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).

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).

//////

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 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).

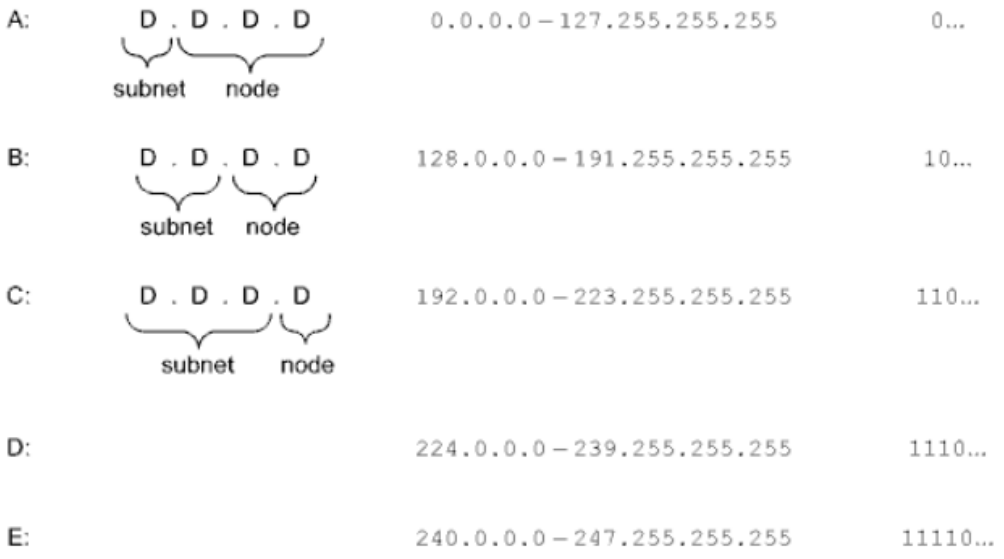
Сетевой интерфейс (network interface) -- это минимально адресуемый в СПД компонент, входящий в состав какой-либо станции. Применительно к компьютерам, как правило, сетевой интерфейс физически выражен в виде сетевого адаптера -- Network Interface Card (NIC).

Станция 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.

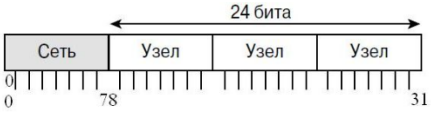
Каждый сетевой интерфейс должен иметь собственный IP-адрес. Если некоторая станция содержит два либо более сетевых интерфейсов, то среди них выделяется главный, ассоциированный с самой станцией. Обычно главный интерфейс «смотрит» в сторону Internet.

6 Классы IPv4-адресов

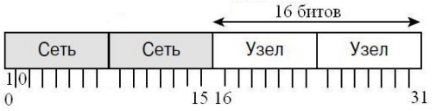
Формально выделяют пять классов IP-адресов. Классы A, B и C являются основными, а классы D и E -- дополнительными. Класс D используется для адресации мультикаст -групп. Класс E зарезервирован для будущего использования. Есть и другие зарезервированные диапазоны.



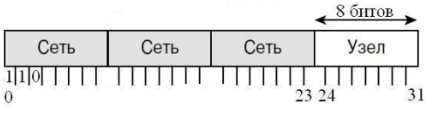
У адресов класса A старший бит установлен 0. Длина сетевого префикса - 8 бит. Для номера узла выделяется 3 байта (24 бита).



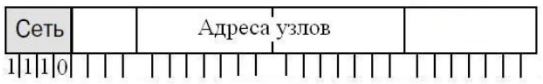
У адресов класса B два старших бита установлены в 1 и 0 соответственно. Длина сетевого префикса - 16 бит. Поле номера узла тоже имеет длину 16 бит. Класс B предназначен для применения в сетях среднего размера.



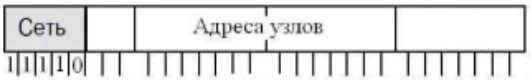
У адресов класса C три старших бита установлены в 1, 1 и 0 соответственно. Префикс сети имеет длину 24 бита, номер узла - 8 бит. Класс C предназначен для сетей с небольшим количеством узлов.



Адреса класса D представляют собой специальные адреса, не относящиеся к отдельным сетям. Первые 4 бита этих адресов равны 1110. Таким образом, значение первого октета этого диапазона адресов находится в пределах от 224 до 239. Адреса класса D используются для многоадресатных пакетов, с помощью которых во многих разных протоколах данные передаются многочисленным группам хостов. Эти адреса можно рассматривать как заранее запрограммированные в логической структуре большинства сетевых устройств.



Адреса в диапазоне 240.0.0.0 - 255.255.255.255 называются адресами класса E. Первый октет этих адресов начинается с битов 1111. Эти адреса зарезервированы для будущих дополнений в схеме адресации IP. Но возможность того, что эти дополнения когда-либо будут приняты, находится под вопросом, поскольку уже появилась версия 6 протокола IP (IPv6).



7 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

8 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).

9 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, станция 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.

10 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, на который пакет отправляется в том случае, если маршрут к сети назначения пакета не известен (не задан явным образом в таблице маршрутизации хоста)[1]. Применяется в сетях с хорошо выраженными центральными маршрутизаторами, 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».

11 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.

12 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.

13 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.

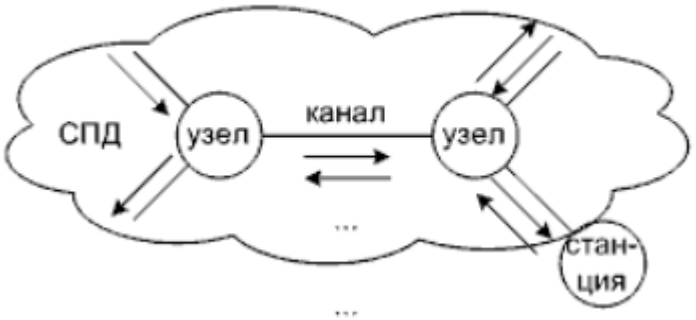
```
Router>enable
Router#configure terminal
Router(config)#interface gi0/0
Router(config-if)#ip address 192.168.11.1 255.255.255.224 !Обязательно
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#exit
Router#disable
Router>
```

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.

14 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.

15 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.

16 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.

	Destination	Netmask	Gateway	Interface	Metric	Options
Route						
Route						
...						
Route						

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. Маска подсети 255.255.255.255 -- маршрут к одному сетевому интерфейсу.

17 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.

18 Структура Internet

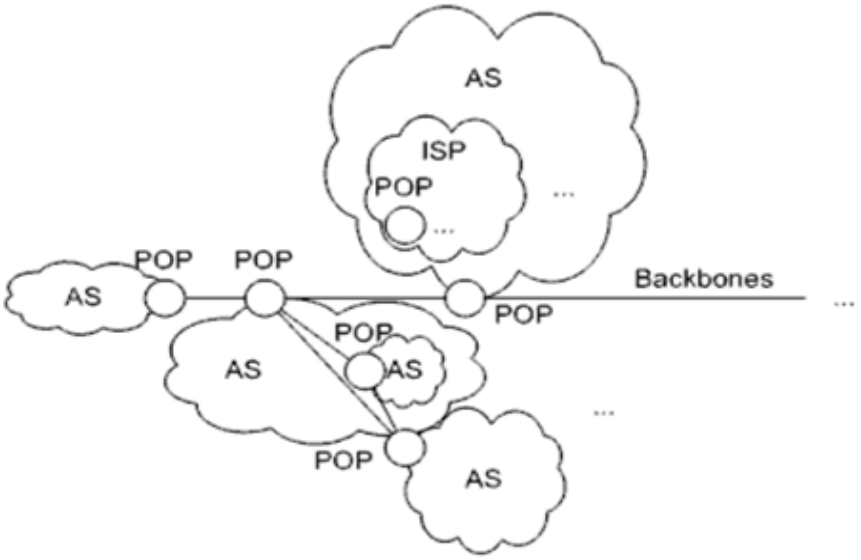


Рисунок -- Структура Internet

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.

19 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.

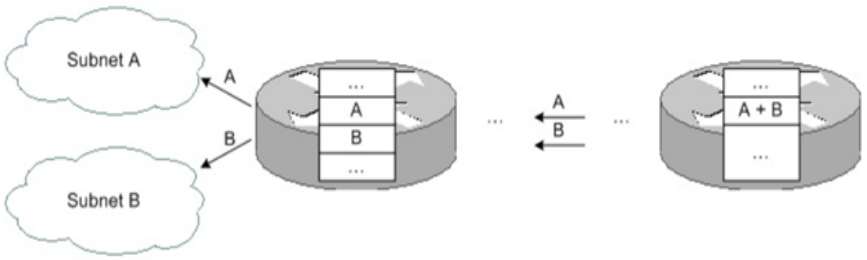


Рисунок -- Агрегация маршрутов

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).

Таблица -- Классификация основных протоколов динамической маршрутизации

	IGP		EGP	
	DVA	LSA	DVA	LSA
IPv4 Classful	RIPv1 ~~IGRP (Cisco)~~	--	--	~~EGP~~
IPv4 Classless	RIPv2 EIGRP (Cisco)	OSPFv2 IS-IS	BGPv4	--
IPv6	RIPng EIGRP for IPv6 (Cisco)	OSPFv3 IS-IS for IPv6	BGPv4+	--

(Практически выведенные из эксплуатации протоколы зачеркнуты.)
(Касательно проприетарных протоколов в скобках указана компания-разработчик.)

Где: RIP -- Routing Information Protocol, EGP -- Exterior Gateway Protocol, OSPF -- Open Shortest Path First, BGP -- Border Gateway Protocol, (E)IGRP -- (Enhanced) Interior Gateway Routing Protocol, IS-IS -- Intermediate System to Intermediate System routing protocol.

20 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.

21 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.

2.0.9.29

```
C:\Users\Administrator>arp -a ;Вывести на экран ARP-таблицу

Interface: 192.168.11.214 --- 0xb
  Internet Address      Physical Address      Type
  192.168.11.193        b8-38-61-81-10-ca     dynamic
  192.168.11.223        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\Administrator>arp -s 10.0.0.10 00-AA-00-4F-2A-9C

C:\Users\Administrator>netsh -c interface ipv4 add neighbors "Local Area
Connection" "192.168.10.10" "00-1d-71-83-6c-00" store=persistent
```

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.

22 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 MAC-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.

23 Практические особенности IPv4-маршрутизации

15 и 17 вопрос

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.

24 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.

```
IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.11.193    192.168.11.221     266
        127.0.0.0        255.0.0.0         On-link         127.0.0.1     306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1     306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1     306
    192.168.11.192  255.255.255.224         On-link    192.168.11.221     266
    192.168.11.221  255.255.255.255         On-link    192.168.11.221     266
    192.168.11.223  255.255.255.255         On-link    192.168.11.221     266
        224.0.0.0        240.0.0.0         On-link         127.0.0.1     306
        224.0.0.0        240.0.0.0         On-link    192.168.11.221     266
  255.255.255.255  255.255.255.255         On-link         127.0.0.1     306
  255.255.255.255  255.255.255.255         On-link    192.168.11.221     266
===========================================================================
Persistent Routes:
  Network Address          Netmask  Gateway Address  Metric
          0.0.0.0          0.0.0.0    192.168.11.193  Default
===========================================================================
...
```

Пример таблицы маршрутизации Windows на пользовательской станции

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.

25 Структура таблицы IPv4-маршрутизации в Linux

2.0.9.4

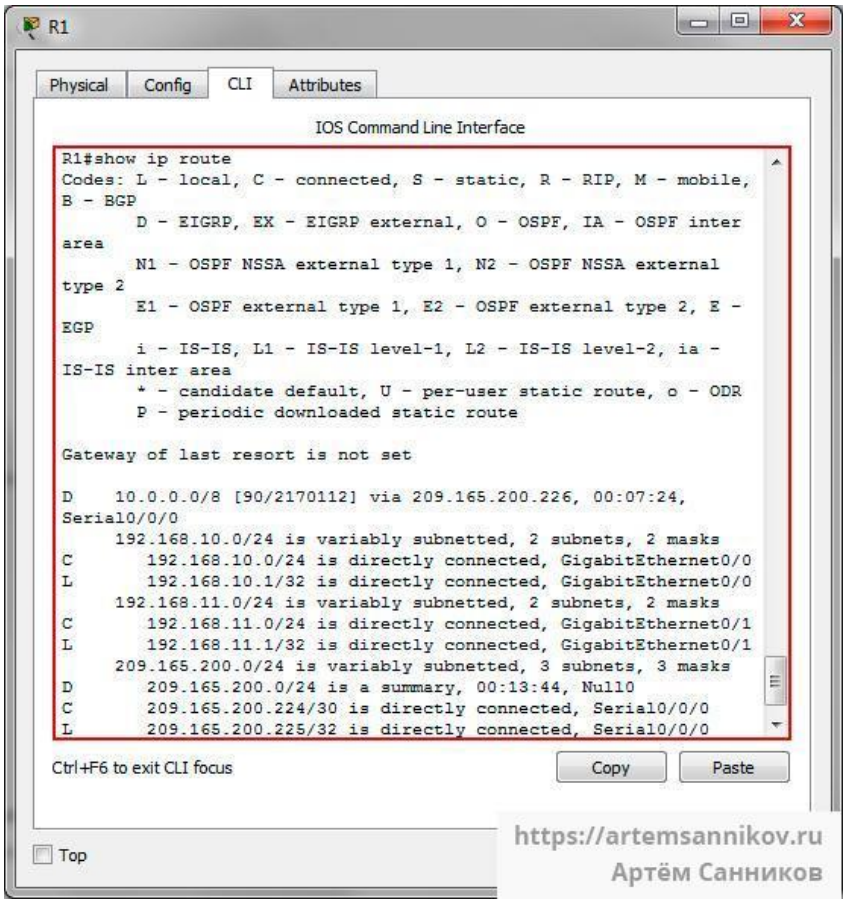
```
#netstat -r #Вывести на экран таблицу маршрутизации ядра

Kenel IP routing table
Destination     Gateway         Genmask         Flags    MSS window   irtt Iface
192.168.11.160  *               255.255.255.240 U          0 0          0 eth0
192.168.11.0    *               255.255.255.128 U          0 0          0 eth1
169.254.0.0     *               255.255.0.0     U          0 0          0 eth1
127.0.0.0       *               255.0.0.0       U          0 0          0 lo
default         192.168.11.1    0.0.0.0         UG         0 0          0 eth1
#
#Флаги: U -- route is Up, G -- use Gateway
#MSS, window, irtt -- параметры TCP (устарело)
#

#netstat -nr #Адреса отображать в цифровой форме (не делать DNS-запросы)

Destination     Gateway         Genmask         Flags    MSS window   irtt Iface
192.168.11.160  0.0.0.0         255.255.255.240 U          0 0          0 eth0
192.168.11.0    0.0.0.0         255.255.255.128 U          0 0          0 eth1
169.254.0.0     0.0.0.0         255.255.0.0     U          0 0          0 eth1
127.0.0.0       0.0.0.0         255.0.0.0       U          0 0          0 lo
0.0.0.0         192.168.11.1    0.0.0.0         UG         0 0          0 eth1
```

Пример таблицы маршрутизации Linux на шлюзе

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
- Windows - TCP-окно
- irtt - время отклика для TCP-соединений
- Всё остальное - устаревшие параметры TCP.

26 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 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.

27 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.

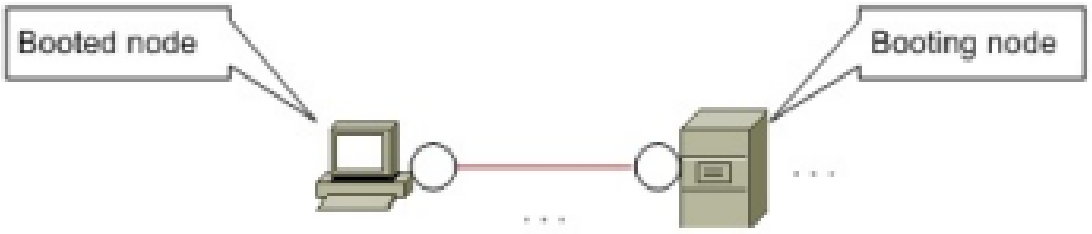
```
Windows: route add 192.168.11.160 mask 255.255.255.240 192.168.11.50
Linux: route add -net 192.168.11.160 netmask 255.255.255.240 gw 192.168.11.50
```

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.

28 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.

29 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, HP 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).

30 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.

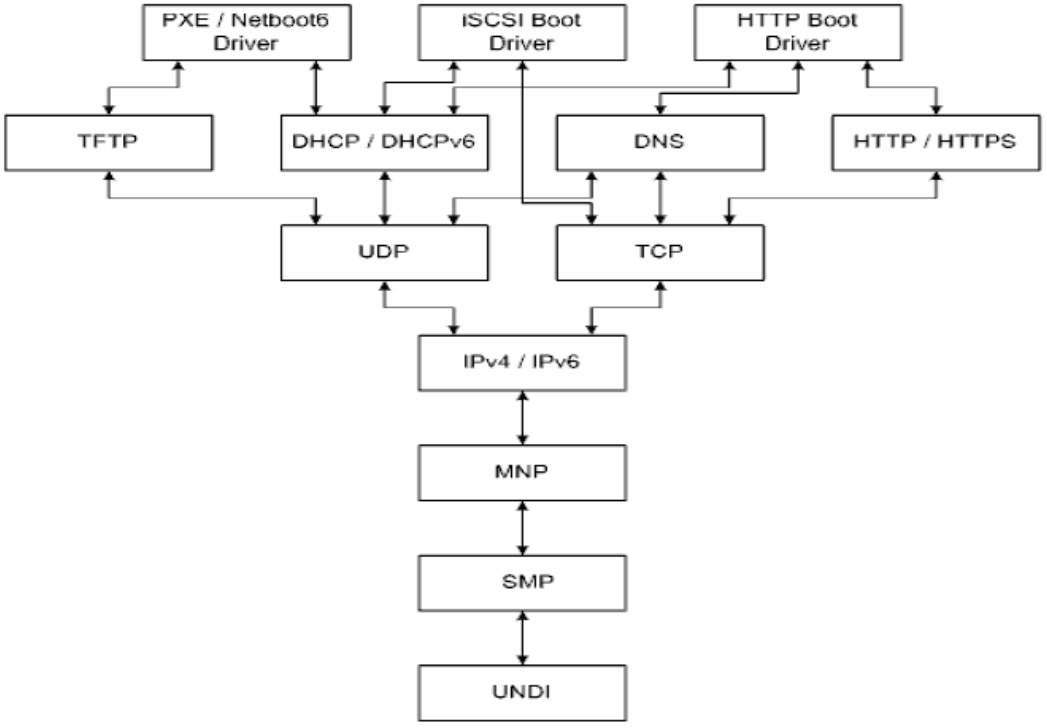
После включения загружаемой станции выполняется так называемый POST (Power On Self Test). При этом BIOS сканирует память в диапазоне C0000h -- EE000h (куда отображаются add-on BIOSes) с инкрементом, равным 2 kByte, в поисках сигнатуры 55AAh, которая свидетельствует о наличии add-on BIOS. Если сигнатура 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.

31 Поддержка удаленной загрузки в UEFI

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

Для обеспечения удаленной загрузки от производителей сетевых контроллеров 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.

//**Если есть возможность**



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

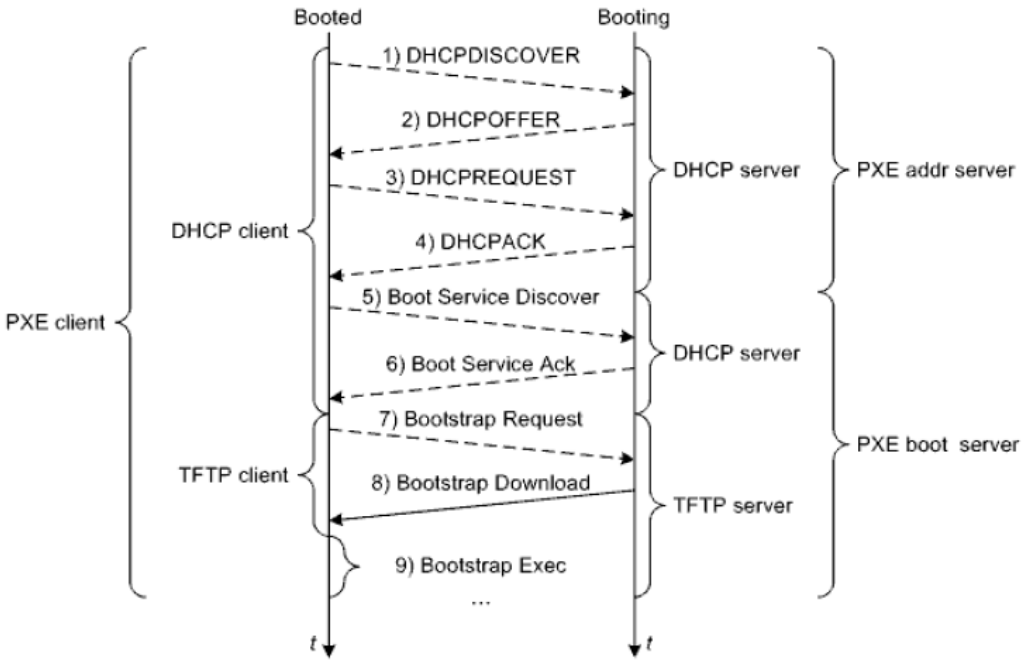
32 Взаимодействие по протоколу PXE

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.



Клиент-серверное взаимодействие по протоколу PXE

1. DHCPDISCOVER -- DHCP-клиент в составе PXE-клиента посылает броадкаст-запрос с целями анонсирования своего«возникновения» и поиска сервиса адресации, коим является DHCP-сервер. Стандартный порт на стороне DHCP-сервера: UDP 67.

При этом ключевыми являются опции: 97 -- UUID/GUID-based Client Identifier (Universally/Globally Unique ID), 93 -- Client System Architecture, 94 -- Client Network Device Interface, 60 -- Class Identifier (в случае с PXE значение начинается с PXEClient).
Если DHCP-клиент по каким
-либо причинам еще не готов полноценно
обрабатывать юникаст
-пакеты, он должен установить флаг B -- Broadcast
Flag, так как форма ответной адресации выбирается исходя из значения этого флага

33 Протоколы BOOTP, DHCP, TFTP и их использование

Первым протоколом, который массово использовали для динамического назначения IP-адресов, является BOOTP. Как альтернативу BOOTP, для нахождения IP-адресов по MAC-адресам, изредка использовали протокол RARP – в современных реализациях практически не поддерживается. DHCP представляет собой расширение BOOTP.

По большому счету, в DHCP-заголовке передается только пара конфигурационных параметров, в первую очередь, IP-адрес. Остальные параметры передаются в виде DHCP-опций. Тип DHCP-сообщения определяется из значения опции 53 – DHCP Message Type. 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.

Остальные типы имеют отношение к опциональному расширению DHCP, позволяющему сторонней станции (не клиенту и не серверу) запрашивать информацию о DHCP-конфигурации. По истечении времени валидности IP-адрес обновляется посредством целенаправленных (юникаст) DHCP_REQUEST и DHCP_ACK.

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

34 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.

35 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.

36 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. Маска подсети 255.255.255.255 -- маршрут к одному сетевому интерфейсу

37 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 X -- 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

38 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. 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.

39 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.

10 bits	54 bits	64 bits
FE80	0 … 0	Interface ID

Рисунок -- Формат адреса вида Link-local Unicast

Как и в других юникаст-адресах, имеется четкое разделение на топологическую и интерфейсную части.

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

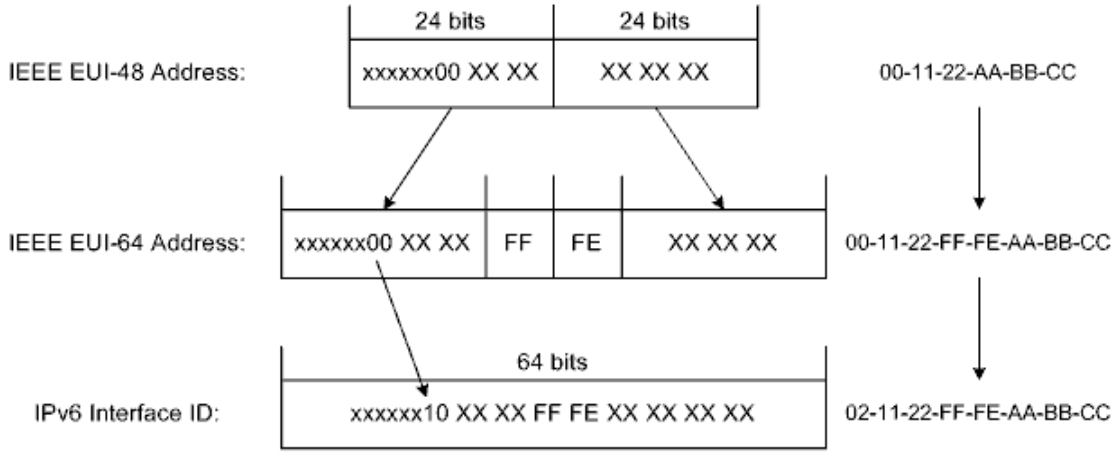


Рисунок -- Алгоритм и пример вычисления значения поля IPv6 Interface ID

В результате, интерфейсная часть соответствует нотации EUI-64 (точнее, модифицированной нотации EUI-64). От приведенного правила можно отступать, но это не рекомендуется.

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). Пакеты с адресами Unique Local Unicast должны подавляться всеми маршрутизаторами кроме внутренних.

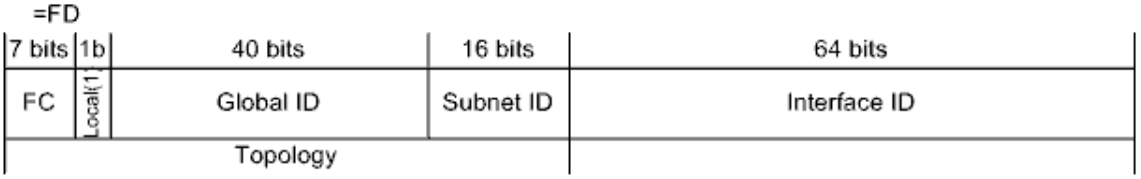


Рисунок -- Формат адреса вида Unique Local Unicast

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

40 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).

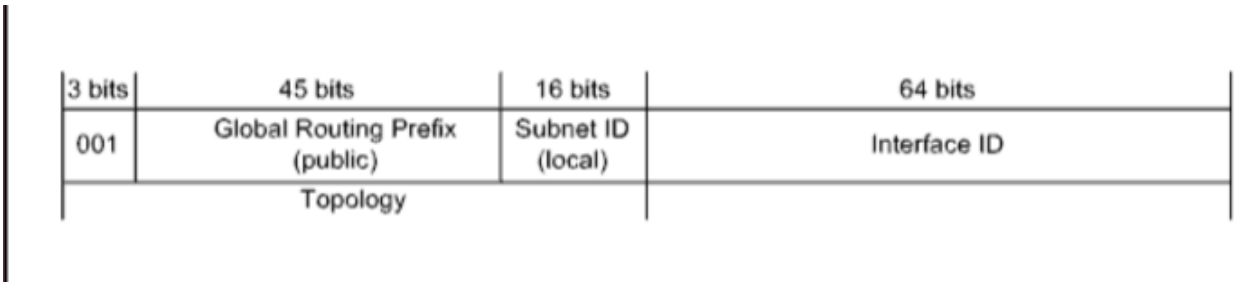


Рисунок -- Формат адреса вида Global Unicast

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.

41 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.

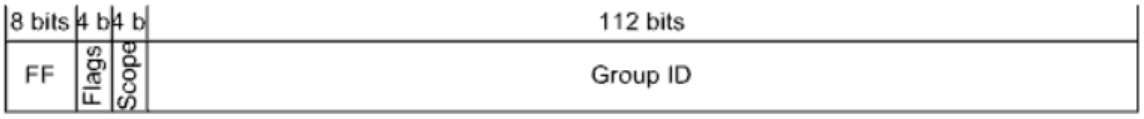


Рисунок -- Формат адреса типа Multicast

Применительно к линку, в качестве замены широковещательных адресов IPv4 позиционируются адреса вида Link-local Scope All-nodes Multicast.

Кроме того, при автоконфигурировании в пределах линка используются специальные адреса вида Solicited-node Multicast, строящиеся на основе адресов Link-local Unicast, из которых переносятся последние 24 бита.

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.

42 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.

Рисунок -- Один из форматов адреса типа Anycast

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.

Рисунок -- Формат адреса вида 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.

43 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.

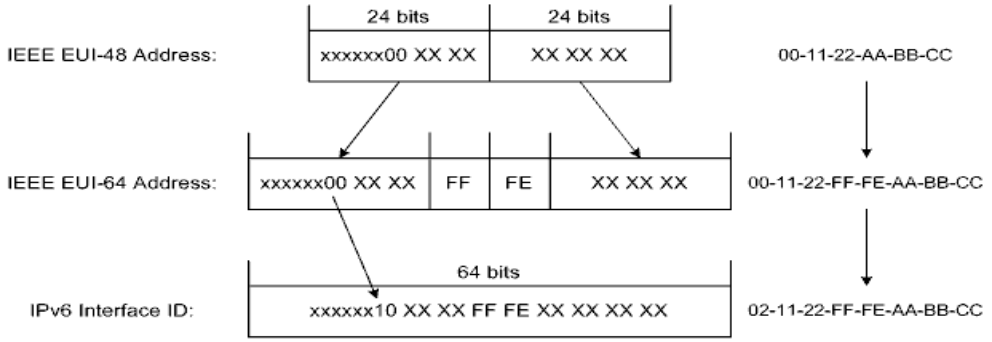


Рисунок -- Алгоритм и пример вычисления значения поля IPv6 Interface ID

В результате, интерфейсная часть соответствует нотации EUI-64 (точнее, модифицированной нотации EUI-64). От приведенного правила можно отступать, но это не рекомендуется.

44 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. Таковые туннели делят на три типа:

1. Host-to-host.
2. Host-to-router и router-to-host.
3. Router-to-router.

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

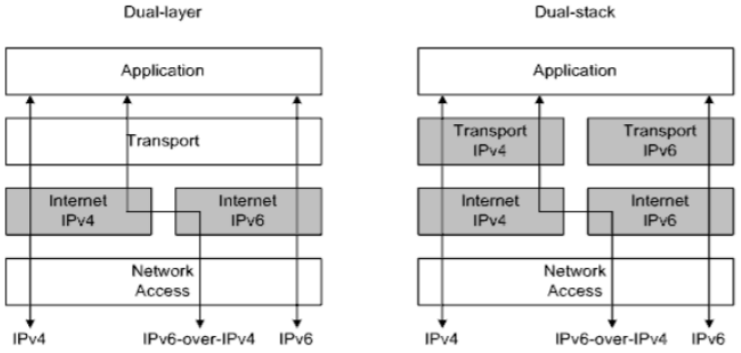
16 bits	32 bits	16 bits	64 bits
2002	Hex IPV4	Subnet ID	Interface ID
	Topology		

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.



45 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.

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.

46 Обнаружение маршрутизаторов и оптимизация маршрутов при IPv6-автоконфигурировании

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.
2. Восстановление значений префиксов подсетей.
3. Восстановление значений некоторых других параметров.
4. Автоконфигурирование адресов.
5. Восстановление MAC-адресов соседних станций.
6. Обнаружение маршрутизаторов следующего звена.
7. Проверка достижимости соседних станций.
8. Проверка конфликтов адресов.
9. Оптимизация маршрутов.

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.

Важно, что задачи ND решают именно в пределах линка. ND -- это механизм, вполне допускающий конфигурирование. Многие параметры могут быть заданы.

ND нельзя рассматривать как альтернативу динамической маршрутизации. ND работает в рамках линка и, по понятным причинам, на ND не возлагают обязанности автоматического нахождения маршрутов к внешним подсетям. А вот автоматически назначать маршрут по умолчанию ND может. Более того, при автоконфигурировании все соседние маршрутизаторы автоматически рассматриваются как кандидаты в маршрутизаторы по умолчанию -- создается специальный список.

47 Восстановление параметров при IPv6-автоконфигурировании

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

Отдельный префикс подсети анонсируется маршрутизатором в виде отдельной 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; включая флаги: L -- данный префикс подсети относится к текущему линку; A -- данный префикс подсети может быть использован для генерирования адресов.

В RA вкладывается столько ND-опций, сколько нужно. Анонсируются все префиксы подсетей из привязанного к сетевому интерфейсу списка AdvPrefixList. Существует настоятельная рекомендация о том, что на маршрутизаторе в этот список по умолчанию вносятся префиксы всех подсетей, к которым относится сетевой интерфейс, исключая префиксы подсетей Link-local Unicast. При необходимости, список может быть дополнен «вручную».

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

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

Первым таковым параметром является Cur Hop Count. Значение будет вписываться в поле Hop Limit заголовка IPv6 каждого передаваемого маршрутизатору пакета.

Вторым параметром является MTU. В линках с вариативным MTU, например, Ethernet, маршрутизатор обязан указывать (ND-опция).

48 Автоконфигурирование адресов и их жизненный цикл при IPv6-автоконфигурировании

В контексте SLAAC 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. При этом воспринимаются только префиксы подсети длиной 64 бита. Если маршрутизаторов несколько, то воспринимаются префиксы от всех маршрутизаторов. Автоконфигурирование позиционируют прежде всего в отношении хостов, однако и сетевые интерфейсы маршрутизаторов могут быть подвержены автоконфигурированию. При этом соответствующие префиксы подсетей в RAs не включаются.

SLAAC и DHCPv6 вполне совместимы друг с другом. «Разделение обязанностей» контролируется двумя флагами в RA: M -- адреса доступны посредством DHCPv6; O -- другие параметры доступны посредством DHCPv6 (если флаг M установлен, то флаг O игнорируется).

Автоконфигурирование адресов подразумевает и автоматическое нахождение маршрутизатора по умолчанию. Адреса DNS-серверов автоматически могут быть получены только посредством DHCPv6.

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.

49 Восстановление адресов при IPv6-автоконфигурировании и проверка конфликтов адресов

Задача NUD (Neighbor Unreachability Detection) является закономерным «продолжением» задачи восстановления MAC-адресов и так же решается использованием связки NS (но не с мультикаст-, а с юникаст-адресом назначения) и NA.

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 (именно так).

5. PROBE -- идет собственно проверка достижимости соседа.

Одна проверка достижимости соседа, как и одно восстановление MACадреса подразумевает несколько попыток (согласно стандарту по умолчанию три и три попытки соответственно).

Проверка конфликтов адресов является 8й из 9 задач для решения в границах линка

Не смотря на всю гибкость IPv6, проверку конфликта адресов никто не отменял -- исключая эникаст-адреса. Задача DAD (Duplicate Address Detection) решается передачей специальным образом наполненного NS (с нулевым IPv6-адресом источника) и проверкой есть ли ответ.

Для решения последней задачи используется специальное сообщение Redirect.

50 Проверка достижимости при IPv6-автоконфигурировании

Задача NUD (Neighbor Unreachability Detection) решается использованием связки NS (но не с мультикаст-, а с юникаст-адресом назначения) и NA.

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.

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.

Алгоритм проверки достижимости опирается на два основных таймера:

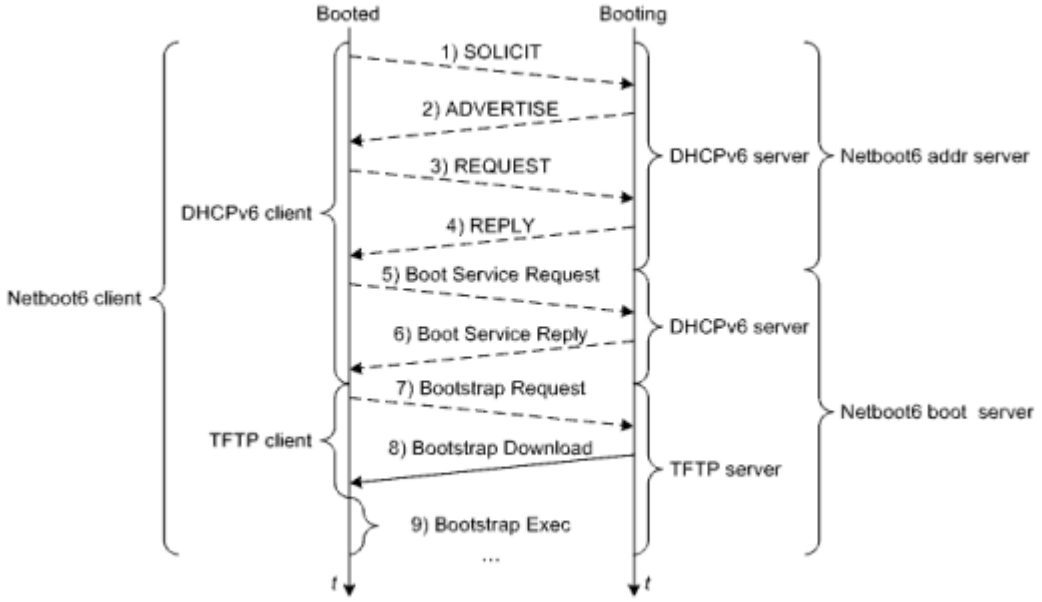
Reachable Time -- интервал времени после приема последнего сообщения NA от соседа, в течение которого этот сосед считается достижимым

Retrans Timer -- интервал между передачей NSes при переходе к следующей попытке.

Маршрутизаторы могут предлагать значения этих таймеров в отношении линка анонсируя RAs с ненулевыми значениями одноименных полей.

51 Взаимодействие по протоколу Netboot6

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



Где:

1.Клиент рассылает сообщение SOLICIT (по идее, по IPC) на стандартный порт DHCP6 (547). Оно содержит следующее:

• Тег версии UNDI клиента.

• Тег для архитектуры клиентской системы.

• Тег для PXE-клиента, данные Vendor Class установлены

• PXEClient:Arch:xxxxx:UNDI:yyyzzz

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

2.Служба DHCP6 или прокси-сервер DHCP6 отвечает отправкой сообщения ADVERTISE на клиент на стандартном порте ответа DHCP6 (546). Если это служба Proxy DHCP6, следующий адрес сервера (загрузочного сервера) доставляется опцией URL-адреса загрузочного файла. Если это DHCP6 служба, новый назначенный адрес клиента доставляется опцией IA.

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

5. Клиент рассылает мультикаст REQUEST на порт Boot Server 4011, которое содержит (по большому счету, Хто я, как в п.1, только подробнее)

• Тег версии UNDI клиента.

• Тег для архитектуры клиентской системы

. • Тег для PXE-клиента, параметр Vendor Class

• «PXEClient:Arch:xxxxx:UNDI:yyyzzz».

6. Сервер отправляет Unicast сообщение REPLY клиенту на его порт, которое содержит инфу о файлах, с которых грузиться: Тэг имени файла, и , если надо, его размер.

7. Клиент по TFTP (69 порт) запрашивает файл.

8. Сервер отдает файл, клиент загружает его в память.

52 Протоколы DHCPv6 и его использование

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

DHCPv6 использует UDP номер порта 546 для клиентов и номер порта 547 для серверов. По сути, такой же DHCP, как и в V4.

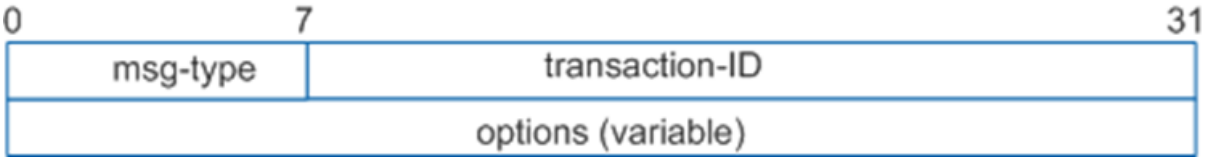
DHCPv6 делится на следующие два типа:

•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.). (Выдает 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.

Формат сообщения DHCPv6:



msg-type - 1байт - Указывает тип сообщения в диапазоне от 1 до 13.

transaction-ID - 3 байта - Идентификатор взаимодействия DHCPv6, также называемый идентификатором транзакции, используется для идентификации обмена сообщениями DHCPv6 в обоих направлениях

Options - Указывает поле параметра DHCPv6. Это поле содержит информацию о конфигурации, назначенную сервером DHCPv6 хосту IPv6, например IPv6-адрес DNS-сервера.
//////////////////////////////////////Мб кратко//////////

Также имеется ряд опций (65536)

Одни и те же DHCPv6-опции могут передаваться в обоих направлениях. В отличие от DHCPv4-опций, DHCPv6-опции имеют сложный формат с вариативным количеством полей и подопций. DHCPv6-клиент не обязан выполнять «предписания» DHCPv6-сервера, даже сам может «высказывать пожелания» о значениях некоторых параметров DHCPv6-серверу. DHCPv6-клиент и DHCPv6-сервер должны иметь уникальные идентификаторы, по которым они однозначно опознают друг друга.

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.

53 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.

54 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.

55 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

56 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.

Генерирование временных адресов:

```
>netsh interface ipv6 set privacy=enabled|disabled
```

Генерирование случайных значений интерфейсных частей постоянных адресов
(в отличие от других основных ОС, включено по умолчанию):

```
>netsh interface ipv6 set global randomizeidentifiers=enabled|disabled
```

Автоконфигурирование адресов:

```
>netsh interface ipv6 set interface Interface_Name_Or_Index routerdiscovery=enabled|disabled|dhcp
```

57 Статическая и динамическая IPv6-адресация в Linux

В Linux поддержка IPv6 имеется в дистрибутивах с ядрами 2.2.x и последующими. Присвоение адресов IPv6 сводится к работе с соответствующими конфигурационными файлами.

• Примеры IPv6-дополнений в конфигурационных файлах Linux:
/etc/sysconfig/network: ...

/etc/sysconfig/network-scripts/ifcfg-eth1 (ветви Red Hat и SUSE)

/etc/network/interfaces (ветвь Debian)

• Примеры управления IPv6-автоконфигурированием в Linux:

Генерирование временных адресов:
#sysctl net.ipv6.conf.default.use_tempaddr=integer

Автоконфигурирование, включая ND: конфигурационный файл /etc/sysconfig/network-scripts/ifcfg-:

демон radvd со стандартным конфигурационным файлом /etc/radvd.conf

• Примеры управления совместимостью с IPv4 в Linux:

6to4: конфигурационный файл /etc/sysconfig/network-scripts/ifcfg-:
ISATAP,
Teredo: пакет Miredo, предоставляющий одноименный сервис, со стандартным конфигурационным файлом /etc/miredo.conf

Обновленные команды для проверки информации и корректной работы ipv6 в Linux:

• ifconfig;
• ping6 ;
• ip -6 neigh show.

Генерирование временных адресов:

```
#sysctl net.ipv6.conf.default.use_tempaddr=integer
```

либо

```
#echo "integer" > /proc/sys/net/ipv6/conf/default/use_tempaddr
```

где `integer`:
<= 0 -- запрет
= 1 -- разрешение, причем временные адреса менее приоритетны
> 1 -- разрешение, причем временные адреса более приоритетны

Автоконфигурирование, включая ND:

конфигурационный файл `/etc/sysconfig/network-scripts/ifcfg-<interface-name>`:

```
...
IPV6_AUTOCONF=yes|no
IPV6_ROUTER=yes|no
...
```

демон `radvd` со стандартным конфигурационным файлом `/etc/radvd.conf`

58 Статическая и динамическая IPv6-адресация в IOS

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

59 Поддержка совместимости IPv6 с IPv4 в Windows, Linux и IOS

Примеры управления совместимостью с IPv4 в Windows:

Ключ реестра:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\TCPIP6\Parameters\DisabledComponents
```

где `DisabledComponents` (`DWORD`) формируется из битов:
бит 0 = 1 -- запрет всех туннельных интерфейсов IPv6-over-IPv4
бит 1 = 1 -- запрет туннельного интерфейса 6to4
бит 2 = 1 -- запрет туннельного интерфейса ISATAP
бит 3 = 1 -- запрет туннельного интерфейса Teredo
бит 4 = 1 -- разрешение IPv6 только посредством туннельных интерфейсов IPv6-over-IPv4
бит 5 = 1 -- IPv4 предпочтительнее IPv6

Варианты команды `netsh interface ipv6`:

```
>netsh interface ipv6 6to4 set state state=enabled|disabled|default
>netsh interface ipv6 isatap set state state=enabled|disabled|default
>netsh interface ipv6 set teredo type=disabled|client|enterpriseclient|server|default
```

Примеры управления совместимостью с IPv4 в Linux:

Возможности `radvd`

6to4:

конфигурационный файл `/etc/sysconfig/network-scripts/ifcfg-<interface-name>`:

```
IPV6TO4INIT=yes
IPV6TO4_ROUTING="eth0-:1::1/64"
IPV6_CONTROL_RADVD=yes
```

ISATAP:

```
#ip tunnel add is0 mode isatap local 192.168.11.216
#ip addr add fd00::6:0:5efe:192.168.11.216/64 dev is0
#ip link set is0 up
```

Teredo:

пакет Miredo, предоставляющий одноименный сервис, со стандартным конфигурационным файлом `/etc/miredo.conf`

Совместимость IOS:

192.0.0.1 -> c000:1

```
R1(config)#interface tunnel 0
R1(config-if)#ipv6 address 2002:c000:1:1::1/16
R1(config-if)#tunnel source 192.0.0.1
R1(config-if)#tunnel mode ipv6ip 6to4
R1(config-if)#exit
R1(config)#ipv6 route 2a00:1760:0:2::/64 2002:aaab:acad:1::1
```

170.171.172.173 -> aaab:acad

```
R2(config)#interface tunnel 0
R2(config-if)#ipv6 address 2002:aaab:acad:1::1/16
R2(config-if)#tunnel source fa0/0
R2(config-if)#tunnel mode ipv6ip 6to4
R2(config-if)#exit
R2(config)#ipv6 route 2a00:1760:0:1::/64 2002:c000:1:1::1
```

60 Таблицы IPv6-маршрутизации в Windows, Linux и IOS

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

В типовую таблицу маршрутизации включаются следующие маршруты:

1. К своим подсетям размером с линк (для всех адресов Link-local Unicast, Unique Local Unicast, Global Unicast).
2. К своим сетевым интерфейсам (для всех адресов Link-local Unicast, Unique Local Unicast, Global Unicast).
3. Маршрут по умолчанию.
4. Маршрут к сетевому интерфейсу -- заглушке.
5. Маршруты, связанные с адресами Multicast.
6. Дополнительные статические и динамические маршруты.
7. Маршруты к туннелям IPv6-over-IPv4.

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.

```
C:\Users\Administrator>route print -6

===========================================================================
Interface List
 13...00 27 0e 1f a0 b9 ......Intel(R) 82567LF-2 Gigabit Network Connection
  1...........................Software Loopback Interface 1
 11...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter
 12...00 00 00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
===========================================================================

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
 13    266 ::/0                     2001:7f8:8b:6::1
 13    266 ::/0                     fd00:0:0:6::1
  1    306 ::1/128                  On-link
 13    266 2001:7f8:8b:6::/64       On-link
 13    266 2001:7f8:8b:6::4/128     On-link
 13    266 fd00:0:0:6::/64          On-link
 13    266 fd00:0:0:6::4/128        On-link
 13    266 fe80::/64                On-link
 11    266 fe80::5efe:192.168.11.216/128
                                    On-link
 13    266 fe80::2978:fe81:4c15:df82/128
                                    On-link
  1    306 ff00::/8                 On-link
 13    266 ff00::/8                 On-link
===========================================================================
Persistent Routes:
 If Metric Network Destination      Gateway
  0 4294967295 ::/0                     2001:7f8:8b:6::1
  0 4294967295 ::/0                     fd00:0:0:6::1
===========================================================================
```

```
#netstat -nr -A inet6
Kernel IPv6 routing table
Destination                    Next Hop                   Flags Metric Ref    Use Iface
2001:7f8:8b:1::/64             ::                         U     256    14       0 eth1
2001:7f8:8b:6::/64             ::                         U     256    0        0 eth0
fd00:0:0:1::/64                ::                         U     256    1        0 eth1
fd00:0:0:6::/64                ::                         U     256    0        0 eth0
fe80::/64                      ::                         U     256    0        0 eth1
fe80::/64                      ::                         U     256    0        0 eth0
::/0                           2001:7f8:8b:1::1           UG    1      32       0 eth1
::/0                           fd00:0:0:1::1              UG    1      0        0 eth1
::1/128                        ::                         U     0      3        1 lo
2001:7f8:8b:1::11/128          ::                         U     0      36       1 lo
2001:7f8:8b:6::1/128           ::                         U     0      0        1 lo
fd00:0:0:1::11/128             ::                         U     0      0        1 lo
fd00:0:0:6::1/128              ::                         U     0      0        1 lo
fe80::227:eff:fe1f:a0e2/128    ::                         U     0      6        1 lo
fe80::2c0:cff:fe72:6867/128    ::                         U     0      3        1 lo
ff00::/8                       ::                         U     256    72       0 eth1
ff00::/8                       ::                         U     256    33       0 eth0
#
#Ref -- количество ссылок (ядром не используется)
#Use -- количество попаданий
#
```

Команды Linux

61 Статическая IPv6-маршрутизация в Windows, Linux и IOS

Основными отличиями являются увеличение количества строк таблицы маршрутизации и изменение набора полей, что вполне адекватно ситуации. В типовую таблицу маршрутизации включаются следующие маршруты:

1. К соседним подсетям размером с линк (для всех адресов Link Local Unicast, Unique Local Unicast, Global Unicast).
2. К собственным сетевым интерфейсам (для всех адресов Link Local Unicast, Unique Local Unicast, Global Unicast).
3. Маршрут по умолчанию.
4. Маршрут к сетевому интерфейсу — заглушке.
5. Маршруты, связанные с мультикаст-адресами.
6. Дополнительные статические и динамические маршруты.
7. Маршруты к туннелям IPv6-over-IPv4.

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.

Некоторые новые и обновленные команды: route print -6 (Windows), netsh interface ipv6 show route (Windows), netstat -nr -A inet6 (Linux), netsh interface ipv6 add route (Windows), route -A inet6 add (Linux), tracert (Windows), traceroute6 (Linux). Включение IPv6 Forwarding в Windows и Linux:

Windows: >netsh interface ipv6 set interface interface=InterfaceNameOrIndex forwarding=enabled либо сервис Routing and Remote Access

Linux: конфигурационный файл /etc/sysconfig/network:

...

IPV6FORWARDING=yes

...

либо

#echo "1" > /proc/sys/net/ipv6/conf/all/forwarding

Следует обратить внимание на возможность «привязки» не ко всем сетевым интерфейсам, а к конкретным.

IOS

Для просмотра таблицы IPv6-маршрутизации используют команду show ipv6 route. Для внесения статического маршрута в таблицу

62 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).

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

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

4. Безопасность - обеспечение защиты информации, передаваемой по открытым для прослушивания сетям.

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

6. Отслеживание угроз.

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

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

9. Преобразование адресов.

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

63 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.

64 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.

65 Защита информации, передаваемой по открытым сетям

Безопасность -- 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.

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

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

Как вариант, возможна маскировка конфиденциальных данных под неконфиденциальные, выражающаяся в различных алгоритмах *стеганографии*.

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

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.

66 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. 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

Можно сказать что-то из следующих вопросов.

67 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.

68 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.

69 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.

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

70 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.

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

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.

71 Пример взаимодействия через прокси

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

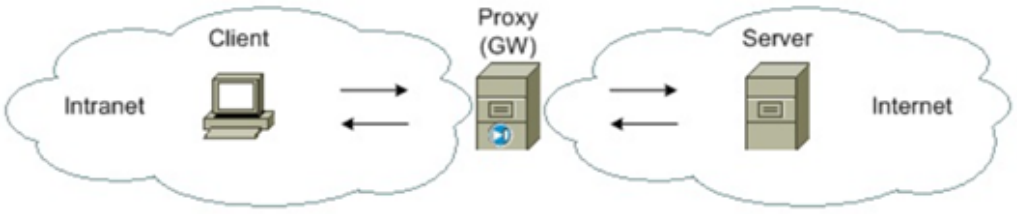


Рисунок -- Прокси-сервер

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

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.

72 Классификация инструментальных средств, реализующих прокси

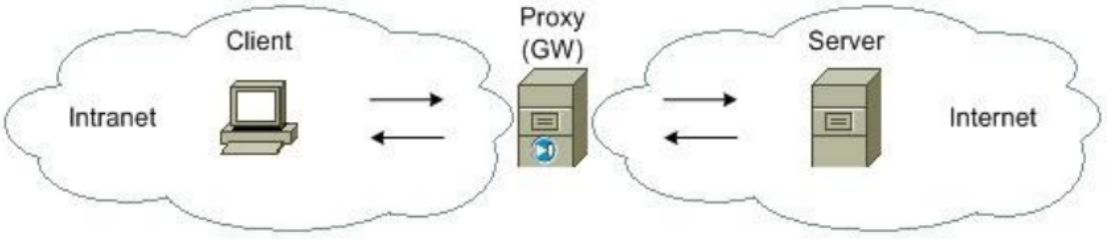
Все задачи, выполняемые прокси, могут решаться самыми разными устройствами на различных уровнях модели OSI. Все способы можно объединить в три направления:

1. Host-based (Server-based + Personal) – сугубо программные решения на базе универсальных серверных и настольных ОС, таких как Windows и Linux. Эти ОС, в свою очередь, «крутятся» на обычных серверах и пользовательских станциях. 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 ОС, таких как IOS и Junos OS, которые «крутятся» на маршрутизаторах и другом сетевом оборудовании. Эти ОС хоть и специализированы как сетевые, но в своей области универсальны, то есть не полностью адаптированы к упомянутым задачам. Степень адаптации повышается за счет специальных аппаратных модулей.

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 (security appliances). Аппаратные сетевые экраны, которые «заточены» для контроля доступа по протоколам прикладного уровня, известны как NACs (Network Admission Controls). Примером может служить Cisco Ironport.

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



73 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.

(Инфа из инета)

Windows 10 Hyper-V разрешает использовать для виртуальной сети собственное преобразование сетевых адресов (NAT).

Пул публичных адресов (**inside global address pool**) доступен для любого устройства во внутренней сети по принципу «первым пришел – первым обслужили». С динамическим NAT один внутренний адрес преобразуется в один внешний адрес. При таком типе перевода должно быть достаточно адресов в пуле для одновременного предоставления для всех внутренних устройств, которым необходим доступ к внешней сети. Если все адреса в пуле были использованы, то устройство должно ждать доступного адреса, прежде чем оно сможет получить доступ к внешней сети.

Рассмотрим настойку по шагам:

1. Определить пул которые будут использоваться для перевода, используя команду **ip nat pool [имя начальный_ip конечный_ip]**. Этот пул адресов обычно представляет собой группу публичных общедоступных адресов. Адреса определяются указанием начального IP-адреса и конечного IP-адреса пула. Ключевые слова **netmask** или **prefix-length** указывают маску.
2. Нужно настроить стандартный **access-list (ACL)**, чтобы определить только те адреса, которые будут транслироваться. Введем команду **access-list [номер_ACL] permit source [wildcard_маска]**. Про стандартные access-list'ы можно прочитать в этой статье (а про расширенные в этой). ACL который разрешает очень много адресов может привести к непредсказуемым результатам, поэтому в конце листа есть команда **deny all**.
3. Необходимо привязать ACL к пулу, и для этого используется команду **ip nat inside source list [номер_ACL] number pool [название_пула]**. Эта конфигурация используется маршрутизатором для определения того, какие устройства (список) получают адреса (пул).
4. Определить, какие интерфейсы находятся внутри, по отношению к NAT, то есть любой интерфейс, который подключен к внутренней сети.
5. Определить, какие интерфейсы находятся снаружи, по отношению к NAT, то есть любой интерфейс, который подключен к внешней сети.

74 Пакет IP Tables

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

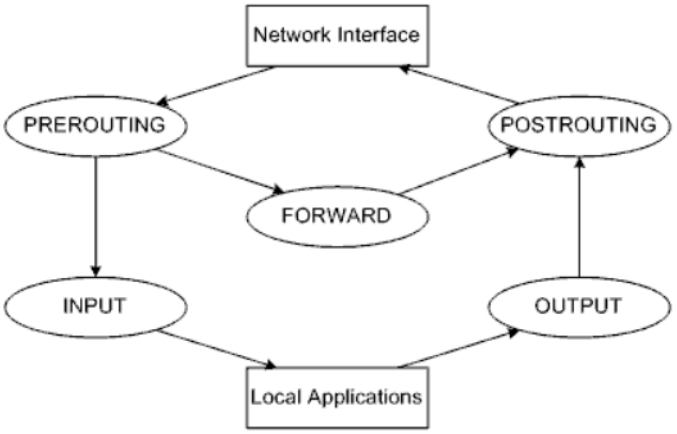


Рисунок -- Стандартные цепочки IP Tables

Общий формат задания правила:

```
iptables [-t table] command [match] [target/jump]
```

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 именем 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 критериев (matches):
`-d` (`--destination`) -- нужен для указания адреса назначения;
`-f` (`--fragment`) -- нужен для включения поддержки фрагментации;
`-i` (`--in-interface`) -- нужен для указания сетевого интерфейса, принимающего пакеты;
`-o` (`--out-interface`) -- нужен для указания сетевого интерфейса, передающего пакеты;
`-p` (`--protocol`) -- нужен для указания протокола;
`-s` (`--source`) -- нужен для указания адреса отправителя.

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.

75 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 (сейчас называется Forefront Threat Management Gateway, но в 2012 г. разработка прекращена), 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 (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.

76 Поддержка NAT в IOS

(Инфа с инета, насчет правильности не особо уверен. Если кто что-нибудь полезное найдет, буду благодарен)

Начиная с версии 11.2 IOS включает в себя поддержку NAT, однако от версии к версии набор функциональных возможностей IOS NAT несколько различается. До версии 12.0 стандартная поставка IOS включает в себя либо только PAT (маскарад), либо не включает NAT вовсе. Поставка «Plus» имеет полную поддержку NAT во всех версиях начиная с 11.2.

Поддерживаемые типы трафика.

Следующие протоколы/приложения поддерживаются Cisco IOS NAT:

* Любой протокол на базе TCP, который не содержит IP-адрес источника или пункта назначения в блоке данных сегмента. Сюда входят протоколы: ICMP, HTTP, BSD rcmd (rcp, rsh, rlogin), SMTP и другие.

* Любой протокол на базе UDP, который не содержит IP-адрес источника или пункта назначения в блоке данных сегмента. Сюда входят протоколы: TFTP, NTP и другие.

* Telnet (RFC 854)

* ICMP (RFC 1256) Протокол управляющих сообщений Internet. Транслируемые Cisco IOS NAT IP-заголовки управляющих сообщений протокола ICMP вновь транслируются в первоначальную форму перед отправкой к пунктам назначения. Cisco IOS NAT обеспечивает перетрансляцию вложенных заголовков пакетов в реальный IP-адрес получателя, чтобы тот мог его распознать.
  Например:
  destination unreachable (пункт назначения недоступен) (3)
  source quench (обрыв источника) (4)
  redirect (перенаправление) (5)
  time exceeded (время просрочено) (11)
  parameter problem (проблема с параметрами) (12)

* FTP (RFC 959) (Протокол передачи файлов)
  Команда FTP PORT и отклик на команду PASV содержат IP-адрес запрашивающей стороны в формате ASCII. Cisco IOS NAT осуществляет контроль управляющего потока FTP и заменяет эти привязки соответствующим ASCII представлением транслируемого IP-адреса.

* SMTP (RFC 821) (Простой протокол передачи электронной почты)
  Действительный e-mail адрес SMTP может содержать IP-адрес, а не полностью квалифицированное имя домена, например, mike@[192.168.1.95]. Cisco IOS NAT не будет транслировать ASCII представление IP-адреса для такого SMTP-адреса. Для трансляции

SMTP-адреса используйте полностью квалифицированное имя домена.

* HTTP (Протокол передачи гипертекстов)
  Строка URL (Унифицированный локатор ресурса), подобная приведенной ниже, может появиться в HTML (Язык разметки гипертекста):
  <a HREF="http://192.168.1.95/foo/bar.html">Sample HTTP link
  Данная строка URL содержит только IP-адрес. Так же, как и в случае SMTP, Cisco IOS NAT не будет транслировать ASCII представление IP-адреса для такой строки URL. Для трансляции адреса в строке URL используйте полностью квалифицированное имя домена. Использование IP-адресов в строках URL в основном не поощряется.

* DNS (Система доменных имен)
  Cisco IOS NAT транслирует IP-адреса в заголовках и полезных нагрузках (payloads) DNS-"адреса" (A), а также инверсный "указатель" (PTR) записей ресурса (RRs). Значения времени существования (Time-to-live - TTL) во всех записях ресурса (RRs), которые получают трансляцию адресов в полезных нагрузках RR, автоматически устанавливаются в ноль. Cisco IOS NAT не транслирует IP-адреса, вложенные в передачу зон DNS.

* NFS (Сетевая файловая система)

* NetBIOS over TCP/IP

* CuSeeMe

* RealAudio

* StreamWorks

//

Неподдерживаемые типы трафика.

Следующие протоколы/приложения НЕ поддерживаются Cisco IOS NAT:

* IP Multicast (IP-мультиотправка)

* Routing Updates (Обновления маршрутизации)
  + Cisco IOS NAT не транслирует адреса сети и хоста в обновлениях таблицы маршрутизации. Для объявления возможности доступа к внутренней виртуальной сети ("внутренние глобальные" адреса) нужно создать статический маршрут null0 к этим адресам, а затем перераспределить их.

* DNS Zone Transfers (Передача зон DNS)

* BOOTP (Протокол самозагрузки)

+ Хотя Cisco IOS NAT будет транслировать IP-адреса в заголовках пакетов "ip-helpered", она не будет транслировать IP-адреса в сообщениях BOOTP.

* talk and ntalk

* H.323

* VDOLive

* NetShow

* VXtreme