. В целом взаимодействие такое же, как и по PXE.



1.Клиент рассылает сообщение SOLICIT

(По сути, клиент описывает себя (какая архитектура и тд)

2.Служба DHCP6 или прокси-сервер DHCP6 отвечает отправкой сообщения ADVERTISE

3-4. Если клиент конфигурируется службой DHCP6, он должен произвести стандартную настройку с помощью DHCP6, отправив на сервер REQUEST и получив REPLY.

5. Boot Service Req -- Netboot6-клиент посредством DHCP посылает

запрос о предоставлении загрузочного сервиса.

6. Boot Service Rep -- DHCP-сервер, находящийся на стороне загрузочного

сервиса, подтверждает предоставление услуг.

7. Bootstrap Request -- TFTP-клиент в составе Netboot6-клиента посылает

запрос о предоставлении файла -- загрузчика bootstrap

8. Bootstrap Download -- TFTP-клиент скачивает файл -- загрузчик

bootstrap.

9. Bootstrap Exec -- загрузчик bootstrap исполняется.

**63 Протоколы DHCPv6 и его использование** *(сокращен)*

DHCPv6 — это сетевой протокол для конфигурации узлов версии 6 (IPv6)

**По сути, такой же DHCP, как и в V4.**

*Можно расписать как работает*

DHCPv6 делится на следующие два типа:

- DHCPv6 имеет автоматическое распределение с отслеживанием состояния. Сервер DHCPv6 автоматически назначает адреса IPv6 / префиксы PD и другие параметры конфигурации сети (адреса DNS, NIS, SNTP-серверов и т. Д.).

Грубо говоря выдаёт IP и все остальное

- Автоконфигурация IPv6 без Ip.

IPv6-адрес хоста по-прежнему автоматически генерируется посредством объявления маршрута. Сервер DHCPv6 выделяет только параметры конфигурации, кроме адреса IPv6, включая параметры DNS.

Грубо говоря НЕ выдаёт IP, но выдаёт все остальное

DHCPv6-сервер способен выдавать как постоянные, так и вре'менные адреса.

Постоянные адреса имеют Valid Lifetime и Preferred Lifetime.

Для обеспечения выдачи и последующего сопровождения адресов, между DHCPv6-клиентом и DHCPv6-сервером создается ассоциация с уникальным идентификатором.

Валидность выданных адресов контролируется двумя таймерами:

T1 -- интервал времени, начиная с приема REPLY, по истечении которого необходимо передать RENEW

T2 -- интервал времени, начиная с приема REPLY, по истечении которого необходимо передать REBIND, если не поступило ответа на RENEW.

Если не поступило ответа на REBIND, то по истечении Valid Lifetime адрес становится недействительным.

Кроме адресов, посредством DHCPv6 можно передавать префиксы подсетей.

**64 Проблемы при IPv6-маршрутизации и их решения**

1. Общее правило IP-адресации гласит, что **подсети, к которым относятся разные сетевые интерфейсы маршрутизатора, не должны перекрываться. Формат адресов LLU напрямую нарушает приведенное правило** и порождает проблему выбора выходного сетевого интерфейса при передаче пакета, созданного на маршрутизаторе вне рамок ND. В таблице маршрутизации возникают несколько абсолютно равноправных маршрутов к подсети. Проблему решают явным указанием выходного сетевого интерфейса.

2. Возникает и еще один закономерный вопрос -- о том, адреса каких видов использовать для указания маршрутизаторов следующего звена при вводе статических маршрутов. Согласно рекомендациям о применении IPv6, **при настройке статической маршрутизации между маршрутизаторами, для ссылки на маршрутизаторы следующего звена рекомендуется использовать адреса Link-local Unicast**, как это и делают протоколы динамической маршрутизации. А маршрутизатор по умолчанию для хостов рекомендуется назначать автоматически -- посредством ND.

Имеет право на существование альтернативный подход, заключающийся в независимой настройке статической маршрутизации в отношении подсетей различных видов.

3. Согласно идеологии IPv4, в качестве адреса источника подставляется адрес выходного интерфейса. **Наличие у одного сетевого интерфейса множества адресов разных видов создает проблему выбора адреса источника при инкапсуляции, когда пакет создан на самом маршрутизаторе и адрес источника явно не задан.**

Сформулированы восемь единых правил для всех реализаций:

1. Приоритетнее адрес, совпадающий с адресом назначения.

2. Приоритетнее адрес из подсети, вид которой более приближен к виду подсети назначения.

3. Preferred-адрес приоритетнее deprecated-адреса.

4. Домашний адрес приоритетнее дорожного адреса.

5. Приоритетнее адрес сетевого интерфейса, обращенного в сторону адреса назначения.

6. Приоритетнее адрес, чья метка равна метке адреса назначения.

7. Временный адрес приоритетнее постоянного.

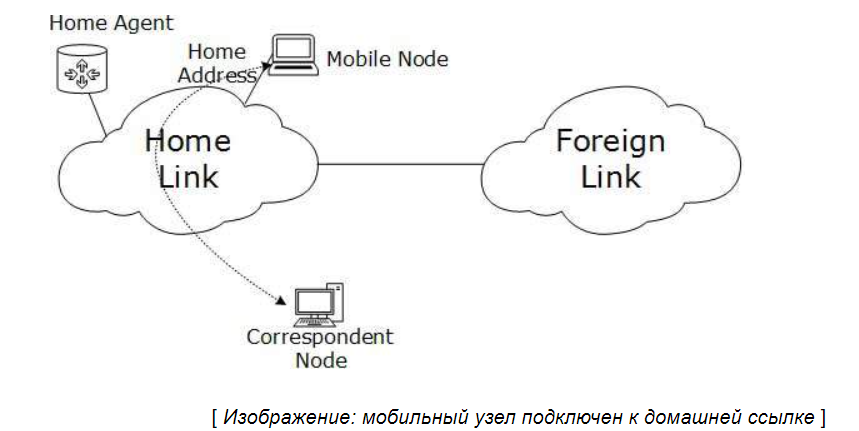
8. Приоритетнее адрес из подсети, которая имеет наиболее длинный общий префикс с подсетью назначения

Адреса сравниваются попарно. Если текущее правило не выявило победителя, то выполняется переход к следующему правилу. Если в результате выявить одного победителя не удалось, то дальнейший выбор зависит от реализации.

**65 Поддержка мобильности в IPv6**

Новой возможностью IPv6 является заложенная целенаправленная поддержка адресации мобильных станций.

Мобильный хост изначально «приписан» к своему домашнему линку. В домашнем линке мобильному хосту, как правило автоматически, назначается домашний адрес. В домашнем линке определен домашний префикс подсети.



Любой доступный линк, в который мобильный хост может быть перемещен из домашнего, является для этого хоста чужим линком. В чужом линке мобильному хосту также назначается адрес – дорожный адрес. В чужом линке определен чужой префикс подсети. Если мобильный хост находится в чужом линке, то он регистрируется у своего домашнего агента, который затем перенаправляет трафик с домашнего адреса на дорожный адрес через специально создаваемый туннель.



Таким образом, мобильный хост всегда доступен по домашнему адресу, вне зависимости от места фактического подключения.

*Поддержка мобильности реализуется посредством следующих составляющих:*

*1. Специальный заголовок Mobility header -- заголовок для обеспечения мобильности.*

*Этот заголовок используется для пересылки восьми типов mobility-сообщений. Все mobility-сообщения обеспечивают привязку мобильного хоста. Mobility-сообщения могут включать в себя различные mobility-опции.*

*2. Дополнительная опция для пересылки с помощью заголовка предназначенных станции назначения опций: Home Address. С помощью этой опции мобильный хост указывает свой домашний адрес.*

*3. Специальный тип маршрутизационного заголовка. Используется для пересылки пакета от станции-корреспондента напрямую к мобильной станции и содержит домашний адрес.*

*4. Четыре вида ICMPv6-сообщений. Используются при взаимодействии мобильного хоста и домашнего агента.*

*5. Дополнения ND.*

**66 Специальные соглашения при IPv6-адресации и IPv6-маршрутизации**

Соглашения в области IPv6-адресации:

1. Unspecified (::/128) -- адрес всех глобальных сетей.

2. Loopback (::1/128) -- адрес сетевого интерфейса -- заглушки.

Изменения в маршрутизации.

Специальные соглашения:

1. ::/0 -- маршрут по умолчанию.

2 X … X < 64 - маршрут к большей чем линк подсети.

3. X:X:X:X/64 -- маршрут к подсети (в том числе и оконечной) размером с линк.

4. X:X:X:X:X:X:X:X/128 -- маршрут к одному сетевому интерфейсу

*В лекции по IPv6 ответ такой же. Больше ничего не нужно*

**67 Статическая и динамическая IPv6-адресация в Windows**

В Windows XP SP2 и Server 2003 поддержка IPv6 уже была интегрирована в составе Advanced Networking Pack и устанавливалась как опциональный компонент с помощью графического интерфейса (свойства сетевых интерфейсов) либо командой netsh interface ipv6 install. Для работы с адресами использовались только расширения команды netsh interface ipv6 (вместо отмененной команды ipv6).

Полноценная поддержка IPv6 доступна начиная с Windows Vista и Server 2008. Может быть задействован как графический интерфейс, так и различные варианты команды netsh interface ipv6.

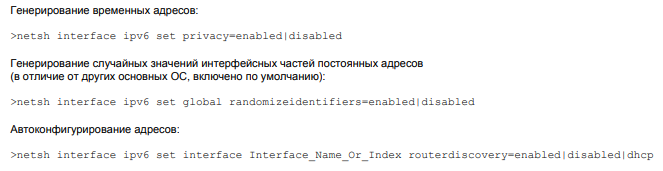
Начиная с Windows 10 1607 по умолчанию запрещен туннельный интерфейс 6to4, Windows 10 1703 -- ISATAP, Windows 10 1803 -- Teredo.

Следует обратить внимание на то, что по умолчанию автоконфигурирование работает даже при статическом конфигурировании адресов.

Конфигурирование IPv6 в Windows 10:

Win + R -> Сеть и интернет -> Настройка параметров адаптера. Далее необходимо выбрать сетевое подключение и открыть его Свойства, В списке выбрать TCP/IPv6 и нажать Свойства. В открывшемся окне можно выбрать автоконфигурирование либо настроить IPv6 вручную.

*В лекции есть еще пункт про автоконфигурирование. Это типо там же где и обычное только галочку (точку в кружке) нужно выставить на автоматическое.*

****

**68 Статическая и динамическая IPv6-адресация в Linux**

В Linux поддержка IPv6 имеется в дистрибутивах с ядрами 2.2.х и последующими.

**Статическая IPv6-адресация в Linux:**

*Назначение статического IPv6-адреса на интерфейс:*

Используйте команду ip addr add <IPv6-адрес>/<префикс> dev <интерфейс>, чтобы добавить статический адрес к интерфейсу.

Настройки IPv6-адресации также могут быть определены в конфигурационных файлах сети, таких как /etc/network/interfaces или /etc/sysconfig/network-scripts/ifcfg-<интерфейс>.

**Динамическая IPv6-адресация в Linux:**

*Использование протокола SLAAC (Stateless Address Autoconfiguration):*

При использовании SLAAC, Linux-хосты могут автоматически получать IPv6-адреса, основываясь на информации в объявлениях маршрутизации (Router Advertisement, RA), отправляемых маршрутизаторами в локальной сети.

*Использование протокола DHCPv6 (Dynamic Host Configuration Protocol for IPv6):*

DHCPv6 позволяет динамически назначать IPv6-адреса, настройки DNS, маршрутов и других параметров хостам в сети.

Для использования DHCPv6 в Linux можно установить соответствующий пакет (например, dhcpd для сервера DHCPv6) и настроить его в соответствии с требованиями сети.

В Linux можно комбинировать статическую и динамическую IPv6-адресацию, используя подход, наиболее подходящий для конкретных сетевых требований.

**69 Статическая и динамическая IPv6-адресация в IOS**

На маршрутизаторах и коммутаторах Cisco IPv6-возможности по умолчанию находятся в административно выключенном состоянии.

В режим IPv6-маршрутизатора переключают командой ipv6 unicast-routing (в глобальном конфигурационном режиме)

*\*без этого не работает IPv6*

Для административного включения на сетевом интерфейсе IPv6 и автоматической генерации адреса Link-local Unicast используют команду ipv6 enable.

Как альтернатива, позволяющая в добавок задействовать возможности ND, выступает команда ipv6 address autoconfig.

Для присвоения сетевому интерфейсу адреса Unique Local Unicast либо Global Unicast, и тем самым активации на нем IPv6, используют команду ipv6 address. После ввода первого такого адреса автоматически генерируется и адрес Link-local Unicast.

Вариант с аргументом eui-64 позволяет автоматически сгенерировать соответствующее значение интерфейсной части адреса.

Вариант с аргументом link-local позволяет заменить автоматически сгенерированный адрес Link-local Unicast.

Вариант с аргументом anycast позволяет добавить соответственно эникаст-адрес.

Router(config-if)#ipv6 address 2001:7f8:8b:6::1/64

Router(config-if)#ipv6 address fd5f:4cf8:7fcd:6::/64 eui-64

Router(config-if)#ipv6 address fe80::1 link-local

Router(config-if)#ipv6 address 2001:7f8:8b:6::/64 anycast

Router(config-if)#ipv6 mld join-group ff04::10

*// При вводе адресов можно использовать заранее подготовленные именованные префиксы, которые создают с помощью команды ipv6 general-prefix. Для работы с мультикаст-группами используют различные варианты команды ipv6 mld, например, ipv6 mld join-group. Шестнадцатеричные цифры в IPv6-адресах при выводе на экран и при переносе в конфигурационные файлы приводятся к верхнему регистру.*

Для вывода на экран IPv6-информации о сетевом интерфейсе используют команду **show ipv6 interface.**

Для управления ND в основном используют различные варианты команды ipv6 nd (в режиме конфигурирования интерфейса).

*\*в лекции идет список примерно 15-20 команд которые учить неадекватно*

Команды ping и traceroute совместимы с IPv6.

Для просмотра таблицы IPv6-маршрутизации используют команду show ipv6 route. *(это уже к маршрутизации относится)*

В Cisco IOS команды для настройки динамической IPv6-адресации включают:

Для настройки SLAAC: ipv6 enable на интерфейсе и ipv6 address autoconfig на глобальном уровне конфигурации.

Для настройки DHCPv6: ipv6 dhcp client на интерфейсе для получения адресов от DHCPv6-сервера.

**70 Поддержка совместимости IPv6 с IPv4 в Windows, Linux и IOS**

Поддержка совместимости IPv6 с IPv4 в операционных системах Windows, Linux и IOS может быть реализована с помощью следующих механизмов:

Двухстековая (**Dual Stack**) поддержка: Этот подход предполагает наличие отдельных стеков для IPv6 и IPv4. Операционные системы с поддержкой двухстековой архитектуры могут работать одновременно с обоими протоколами. При наличии IPv6-сети, система будет использовать IPv6, а при отсутствии таковой - IPv4.

Туннелирование (**Tunneling**): Этот механизм позволяет протоколу IPv6 передавать свои пакеты через IPv4-сеть. Примеры протоколов туннелирования включают 6to4, ISATAP (Intra-Site Automatic Tunnel Addressing Protocol), Teredo и GRE (Generic Routing Encapsulation).

Механизмы перевода адресов (**Address Translation**): Эти механизмы позволяют связывать IPv6- и IPv4-сети с помощью преобразования адресов. Примеры включают NAT64 (Network Address Translation 64) и NPTv6 (Network Prefix Translation IPv6).

В Windows, поддержка совместимости IPv6 с IPv4 доступна с помощью функций TCP/IP стека,

а в Linux - с использованием модулей ядра и инструментов, таких как ipv6, netfilter, ip6tables

В IOS (Cisco Internetwork Operating System), поддержка IPv6 и IPv4 реализуется через команды и настройки маршрутизатора.

**Конкретные примеры** поддержки совместимости IPv6 с IPv4 в различных операционных системах:

***В Windows:***

Настройка двухстековой поддержки: netsh interface ipv6 install

Активация туннельных протоколов: netsh interface ipv6 6to4 set state enabled

Настройка NAT64: netsh interface ipv6 set prefixpolicy ::ffff:0:0/96 46 1

***В Linux:***

Активация IPv6: sysctl -w net.ipv6.conf.all.disable\_ipv6=0

Настройка туннельных протоколов: ip tunnel add tunnel0 mode sit remote <IPv4-адрес> local <IPv6-адрес>

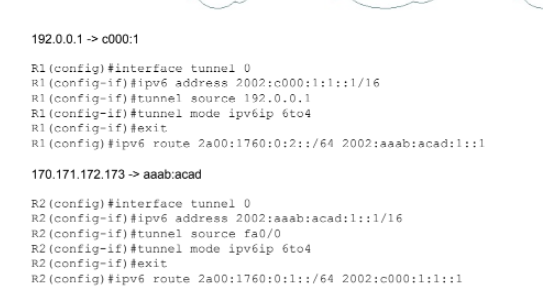
Настройка NAT64: ip6tables -t nat -A PREROUTING -d <IPv6-адрес> -j NAT64

***В IOS (Cisco):***

Активация IPv6: ipv6 unicast-routing

Настройка туннельных протоколов: interface Tunnel0, ipv6 address <IPv6-адрес>/<длина префикса>, tunnel mode <протокол>

Настройка NAT64: ipv6 nat translation prefix <IPv6-префикс> <IPv4-префикс>

**Совместимость IOS:  
**

**71 Таблицы IPv6-маршрутизации в Windows, Linux и IOS**

*Я бы переписал этот вопрос почти полностью:*

В типовую таблицу маршрутизации могут включаться маршруты по умолчанию, к сетевым интерфейсам, к loopback-у, к туннелям IPv6 over IPv4, маршруты связанные с мультикаст-адресами.

Как и в случае с IPv4, при выборе маршрута применяется правило наиболее точного соответствия. В первую очередь выбирается маршрут к сетевому интерфейсу, в последнюю -- маршрут по умолчанию.

Таблица маршрутизации IPv6 в **Windows** напоминает таблицу IPv4 исключая маску подсети, т.к. маски как таковой не существует в IPv6. (таблица на рисунке ниже).

Вывести на экран можно командой **route print**

Включает поля:

If – интерфейс,

Metric – метрика,

Network Destination – маршрут назначения,

Gateway – шлюз.

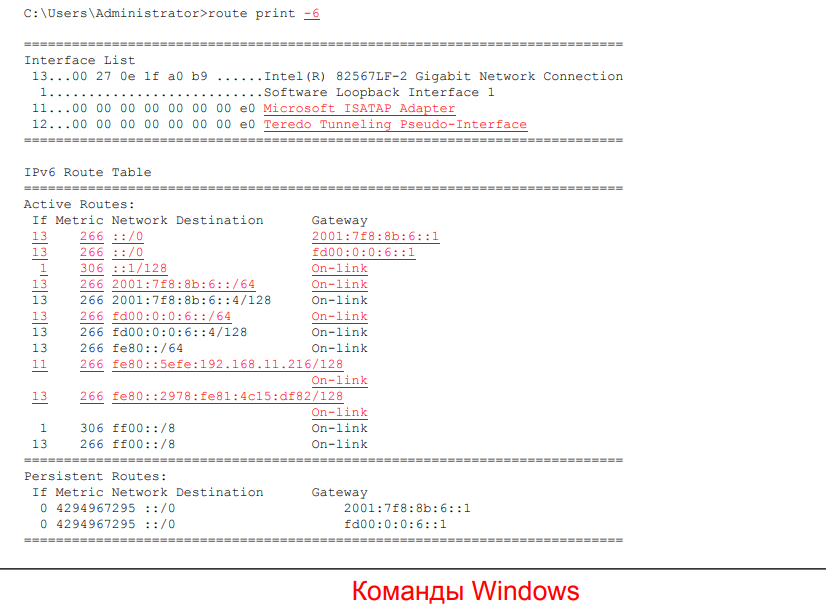


Таблица маршрутизации IPv6 в **Linux** включает в себя поля

Destination,

Next Hop,

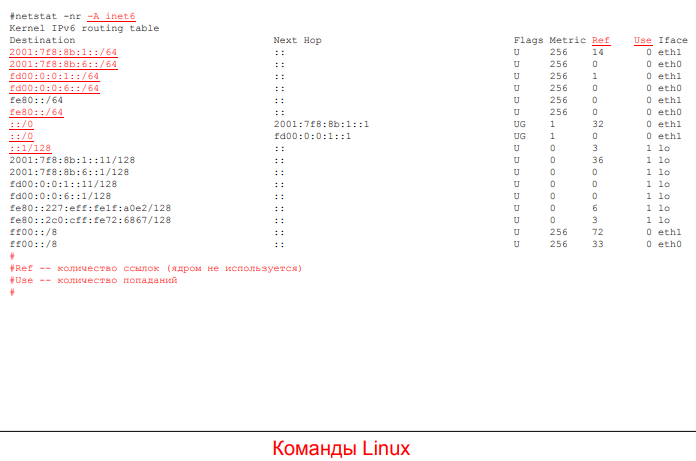
Flags (U-Up, G – используется шлюз),

Metric,

Ref (количество ссылок),

Use (Количество попаданий),

Iface



В IOS таблица маршрутизации выводится командой show ipv route

В самом начале выводится легенда с обозначениями маршрутов (буква – тип маршрута). После выводятся маршруты в виде Тип маршрута - Адрес – интерфейс

**72 Статическая IPv6-маршрутизация в Windows, Linux и IOS**

Маршруты хранятся в таблицах маршрутизации

В типовую таблицу маршрутизации могут включаться маршруты по умолчанию, к сетевым интерфейсам, к loopback-у, к туннелям IPv6 over IPv4, маршруты связанные с мультикаст-адресами.

Основными отличиями являются увеличение количества строк таблицы маршрутизации и изменение набора полей, что вполне адекватно ситуации.

Как и в случае с IPv4, при выборе маршрута применяется правило наиболее точного соответствия. В первую очередь выбирается маршрут к сетевому интерфейсу, в последнюю — маршрут по умолчанию.

Сама маршрутизация:

В Windows:

Добавление статического маршрута в Windows осуществляется с помощью команды "route add" или через интерфейс графической оболочки (Routing and Remote Access). Пример команды: "netsh interface ipv6 add route <destination\_prefix> <next\_hop>"

В Linux:

Добавление статического маршрута в Linux может быть выполнено с помощью команды "ip route add" или через файл конфигурации маршрутизации (/etc/network/interfaces или /etc/sysconfig/network-scripts/route-интерфейс). Пример команды: "ip -6 route add <destination\_prefix> via <next\_hop> dev <interface>"

В IOS (Cisco):

Настройка статической маршрутизации в IOS выполняется с помощью команды "ipv6 route". Пример команды: "ipv6 route <destination\_prefix> <интерфейс><next\_hop>"

**73 Понятие прокси и место прокси в компьютерной сети**

Совокупность инструментальных средств (программного и аппаратного обеспечения), предназначенную для контроля доступа к сетевым ресурсам принято называть прокси.

Если прокси «не виден» для клиентского ПО, то он называется прозрачным.

Задачи, выполняемые прокси:

1. Аутентификация (идентификация/ Аутентификация /авторизация). (вопрос 74)

2. Фильтрация – запрет или разрешение прохождения входящих или исходящих пакетов по выбранным критериям. (75)

3. Сетевой (межсетевой) экран - запрет или разрешение доступа к определенным категориям сетевых ресурсов. (75)

*Сетевой экран в основном выполняет фильтрацию, но это более общее*

*понятие.*

4. Безопасность (или, точнее, кибербезопасность (cybersecurity)

-- в данном контексте) - обеспечение защиты информации, передаваемой по открытым для прослушивания сетям. (76)

5. Ведение журналов -- протоколирование различных событий, связанных с доступом к сетевым ресурсам.

6. Отслеживание угроз. (77)

7. Акселерация -- ускорение доступа к сетевым ресурсам за счет определенных оптимизаций. (80)

Основные способы:

1. Кэширование.

2. Многопоточность.

3. Поддержка «докачки».

8. Формирование трафика -- распределение приоритетов при доступе к сетевым ресурсам по определенным критериям. (80)

9. Преобразование адресов. *(тут про NAT можно подробно рассказать)* (80)

10. Прочие задачи, связанные с преобразованием передаваемой информации и, как правило, не требующие обеспечения конфиденциальности.

*(в целом по каждому из пунктов можно накидать базу из следующих вопросов)*

**74 Аутентификация и ее проявления в компьютерных сетях**

Аутентификация -- определение круга пользователей, имеющих права доступа к сетевым ресурсам.

Если рассматривать более подробно, то в рамках аутентификации, можно выделить:

1. Идентификацию -- назначение пользователям и ресурсам уникальных символьных или цифровых обозначений, то есть имен или идентификаторов в ОС.

2. Аутентификацию -- обеспечение гарантии, что пользователи являются теми, за кого они себя выдают.

3. Авторизацию -- назначение аутентифицированным пользователям прав, что обычно неотделимо от аутентификации.

Аутентификация обычно выполняется по учетной записи, то есть совокупности имени пользователя и пароля.

В общем же случае, может выполняться как более сложно, например, по карте доступа или биометрически, так и по IP-адресу или MAC-адресу.

Аутентификация может проводиться:

1. Локально -- запрос обрабатывается на том же устройстве, которое обеспечивает доступ, или к которому требуется доступ.

2. Удаленно -- запрос перенаправляется на внешний выделенный для этого сервер по специальным протоколам таким как RADIUS (Remote Authentication Dial In User Services) и TACACS+ (Terminal Access Controller Access Control System Plus).

**75 Сетевые экраны и фильтрация трафика**

Фильтрация -- запрет или разрешение прохождения входящих или исходящих пакетов по выбранным критериям.

В качестве объекта фильтрации выступает пакет.

Может выполняться по IP-адресам, по портам, по содержимому и так далее.

Сетевой (межсетевой) экран -- запрет или разрешение доступа к определенным категориям сетевых ресурсов (как правило централизованным или внешним).

Сетевой экран в основном выполняет фильтрацию, но это более общее понятие.

Классификация сетевых экранов:

1. Packet Firewalls -- просто пропускают или отбрасывают пакеты. Работают на третьем уровне (очень редко на втором)

2. Stateful Firewalls -- способны следить за состоянием TCP-соединений, то есть выполнять *инспекцию* трафика. Работают на третьем и четвертом уровнях.

3. Application Gateway Firewalls -- способны следить за сообщениями протоколов прикладного уровня. Работают на третьем -- седьмом уровнях.

**76 Защита информации, передаваемой по открытым сетям**

*Безопасность* (кибербезопасность) -- обеспечение защиты информации, передаваемой по открытым для прослушивания сетям.

Задачи, решаемые в рамках обеспечения безопасности:

1. *Аутентификация* -- в данном контексте, обеспечение гарантии, что сообщение пришло от того, от кого оно ожидается. Как правило, заключается в манипулировании с ключами. Алгоритмы: PSK, RSA и другие.

2. *Целостность* -- обеспечение гарантии, что при пересылке сообщение не было повреждено и не было подменено. Как правило, заключается в подсчете значений хэш-функций. Алгоритмы: MD5 и SHA-256 и другие.

3. *Конфиденциальность* -- обеспечение гарантии, что перехваченное сообщение не может быть прочитано. Как правило, заключается в шифровании данных. Алгоритмы: 3DES, AES и другие.

Как вариант, возможна маскировка конфиденциальных данных под неконфиденциальные.

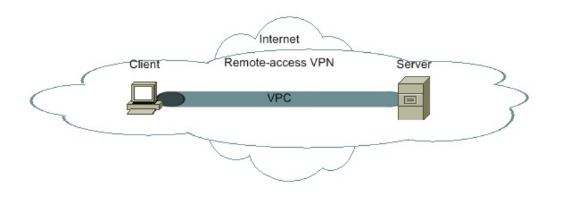
Во многие алгоритмы для формирования доверительных отношений между абонентами заложено использование *цифровых подписей* или *цифровых сертификатов*.

При этом гарантом ответственности может выступать третья сторона, которой доверяют обе взаимодействующие стороны.

Современные тенденции в области безопасности компьютерных сетей сводятся к формированию так называемых *виртуальных частных сетей* -- VPN, охватывающих взаимодействующие станции. При этом взаимодействие осуществляется по формируемым особым образом *виртуальным частным каналам* -- VPC, которые на практике обычно представляют собой защищенные туннели, проложенные через открытые для прослушивания сети.

Все VPN можно разделить на два типа:

1. Site-to-site -- в рядовом случае связывают одноранговые шлюзы и являются статическими.

2. Remote-access -- в рядовом случае обеспечивают подключение удаленных пользователей, создаются динамически и базируются на клиент-серверной модели.

При администрировании, существуют две основополагающие политики обеспечения безопасности:

1. Разрешено всё, что не запрещено.

2. Запрещено все, что не разрешено.

В настоящее время наиболее оправданным признан второй вариант.

**77 Отслеживание и подавление угроз в компьютерных сетях**

Если сделать упор на реакцию при возникновении угроз, то можно выделить два типа прокси:

1. IDSes (Intrusion **Detection** Systems) -- своеобразные сенсоры, которые отслеживают вредоносный трафик по сигнатурам и различными способами **оповещают** **при его обнаружении**. Обычно трафик через них не «пропускается», а копируется в их сторону для параллельного анализа.

2. IPSes (Intrusion **Prevention** Systems) -- не просто отслеживают вредоносный трафик, а и **способны самостоятельно его заблокировать**. Обычно трафик «пропускается» через них.

Существует много видов и способов атак, но также есть и достаточное количество способов защиты от них.

Основные из них:

1 Reconnaissance Attacks -- разведывательные, целью которых является несанкционированный сбор информации.

2 Access Attacks -- связанные с доступом, целью которых является получение несанкционированного доступа к информации или подмена информации.

3 DoS (Denial of Service) Attacks -- связанные с сервисами, целью которых является отказ в обслуживании по тому или иному протоколу.

DDoS (Distributed DoS) отличается тем, атаку проводят множество станций.

При работе в Интернете рекомендуется выполнять следующие требования:

 - Пользуйтесь паролями

 - Работайте на компьютере под учетной записью с ограниченными правами

 - Используйте шифрование данных

 - Регулярно выполняйте обновления программного обеспечения

- Используйте и регулярно обновляйте антивирусные программы

 - Используйте межсетевой экран

**78 Примеры вредоносных атак в компьютерных сетях**

Cisco выделяет три типа вредоносных атак:

1 Reconnaissance Attacks -- разведывательные, целью которых является несанкционированный сбор информации.

**Примеры**: просмотр содержимого пакетов сниферами, сканирование адресов в поисках станций, сканирование портов в поисках сервисов, ловля на доверие, социальная инженерия, поиск информации в Internet.

2 Access Attacks -- связанные с доступом, целью которых является получение несанкционированного доступа к информации или подмена информации.

**Примеры**: подбор паролей методом «грубой силы», использование имеющихся прав не по назначению, перенаправление пользовательских запросов на ложные серверы, различные варианты подмены информации в каналах, использование уязвимостей ПО.

3 DoS (Denial of Service) Attacks -- связанные с сервисами, целью которых является отказ в обслуживании по тому или иному протоколу.

DDos (Distributed DoS) отличается тем, атаку проводят множество станций.

**Примеры**: ping с длиной пакета 65535 Byte с целью «завешивания» некоторых старых ОС, порождение с помощью особенностей SNMP-запросов многочисленных станций-«зомби» с целью «забрасывания» SNMP-ответами выбранной станции-«жертвы», последовательное создание многочисленных полуоткрытых TCP-соединений.

**79 Примеры злоумышленников и вредоносных программ в**

**компьютерных сетях**

Cisco выделяет следующие типы компьютерных преступников:

1 Hackers -- наиболее общий термин, но характерной чертой является наличие знаний в области компьютерных технологий.

2 Black Hats -- злоумышленники, которые могут и не обладать больши'ми знаниями.

3 White Hats -- выполняют несанкционированные атаки, но из благих побуждений (например, сообщают администратору об обнаруженных проблемах).

4 Crackers (взломщики) -- специализируются на взломе защиты КС, или ПО, или еще чего-либо.

5 Spammers -- массово рассылают электронную почту.

6 War Drivers -- путешествуют в поисках «халявы» (обычно незащищенных беспроводных сетей).

7 Phishers -- пытаются под различными предлогами «выудить» конфиденциальную информацию.

8 Phrickers -- используют особенности телефонных сетей для совершения преступлений.

Cisco выделяет три типа вредоносных программ:

1. Viruses -- наиболее общий термин, но характерной чертой является распространение с помощью внедрения вредоносного кода в пользовательские программы.

2. Worms -- характерной чертой является самостоятельное распространение посредством СПД, протекающее в три фазы: поиск или создание известных вирусу заранее уязвимостей, внедрение путем копирования через сеть, вредоносное проявление.

3. Trojan horses -- характерной чертой является маскировка под «безобидные» программы.

**80 Задачи прокси, непосредственно не связанные с безопасностью**

*Акселерация* -- ускорение доступа к сетевым ресурсам за счет определенных оптимизаций.

Основные способы:

1 Кэширование.

2 Многопоточность.

3 Поддержка «докачки».

*Формирование трафика* -- распределение приоритетов при доступе к сетевым ресурсам по определенным критериям.

Может быть программным или аппаратным, статическим или динамическим. Может осуществляться по разным критериям, например, по времени.

*Преобразование адресов*.

Особую проблему при организации коллективного доступа в Internet представляет собой «невидимость» внутренних адресов.

Первоначально задачу можно сформулировать так: требуется, чтобы несколько пользовательских станций из внутренней подсети могли совместно пользоваться одним реальным адресом.

NAT (Network Address Translation) -- наиболее общий стандарт, решающий задачу путем прозрачной подмены адресов на маршрутизаторах.

Для обеспечения правильности выполнения преобразований строится таблица, то есть NAT работает по табличному принципу.

Все реализации NAT, в первую очередь, делят на два типа:

1. Статические (static) -- преобразования осуществляется исходя из

строгого соответствия пар адресов.

2. Динамические (dynamic) -- при преобразованиях адреса' по мере

надобности выделяются из пулов по определенным критериям.

Однако есть дополнение в виде портовой реализации PAT (Port Address Translation). PAT-дополнение совместимо и со статическим, и с динамическим вариантами NAT.

Также можно упомянуть, прозрачное сжатие данных, балансировку нагрузки и вполне легальное перенаправление прикладных сервисов.

**81 NAT и другие манипуляции адресами**

Особую проблему при организации коллективного доступа в Internet представляет собой «невидимость» внутренних адресов.

Первоначально задачу можно сформулировать так: требуется, чтобы несколько пользовательских станций из внутренней подсети могли совместно пользоваться одним реальным адресом.

NAT (Network Address Translation) -- наиболее общий стандарт, решающий задачу путем прозрачной подмены адресов на маршрутизаторах.

Обобщенный алгоритм работы IP NAT:

1 Клиент передает пакет прокси с поддержкой NAT.

2 Прокси запоминает IP-адрес назначения, IP-адрес источника, подставляет в качестве IP-адреса источника свой адрес и передает пакет серверу, запрашиваемому клиентом.

3. После получения ответного пакета от сервера выполняются обратные преобразования на основе запомненной информации.

4 Ответный пакет передается клиенту.

Для обеспечения правильности выполнения преобразований строится таблица, то есть NAT **работает по табличному принципу**.

Все реализации NAT, в первую очередь, делятся на два типа:

1 Статические -- преобразования осуществляется исходя из строгого соответствия пар адресов.

1. Динамические -- при преобразованиях адреса по мере надобности выделяются из пула по определенному критерию.

+ (3) Однако есть дополнение в виде портовой реализации PAT (Port Address Translation). PAT-дополнение совместимо и со статическим, и с динамическим вариантами NAT.

 Первоначальная постановка задачи предполагает, что подменяется IP-адрес источника, но возможна подмена IP-адреса назначения либо обоих адресов.

Первоначальная постановка задачи так же предполагает, что внутренний IP-адрес замещается реальным, но, в общем случае, возможна произвольная комбинация.

Наконец, первоначальная постановка задачи предполагает, что запросы порождаются клиентами во внутренней сети, но, поскольку «правильная» NAT-таблица работает в двух направлениях открыта возможность обслуживания запросов со стороны внешних клиентов.

Так, статический вариант NAT позволяет разместить во внутренней сети сервер и адресовать его из Internet.

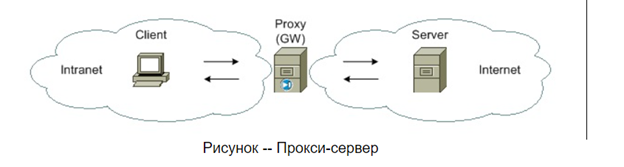
Более того, статический вариант PAT позволяет перенаправлять запросы из Internet об определенных сервисах на соответствующие отдельные внутренние серверы.

Все варианты NAT совместимы с туннелированием.

**NAT полностью противоречит логике IPv6, поэтому, касательно IPv6, его поддержка не рекомендуется.**

**82 Пример взаимодействия через прокси**

Типичное место расположения прокси -- это граница между внутренней сетью и сетью публичного доступа.



Две основные проблемы прокси: при использовании следует избегать каскадирования большого числа прокси, а также перегрузки ядра OC.

Касательно протоколов прикладного уровня, подавляющее большинство прокси -- это HTTP-прокси, причем двух вариантов:

1 Get Method.

2 Connect Method.

На втором месте находятся прокси электронной почты. Все остальные встречаются крайне редко.

Обобщенная последовательность действий при взаимодействии клиента и сервера на примере HTTP-прокси.

Пример взаимодействия через прокси может быть следующим:

Клиент отправляет запрос на подключение к определенному веб-серверу.

Запрос попадает на прокси-сервер, который настроен на прослушивание и перенаправление соединений.

Прокси-сервер принимает запрос от клиента и проверяет свои правила и настройки для обработки этого запроса.

Прокси-сервер устанавливает соединение с веб-сервером от имени клиента и передает ему запрос.

Веб-сервер получает запрос от прокси-сервера и обрабатывает его, генерируя соответствующий ответ.

Ответ от веб-сервера передается обратно прокси-серверу.

Прокси-сервер получает ответ и применяет свои правила и настройки для обработки ответа.

Прокси-сервер передает ответ обратно клиенту, который инициировал запрос

**83 Классификация инструментальных средств, реализующих прокси**

Все задачи, выполняемые прокси, могут решаться самыми разными устройствами на различных уровнях модели OSI. Все способы можно объединить в три направления:

1. Host-based (Server-based + Personal) – сугубо программные решения на базе Windows и Linux.

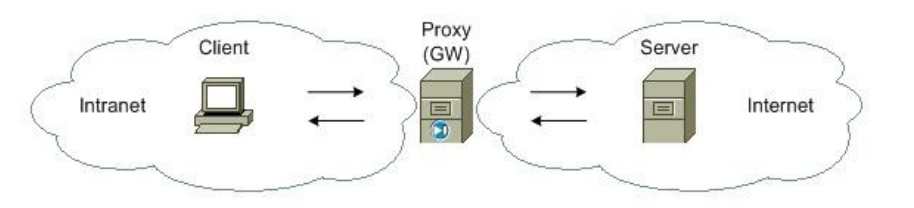
Здесь можно сразу выделить два уровня, на которых можно решать упомянутые задачи: ядро ОС и прикладное ПО.

2. Integrated — программно-аппаратные решения на базе специализированных сетевых OC, таких как IOS и Junos OS, которые «крутятся» на маршрутизаторах и другом сетевом оборудовании.

3. Appliance-based — полностью специализированные аппаратные решения, такие как Cisco ASA и SafeNet eSafe.

Такие устройства «сосредоточены» на обеспечении безопасности и называются аппаратными сетевыми экранами.

Типичное место расположения прокси — это граница между внутренней сетью и сетью публичного доступа.

****

**84 Фильтрация и NAT в Windows**

Фильтрация (filtering) -- запрет или разрешение прохождения входящих или исходящих пакетов по выбранным критериям.

В качестве объекта фильтрации выступает пакет.

Может выполняться по IP-адресам, по портам, по содержимому и так далее.

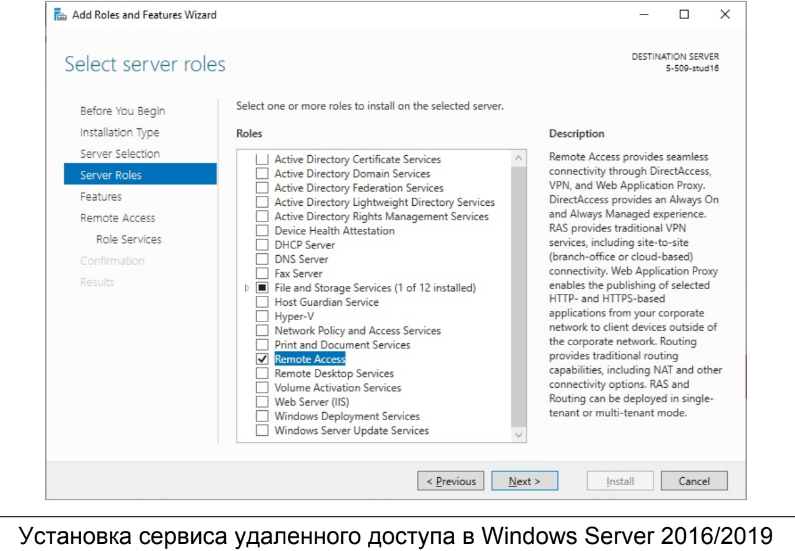
Фильтры строятся на основе правил (rules). Каждое правило -- это строка, содержащая в себе условия, определяющие подпадает ли пакет под правило, и действие, которое необходимо осуществить в случае выполнения условий. Правила могут объединяться в цепочки (chains) и образовывать сложную иерархию.

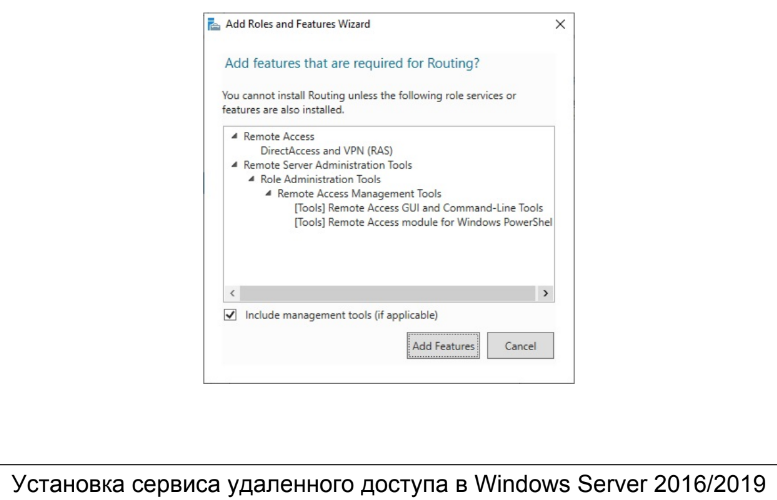
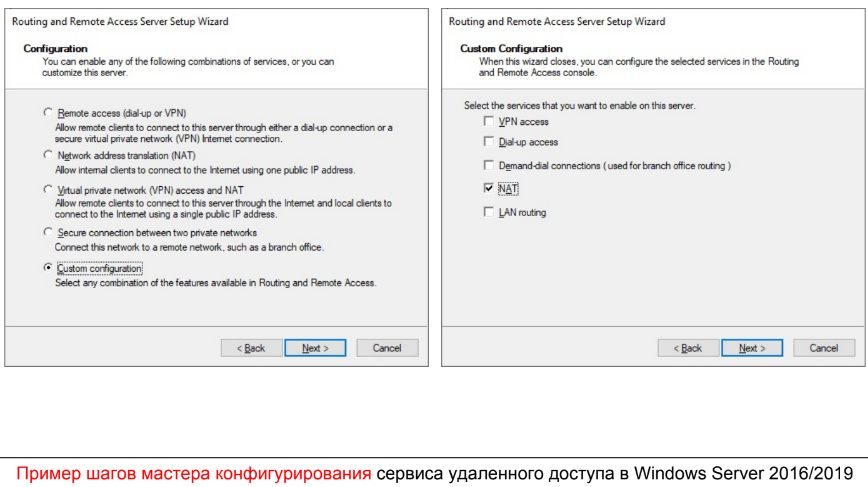
Полноценная поддержка NAT с графическим интерфейсом доступна в серверных редакциях -- в составе Routing and Remote Access.

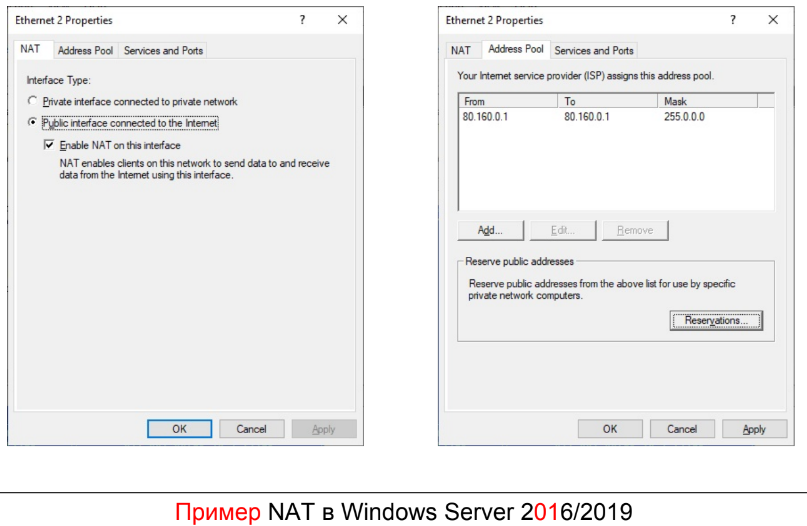
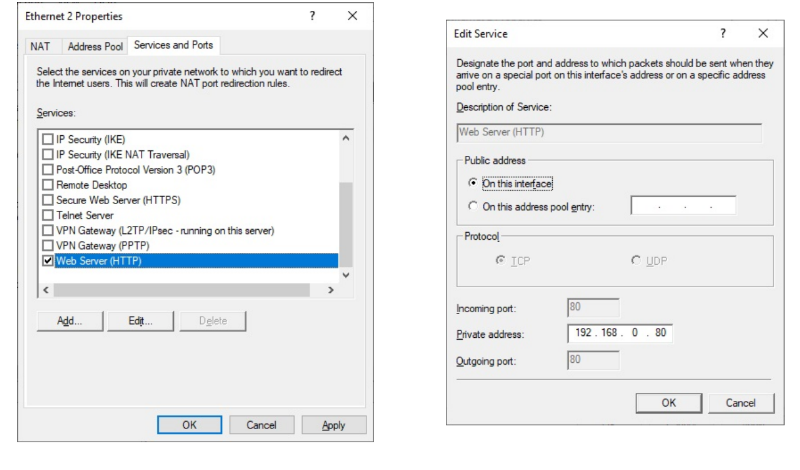
В Server 2003 R2 был интегрированный компонент Routing and Remote Access,

в Server 2008 R2 -- опциональная роль с тем же названием.

Начиная с Server 2012, роль имеет название Remote Access.

Для конфигурирования используют оснастку Routing and Remote Access (rrasmgmt.msc).



****

*\*Если простыми словами, то есть такая роль (так называется функционал) под названием Remote Access, которую нужно типо скачать. После чего нужно сконфигурировать (настроить ее). Конкретно для NAT можно перейти в раздел настроек через Custom Configuration поставив галочку рядом с ним. Далее нужно провести настройку NAT.*

**85 Пакет IP Tables**

В большинстве систем UNIX широко применяется пакет IP Filter -- в основном для целей **фильтрации и NAT.** (в линукс)

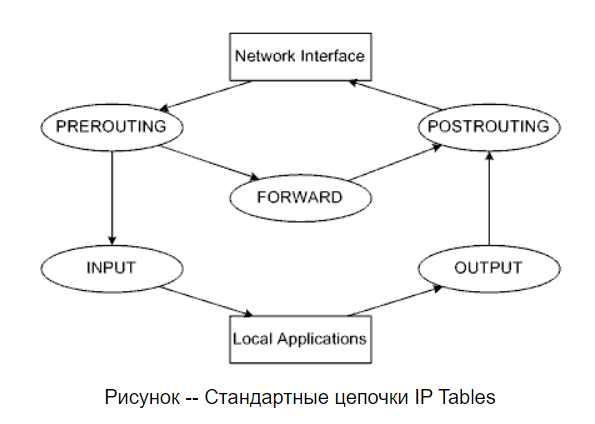
В Linux эту роль выполняет пакет IP Tables (пришел на смену Ipfwadm и IP Chains).

Для нормальной работы IP Tables должны быть включены некоторые опции ядра.

Для обеспечения возможности управления введен одноименный **сервис** **iptables**.

Фильтры строятся на основе правил (rules). Каждое правило -- это строка, содержащая в себе условия, определяющие подпадает ли пакет под правило, и действие, которое необходимо осуществить в случае выполнения условий.

Правила могут объединяться в цепочки (chains) и образовывать сложную иерархию.

Следовательно, при работе с IP Tables необходимо внимательно проверять содержимое и последовательность правил.

Примеры таблиц (**tables**):

filter -- нужна для фильтрации пакетов;

mangle -- нужна для внесения изменений в заголовки пакетов (например,

в поле TTL);

nat -- нужна для преобразования адресов.

Примеры команд (**commands**):

-A (--append) -- добавить новое правило в конец цепочки;

-D (--delete) -- удалить правило из цепочки;

-F (--flush) -- удалить все правила из цепочки;

-I (--insert) -- вставить новое правило в цепочку;

-L (--list) -- вывести на экран список правил в цепочке;

-N (--new-chain) -- создать новую цепочку с названием в таблице;

-P (--policy) -- определить политику по умолчанию для цепочки;

-R (--replace) -- заменить одно правило другим в цепочке;

-X (--delete-chain) -- удалить цепочку из таблицы.

Примеры действий (**targets**):

ACCEPT -- пакет прекращает движение по цепочке (и всем цепочкам, приведшим к текущей) и считается пропущенным, но он может быть отвергнут следующими цепочками;

DNAT -- подмена адреса назначения;

DROP -- пакет отвергается (окончательно);

LOG -- протоколирование пакета или связанных с его прохождением событий;

MASQUERADE -- подмена адреса источника без явного указания заменяющего адреса;

REJECT -- равно DROP плюс посылка ответного ICMP-сообщения о недостижимости;

SNAT -- подмена адреса источника.

Переходы (jumps) позволяют передавать пакет другим цепочкам.

GPT

IPTables - это инструмент в Linux для настройки фильтрации пакетов и управления межсетевым экраном (firewall). Он позволяет администраторам системы устанавливать правила для контроля трафика, определять, какие пакеты пропускать или блокировать, а также выполнять преобразование сетевых адресов (NAT - Network Address Translation).

IPTables работает на уровне ядра операционной системы и обрабатывает пакеты, проходящие через сетевой стек. Он может использоваться для обеспечения безопасности сети, фильтрации трафика, настройки портов перенаправления, маскарадинга (SNAT - Source Network Address Translation), проброса портов (Port Forwarding) и других задач.

С помощью IPTables администратор может создавать и управлять цепочками правил, определять условия для применения правил (например, исходный и целевой IP-адрес, порт и протокол) и задавать действия, которые должны выполняться для соответствующих пакетов (например, пропуск, блокировка, перенаправление).

IPTables является мощным инструментом для управления сетевой безопасностью и конфигурации сетевых правил в Linux. Он широко используется администраторами систем для защиты серверов и сетей от нежелательного трафика и атак извне.

**86 Полноценные прокси на базе Windows и Linux**

*Windows*

Начиная с Windows XP SP2 в ядро интегрирован Windows Firewall, позиционируемый как базовый персональный сетевой экран.

Упрощенными для удобства пользователя специфическими формами NAT являются Network Bridge Connection и Internet Connection Sharing («вытесняют» сервис Routing and Remote Access).

**Network Bridge Connection** позволяет объединить два возможно разнородных сегмента с целью эмуляции одного сегмента.

**Internet Connection Sharing** позволяет нескольким пользователям из одной подсети разделять (совместно использовать в режиме разделения времени) один сетевой интерфейс из другой подсети, обычно с целью доступа к Internet.

Полноценная поддержка NAT с графическим интерфейсом доступна в Server 2003 и Server 2008 в составе RRAS (**Routing and Remote Access Service**).

В случае масштабного применения для серверов Windows предлагается стандартный пакет **ISA** (Internet Security and Acceleration) **Server**, позиционируемый как кэширующий прокси-сервер с возможностями сетевого экрана.

Кроме того, очень широко применяются пакеты сторонних производителей, среди которых следует выделить Qbik WinGate и Kerio KerioControl.

*Linux*

В большинстве систем UNIX широко применяется пакет IP Filter -- в основном для целей фильтрации и NAT. В Linux эту роль выполняет пакет IP Tables.

Для нормальной работы IP Tables должны быть включены некоторые опции ядра.

Для обеспечения возможности управления введен одноименный сервис iptables.

IPTables - это инструмент в Linux для настройки фильтрации пакетов и управления межсетевым экраном (firewall). Он позволяет администраторам системы устанавливать правила для контроля трафика, определять, какие пакеты пропускать или блокировать, а также выполнять преобразование сетевых адресов (NAT - Network Address Translation).

IPTables работает на уровне ядра операционной системы и обрабатывает пакеты, проходящие через сетевой стек. Он может использоваться для обеспечения безопасности сети, фильтрации трафика, настройки портов перенаправления, маскарадинга (SNAT - Source Network Address Translation), проброса портов (Port Forwarding) и других задач.

С помощью IPTables администратор может создавать и управлять цепочками правил, определять условия для применения правил (например, исходный и целевой IP-адрес, порт и протокол) и задавать действия, которые должны выполняться для соответствующих пакетов (например, пропуск, блокировка, перенаправление).

IPTables является мощным инструментом для управления сетевой безопасностью и конфигурации сетевых правил в Linux. Он широко используется администраторами систем для защиты серверов и сетей от нежелательного трафика и атак извне.

**87 Поддержка NAT в IOS**

 ВОПРОС ПЕРЕПИСАН ПОЛНОСТЬЮ

Cisco IOS поддерживает все теоретические варианты NAT.

Статические и динамические преобразования совместимы (даже в одном направлении), но статические нужно конфигурировать раньше динамических.

Отдельно взятое правило преобразований задают командой **ip nat**.

Обязательно нужно «привязать» сетевые интерфейсы к внутренней сети (inside) и сети публичного доступа (outside) командами **ip nat inside** и **ip nat outside** соответственно (можно и не по одному).

Важно правильно понимать Cisco-терминологию, связанную с адресацией при NAT-преобразованиях:

1. Inside local -- адрес расположенной в сети inside станции как он

«виден» в сети inside.

2. Inside global -- адрес расположенной в inside-сети станции как он

«виден» в сети outside.

3. Outside local -- адрес расположенной в сети outside станции как он

«виден» в сети inside.

4. Outside global -- адрес расположенной в сети outside станции как он

«виден» в сети outside.

Cisco-варианты NAT:

1. ip nat inside source static -- в направлении из сети inside в сеть outside

подменяется IP-адрес источника (и IP-адрес назначения в обратном направлении), то есть адрес inside local заменяется адресом inside global (классический статический source NAT).

2. ip nat inside source list -- замены аналогично ip nat inside sourcestatic, плюс позволяет задействовать список адресов inside local и пул адресовinside global (динамический source NAT).

4. ip nat outside source static -- в направлении из сети outside в сеть inside подменяется IP-адрес источника, то есть адрес outside global заменяется адресом outside local (статический destination NAT, используется редко).

5. ip nat outside source list -- замены аналогично ip nat outsidesource static, плюс позволяет задействовать список адресов outside global и пул адресов outside local (динамический destination NAT, используется редко).

Для просмотра NAT-таблицы используют команду **show ip nat translations**, для просмотра статистики -- **show ip nat statistics**.