Вопросы к экзамену по дисциплине «Компьютерные сети» 2020

# Разновидности физических сетевых топологий. Сравнительный анализ топологий "шина", "звезда", "кольцо".

Сетевая топология – это конфигурация графа, вершинам которого соответствуют конечные узлы сети (компьютеры) и коммуникационное оборудование (маршрутизаторы), а рёбрам – физические или информационные связи между вершинами.

Сетевая топология может быть

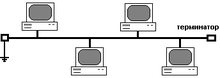
физической – описывает реальное расположение и связи между узлами сети.

логической – описывает хождение сигнала в рамках физической топологии.

информационной – описывает направление потоков информации, передаваемых по сети.

управления обменом – это принцип передачи права на пользование сетью.

**Шина**



Топология данного типа представляет собой общий кабель (называемый шина или магистраль), к которому подсоединены все рабочие станции. На концах кабеля находятся терминаторы, для предотвращения отражения сигнала.

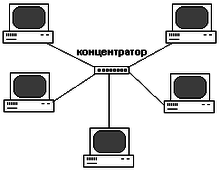
Преимущества сетей шинной топологии:

* расход кабеля существенно уменьшен;
* отказ одного из узлов не влияет на работу сети в целом;
* сеть легко настраивать и конфигурировать;
* сеть устойчива к неисправностям отдельных узлов.

Недостатки сетей шинной топологии:

* разрыв кабеля может повлиять на работу всей сети;
* ограниченная длина кабеля и количество рабочих станций;
* недостаточная надежность сети из-за проблем с разъемами кабеля;
* низкая производительность, обусловлена разделением канала между всеми абонентами.

**Звезда**



В сети, построенной по топологии типа «звезда», каждая рабочая станция подсоединяется кабелем (витой парой) к концентратору, или хабу. Концентратор обеспечивает параллельное соединение ПК и, таким образом, все компьютеры, подключенные к сети, могут общаться друг с другом.

Данные от передающей станции сети передаются через хаб по всем линиям связи всем ПК. Информация поступает на все рабочие станции, но принимается только теми станциями, которым она предназначается. Так как передача сигналов в топологии физическая звезда является широковещательной, то есть сигналы от ПК распространяются одновременно во все направления, то логическая топология данной локальной сети является логической шиной.

Данная топология применяется в локальных сетях с архитектурой 10Base-T Ethernet.

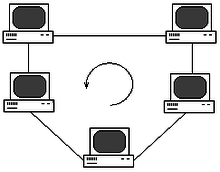
Преимущества сетей топологии звезда:

* легко подключить новый ПК;
* имеется возможность централизованного управления;
* сеть устойчива к неисправностям отдельных ПК и к разрывам соединения отдельных ПК.

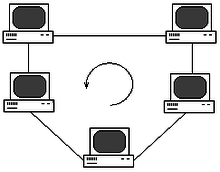
Недостатки сетей топологии звезда:

* отказ хаба влияет на работу всей сети;
* большой расход кабеля.

**Кольцо**



В сети с топологией типа «кольцо» все узлы соединены каналами связи в неразрывное кольцо, по которому передаются данные. Выход одного ПК соединяется со входом другого ПК. Начав движение из одной точки, данные, в конечном счете, попадают на его начало. Данные в кольце всегда движутся в одном и том же направлении.

[](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Ring_topology.PNG)

Топология кольцо

Принимающая рабочая станция распознает и получает только адресованное ей сообщение. В сети с топологией типа физическое кольцо используется маркерный доступ, который предоставляет станции право на использование кольца в определенном порядке. Логическая топология данной сети – логическое кольцо. Данную сеть очень легко создавать и настраивать.

К основному недостатку сетей топологии кольцо относится то, что повреждение линии связи в одном месте или отказ ПК приводит к неработоспособности всей сети.

Как правило, в чистом виде топология «кольцо» не применяется из-за своей ненадёжности, поэтому на практике применяются различные модификации кольцевой топологии.

# Концепция и основные понятия эталонной модели взаимодействия открытых систем (ISO/OSI). Функции отдельных уровней OSI.

OSI расшифровывается как Open System Interconnection - эталонная модель, которая описывает, как информация из программного приложения на одном компьютере перемещается через физический носитель к программному приложению на другом компьютере.

OSI состоит из семи уровней, и каждый уровень выполняет определенную сетевую функцию.

Модель OSI была разработана Международной организацией по стандартизации (ISO) в 1984 году, и теперь она рассматривается как архитектурная модель для межкомпьютерных коммуникаций.

Модель OSI делит всю задачу на семь небольших и управляемых задач. Каждому слою назначается определенная задача.

Каждый уровень является автономным, поэтому задача, назначенная каждому уровню, может выполняться независимо.

**Функции уровней OSI**

Есть семь уровней OSI. Каждый слой имеет разные функции. Список из семи слоев приведен ниже:

1. Физический слой, Уровень приложений, так же называют прикладной

2. Уровень представления

3. Сессионный слой

4. Транспортный уровень

5. Сетевой уровень

6. Канал данных, так же называют канальным

7. Физический слой.

**Физический слой**

Основная функциональность физического уровня заключается в передаче отдельных битов от одного узла к другому узлу. Это самый низкий уровень модели OSI. Он устанавливает, поддерживает и деактивирует физическую связь. Он определяет механические, электрические и процедурные характеристики сетевого интерфейса.

**Функции физического уровня:**· Конфигурация линии: определяет способ физического соединения двух или более устройств.· Передача данных : определяет режим передачи между двумя устройствами в сети: симплексный, полудуплексный или полудуплексный. Топология : определяет способ организации сетевых устройств.· Сигналы: определяет тип сигнала, используемого для передачи информации.

**Канальный**

1. Этот слой отвечает за безошибочную передачу кадров данных. Он определяет формат данных в сети, обеспечивает надежную и эффективную связь между двумя или более устройствами, отвечает за уникальную идентификацию каждого устройства, которое находится в локальной сети.

Уровень содержит содержит два подслоя:  
**1)** **Уровень управления логической связью:**  
Отвечает за передачу пакетов на сетевой уровень принимающего получателя.  
Так же идентифицирует адрес протокола сетевого уровня из заголовка.  
Это также обеспечивает управление потоком.  
**2)** **Уровень контроля доступа к медиа:**  
Уровень управления доступом к среде является связующим звеном между уровнем управления логическим каналом и физическим уровнем сети.  
Он используется для передачи пакетов по сети.

**Функции канального уровня**

* *Кадрирование*: канальный уровень преобразует необработанный битовый поток физического объекта в пакеты, известные как кадры. Уровень передачи данных добавляет заголовок и трейлер к фрейму. Заголовок, который добавляется к фрейму, содержит аппаратный адрес назначения и адрес источника.
* *Физическая адресация*: канальный уровень добавляет заголовок к фрейму, который содержит адрес назначения. Кадр передается по адресу назначения, указанному в заголовке.
* *Управление потоком*: управление потоком является основной функциональностью уровня канала передачи данных. Это метод, с помощью которого поддерживается постоянная скорость передачи данных с обеих сторон, чтобы не повредить данные. Это гарантирует, что передающая станция, такая как сервер с более высокой скоростью обработки, не превышает принимающую станцию, с более низкой скоростью обработки.
* *Контроль ошибок*: Контроль ошибок достигается путем добавления вычисленного значения CRC (англ. Cyclic redundancy check), которое помещается в трейлер уровня звена данных, который добавляется в кадр сообщения перед его отправкой на физический уровень. Принцип работы CRC такова: два устройства работают по стандарту - одно передающее другое принимающее, оба формируют поле для внесения контрольной суммы - отправитель пишут сумму получатель проверяет сумму, если сумма не совпадает - кадр отбрасывается, после чего получатель отправляет подтверждение для повторной передачи поврежденных кадров.
* *Контроль доступа*: когда два или более устройств подключены к одному и тому же каналу связи, протоколы уровня канала передачи данных используются для определения того, какое устройство имеет контроль над каналом в данный момент времени.

**Сетевой уровень.**

Это уровень 3, который управляет адресацией устройств, отслеживает расположение устройств в сети.  
Он определяет наилучший путь для перемещения данных из источника в место назначения в зависимости от состояния сети, приоритета обслуживания и других факторов.  
Канальный уровень передачи данных отвечает за маршрутизацию и пересылку пакетов.  
Маршрутизаторы - это устройства уровня 3, они указаны на этом уровне и используются для предоставления услуг маршрутизации в пределах межсетевого взаимодействия.  
Протоколы, используемые для маршрутизации сетевого трафика, называются протоколами сетевого уровня. Примерами протоколов являются IPV4 и Ipv6.

**Функции сетевого уровня:**

* *Межсетевое взаимодействие*: межсетевое взаимодействие является основной обязанностью сетевого уровня. Это обеспечивает логическую связь между различными устройствами.
* *Адресация*: Сетевой уровень добавляет адрес источника и назначения в заголовок кадра. Адресация используется для идентификации устройства в интернете.
* *Маршрутизация*. Маршрутизация является основным компонентом сетевого уровня и определяет оптимальный оптимальный путь из нескольких путей от источника к месту назначения.
* *Пакетирование*: сетевой уровень получает пакеты от верхнего уровня и преобразует их в пакеты. Этот процесс известен как Пакетирование. Это достигается с помощью интернет-протокола (IP).

**Транспортный уровень**

Транспортный уровень - это Уровень 4, гарантирующий, что сообщения передаются в том порядке, в котором они были отправлены, и нет дублирования данных.  
Основная ответственность транспортного уровня заключается в полной передаче данных.  
Он получает данные из верхнего уровня и преобразует их в меньшие единицы, известные как сегменты.  
Этот уровень можно назвать сквозным уровнем, поскольку он обеспечивает двухточечное соединение между источником и пунктом назначения для надежной доставки данных.

**Два протокола, используемые на этом уровне:**

1. Протокол управления передачей  
   Это стандартный протокол, который позволяет системам общаться через Интернет.  
   Он устанавливает и поддерживает связь между хостами.  
   Когда данные отправляются через соединение TCP, тогда протокол TCP делит данные на более мелкие единицы, известные как сегменты. Каждый сегмент проходит через Интернет, используя несколько маршрутов, и они прибывают в пункт назначения в разных порядках. Протокол управления передачей переупорядочивает пакеты в правильном порядке на принимающей стороне.
2. Протокол пользовательских датаграмм  
   Протокол пользовательских дейтаграмм - это протокол транспортного уровня.  
   Это ненадежный транспортный протокол, так как в этом случае получатель не отправляет подтверждение при получении пакета, отправитель не ожидает подтверждения. Следовательно, это делает протокол ненадежным.

**Функции транспортного уровня:**

* *Адресация точки обслуживания*: компьютеры запускают несколько программ одновременно, по этой причине происходит передача данных из источника в место назначения не только с одного компьютера на другой компьютер, но и от одного процесса к другому процессу. Транспортный уровень добавляет заголовок, который содержит адрес, известный как адрес точки обслуживания или адрес порта. Ответственность сетевого уровня заключается в передаче данных с одного компьютера на другой компьютер, а ответственность транспортного уровня - в передаче сообщения правильному процессу.
* *Сегментация и повторная сборка*: когда транспортный уровень получает сообщение от верхнего уровня, он разделяет сообщение на несколько сегментов, и каждому сегменту присваивается порядковый номер, который уникально идентифицирует каждый сегмент. Когда сообщение прибыло в пункт назначения, тогда транспортный уровень повторно собирает сообщение на основе их порядковых номеров.
* *Управление соединением*: Транспортный уровень предоставляет две службы: служба, ориентированная на соединение, и служба без соединения.

1. Служба без установления соединения обрабатывает каждый сегмент как отдельный пакет, и все они перемещаются по разным маршрутам, чтобы достичь пункта назначения.
2. Служба, ориентированная на установление соединения, устанавливает соединение с транспортным уровнем на машине назначения - до доставки пакетов. В сервисе, ориентированном на соединение, все пакеты передаются по одному маршруту.

* *Управление потоком*: транспортный уровень также отвечает за управление потоком.
* *Контроль ошибок*: Транспортный уровень также отвечает за контроль ошибок. Контроль ошибок выполняется сквозным, а не по одной ссылке. Транспортный уровень отправителя гарантирует, что сообщение достигнет пункта назначения без каких-либо ошибок.

**Сессионный слой**

Это уровень 3 в модели OSI.  
Сеансовый уровень используется для установления, поддержания и синхронизации взаимодействия между устройствами связи.  
**Функции сессионного слоя**:

* *Диалоговое управление*: Сеансовый уровень действует как диалоговый контроллер, который создает диалог между двумя процессами, или мы можем сказать, что он обеспечивает связь между двумя процессами, которые могут быть либо полудуплексными, либо полнодуплексными.
* *Синхронизация*: Сеансовый уровень добавляет некоторые контрольные точки при передаче данных в последовательности. Если во время передачи данных произойдет какая-либо ошибка, то передача будет повторяться с контрольной точки. Этот процесс известен как Синхронизация и восстановление.

**Уровень представления**

Уровень представления в основном касается синтаксиса и семантики информации, которой обмениваются две системы.  
Он действует как переводчик данных для сети.  
Этот слой является частью операционной системы, которая преобразует данные из одного формата представления в другой формат.  
Уровень представления также известен как уровень синтаксиса

**Функции презентационного слоя**:

* Перевод: процессы в двух системах обмениваются информацией в виде символьных строк, чисел и так далее. Разные компьютеры используют разные методы кодирования, уровень представления управляет взаимодействием между различными методами кодирования. Он преобразует данные из зависимого от отправителя формата в общий формат и изменяет общий формат в зависимый от получателя формат на принимающей стороне.
* Шифрование. Шифрование необходимо для обеспечения конфиденциальности. Шифрование - это процесс преобразования передаваемой отправителем информации в другую форму и отправки полученного сообщения по сети.
* *Сжатие* - это процесс сжатия данных, т.е. Сокращение числа передаваемых битов. Сжатие данных очень важно в мультимедиа, таких как текст, аудио, видео.

**Уровень приложений**

Прикладной уровень служит окном для пользователей и процессов приложений для доступа к сетевому сервису.  
Он решает такие вопросы, как прозрачность сети, распределение ресурсов и т. Д.  
Прикладной уровень не является приложением, но он выполняет функции прикладного уровня.  
Этот уровень предоставляет сетевые услуги конечным пользователям.

**Функции прикладного уровня**:

* Передача, доступ и управление файлами (FTAM): прикладной уровень позволяет пользователю получать доступ к файлам на удаленном компьютере, извлекать файлы с компьютера и управлять файлами на удаленном компьютере.
* Почтовые службы: прикладной уровень предоставляет средства для пересылки и хранения электронной почты.
* Службы каталогов: приложение предоставляет источники распределенной базы данных и используется для предоставления этой глобальной информации о различных объектах.

# Физический уровень модели OSI. Типы медных кабелей и оптических кабелей

**Физический слой**

Основная функциональность физического уровня заключается в передаче отдельных битов от одного узла к другому узлу.Это самый низкий уровень модели OSI. Он устанавливает, поддерживает и деактивирует физическую связь.Он определяет механические, электрические и процедурные характеристики сетевого интерфейса.

**Функции физического уровня:**· Конфигурация линии: определяет способ физического соединения двух или более устройств.· Передача данных : определяет режим передачи между двумя устройствами в сети: симплексный, полудуплексный или полудуплексный.·Топология : определяет способ организации сетевых устройств.· Сигналы: определяет тип сигнала, используемого для передачи информации.

**Силовые кабели**

Среди наиболее популярных в последнее время видов кабельной продукции можно назвать кабель ВВГ и его модификации. ВВГ обозначается силовой кабель с изоляцией ТПЖ из ПВХ, оболочкой (кембриком) из ПВХ, медным материалом жилы, не имеющий внешней защиты.

Используется для передачи и распределения электрического тока, рабочее напряжение 660 – 1000 В, частота 50 Гц. Количество жил может варьироваться от 1 до 5. Сечение – от 1,5 кв.мм до 240 кв.мм. Жилы могут быть как одно-, так и многопроволочными.

**Коаксиальный кабель**

Коаксиальный кабель представляет собой электрический кабель, состоящий из центрального провода и металлической оплетки, разделенных между собой слоем диэлектрика (внутренней изоляции) и помещенных в общую внешнюю оболочку.

К нему труднее механически подключиться для несанкционированного прослушивания сети, он также дает заметно меньше электромагнитных излучений вовне. Однако монтаж и ремонт коаксиального кабеля существенно сложнее, чем витой пары, а стоимость его выше (он дороже примерно в 1,5-3 раза по сравнению с кабелем на основе витых пар). Сложнее и установка разъемов на концах кабеля. Поэтому его сейчас применяют реже, чем витую пару.

**Витая пара**

Служит для построения компьютерных сетей. Витая пара может быть экранированной и неэкранированной.

Состоит из одной или нескольких пар проводов, перевитых попарно, что делается в целях улучшения приема и передачи сигнала. Проводники в парах изготовлены из монолитной медной проволоки толщиной 0,4—0,6 мм. Скручивание проводов снижает влияние внешних и взаимных помех на полез­ные сигналы, передаваемые по кабелю (электромагнитные помехи одинаково влияют на оба провода пары).

Также внутри кабеля встречается так называемая «разрывная нить» (обычно капрон), которая используется для облегчения разделки внешней оболочки — при вытягивании она делает на оболочке продольный разрез, который открывает доступ к кабельному сердечнику, гарантированно не повреждая изоляцию проводников. Также разрывная нить, ввиду своей высокой прочности на разрыв, выполняет защитную функцию.

Каждый проводник заключен в изоляцию из ПВХ или пропилена. Внешняя оболочка также из ПВХ. Кабель может быть дополнительно оснащен влагонепроницаемой оболочкой из полипропилена.

**Оптоволокно**

Оптоволоконный кабель (он же волоконно-оптический) — это принципиально иной тип кабеля по сравнению с другими типами электрических или медных кабелей. Информация по нему передается не электрическим сигналом, а световым. Главный его элемент — это прозрачное стекловолокно, по которому свет проходит на огромные расстояния (до десятков километров) с незначительным ослаблением.

Волоконно-оптический кабель состоит из тонких (5-60 микрон) гибких стек­лянных волокон (волоконных световодов), по которым распространяются свето­вые сигналы. Это наиболее качественный тип кабеля — он обеспечивает переда­чу данных с очень высокой скоростью (до 10 Гбит/с и выше) и к тому же лучше других типов передающей среды обеспечивает защиту данных от внешних помех (в силу особенностей распространения света такие сигналы легко экранировать).

Каждый световод состоит из центрального проводника света (сердцевины) — стеклянного волокна, и стеклянной оболочки, обладающей меньшим показате­лем преломления, чем сердцевина. Распространяясь по сердцевине, лучи света не выхолят за ее пределы, отражаясь от покрывающего слоя оболочки.

# Понятие домена коллизий. Случайные методы доступа CSMA/CD и CSMA/CА.

**Доме́н колли́зий** – часть сети Ethernet, все узлы которой конкурируют за общую разделяемую среду передачи и, следовательно, каждый узел которой может создать коллизию с любым другим узлом этой части сети.

Другими словами – сегмент сети, имеющий общий канальный уровень модели OSI, в котором передать фрейм может только один абонент одновременно. Задержка распространения фреймов между станциями, либо одновременное начало передачи вызывает возникновение коллизий, которые требуют специальной обработки и снижают производительность сети.

Чем больше узлов в таком сегменте – тем выше вероятность коллизий. Для разделения домена коллизий применяются коммутаторы.

Понятие существует независимо от применяемого стандарта физического уровня.

**CSMA/CA**

Метод доступа к передающей среде CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), или метод множественного доступа с контролем несущей частоты и избеганием коллизий, также является модификацией протокола CSMA. Данный метод доступа к среде чаще всего используется в беспроводных сетях, работа которых описана спецификацией IEEE 802.11.

В отличие от метода доступа к среде **CSMA/CD**, в которой jam-сигнал высылается только при обнаружении коллизии, метод CSMA/CA сначала отправляет jam-сигнал, информирующий о том, что станция хочет передавать данные, и только потом передает сигнал. После того как выслан jam-сигнал, станция еще некоторое время ожидает и проверяет канал на наличие аналогичных jam-пакетов. Если таковой обнаружен, то есть кто-то уже ведет вещание, станция ждет случайный промежуток времени, и затем процесс повторяется. Если никаких чужих передач не обнаружено, станция начинает передавать данные до тех пор, пока все они не будут переданы. При таком подходе, даже если будет обнаружен чужой jam-пакет, это приведет не к коллизии при передаче данных, а лишь к коллизии jam-пакетов.

Метод множественного доступа с прослушиванием несущей и разрешением коллизий (**CSMA/CD**) устанавливает следующий порядок: если рабочая станция хочет воспользоваться сетью для передачи данных, она сначала должна проверить состояние канала: начинать передачу станция может, если канал свободен. В процессе передачи станция продолжает прослушивание сети для обнаружения возможных конфликтов. Если возникает конфликт из-за того, что два узла попытаются занять канал, то обнаружившая конфликт интерфейсная плата, выдает в сеть специальный сигнал, и обе станции одновременно прекращают передачу. Принимающая станция отбрасывает частично принятое сообщение, а все рабочие станции, желающие передать сообщение, в течение некоторого, случайно выбранного промежутка времени выжидают, прежде чем начать сообщение.

# Канальный уровень. Назначение LLС и МАС подуровня.

**Канальный**

1. Этот слой отвечает за безошибочную передачу кадров данных. Он определяет формат данных в сети, обеспечивает надежную и эффективную связь между двумя или более устройствами, отвечает за уникальную идентификацию каждого устройства, которое находится в локальной сети.

Уровень содержит содержит два подслоя:  
**1)** **Уровень управления логической связью:**  
Отвечает за передачу пакетов на сетевой уровень принимающего получателя.  
Так же идентифицирует адрес протокола сетевого уровня из заголовка.  
Это также обеспечивает управление потоком.  
**2)** **Уровень контроля доступа к медиа:**  
Уровень управления доступом к среде является связующим звеном между уровнем управления логическим каналом и физическим уровнем сети.  
Он используется для передачи пакетов по сети.

**Функции канального уровня**

* *Кадрирование*: канальный уровень преобразует необработанный битовый поток физического объекта в пакеты, известные как кадры. Уровень передачи данных добавляет заголовок и трейлер к фрейму. Заголовок, который добавляется к фрейму, содержит аппаратный адрес назначения и адрес источника.
* *Физическая адресация*: канальный уровень добавляет заголовок к фрейму, который содержит адрес назначения. Кадр передается по адресу назначения, указанному в заголовке.
* *Управление потоком*: управление потоком является основной функциональностью уровня канала передачи данных. Это метод, с помощью которого поддерживается постоянная скорость передачи данных с обеих сторон, чтобы не повредить данные. Это гарантирует, что передающая станция, такая как сервер с более высокой скоростью обработки, не превышает принимающую станцию, с более низкой скоростью обработки.
* *Контроль ошибок*: Контроль ошибок достигается путем добавления вычисленного значения CRC (англ. Cyclic redundancy check), которое помещается в трейлер уровня звена данных, который добавляется в кадр сообщения перед его отправкой на физический уровень. Принцип работы CRC такова: два устройства работают по стандарту - одно передающее другое принимающее, оба формируют поле для внесения контрольной суммы - отправитель пишут сумму получатель проверяет сумму, если сумма не совпадает - кадр отбрасывается, после чего получатель отправляет подтверждение для повторной передачи поврежденных кадров.
* *Контроль доступа*: когда два или более устройств подключены к одному и тому же каналу связи, протоколы уровня канала передачи данных используются для определения того, какое устройство имеет контроль над каналом в данный момент времени.

**Подуровни LLC и MAC канального уровня, формат кадра Ethernet**

Канальный уровень разделен на 2 подуровня: LLC (Logical Link Control) - подуровень логической передачи данных и MAC (Media Access Control) - подуровень управления доступом к среде.

Международным институтом инженеров по электротехнике и радиоэлектронике IEEE (Institute of Electrical and Electronics Engineers) разработано семейство стандартов 802.х, описывающих работу канального и физического уровня модели OSI.

**Подуровень LLC канального уровня**

Реализует связь с сетевым уровнем, на этом подуровне существуют алгоритмы (логические процедуры), позволяющие устанавливать или не устанавливать связь перед передачей кадров, восстанавливать или не восстанавливать кадры при их потере или обнаружении ошибок. Протокол 802.2 реализует связь с сетевым уровнем и определение логических процедур передачи кадров по сети.

Все кадры LLC подразделяются на три типа — информационные, управляющие, ненумерованные и имеют единый формат:

FLAG DSAP SSAP CONTROL DATA FLAG

Flag - границы кадров LLC.

Data - поле данных данные сетевых протоколов.

DSAP (Destination Service Access Point) - поле адреса точки входа службы назначения (получателя).

SSAP (Source Service Access Point) - поле адреса точки входа службы источника.

Эти поля указывают службу верхнего уровня, которая передает и принимает пакеты данных. Например, служба IP имеет значение SAP равное 0х6. Обычно это одинаковые адреса. Адреса DSAP и SSAP могут различаться только в том случае, если служба имеет несколько адресов точек входа.

Control - поле управления имеет длину 1 или 2 байта в зависимости от того, какой тип кадра передается: информационный, управляющий, ненумерованный.

Тип кадра определяется процедурой управления логическим каналом LLC. Стандартом 802.2 предусмотрено 3 типа таких процедур:

LLC1 - процедура без установления соединения и подтверждения, используется при дейтаграммном режиме передачи данных. Для передачи данных используются ненумерованные кадры. Восстановление принятых с ошибками данных производят протоколы верхних уровней. В данном режиме функционирует, например, протокол IP.

LLC2 - процедура с установлением соединения и подтверждением, перед началом передачи данных устанавливает соединение, послав соответствующий запрос и получив подтверждение, после чего передаются данные. Процедура позволяет восстанавливать потерянные и исправлять ошибочные данные, используя режим скользящего окна. Для этих целей она использует все три типа кадров (информационные, управляющие, ненумерованные). Данная процедура более сложная и менее быстродействующая по сравнению с LLC1, поэтому она используется в локальных сетях значительно реже, чем LLC1, например, протоколом NetBIOS/NetBEUI. Широкое применение процедура, подобная LLC2, получила в глобальных сетях для надежной передачи данных по ненадежным линиям связи. Например, она используется в протоколе LAP-B сетей Х.25, в протоколе LAP-D сетей ISDN, в протоколе LAP-M сетей с модемами, частично – в протоколе LAP-F сетей Frame Relay.

LLC3 - процедура без установления соединения, но с подтверждением, используется в системах управления технологическими процессами, когда необходимо высокое быстродействие и знание того, дошла ли управляющая информация до объекта.

Подуровень МАС канального уровня

Определяет особенности доступа к физической среде при использовании различных технологий локальных сетей. Протоколы МАС-уровня ориентированы на совместное использование физической среды абонентами. Разделяемая среда (shared media) используется в таких широко распространенных в локальных сетях технологиях как Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring, FDDI. Использование разделяемой между пользователями среды улучшает загрузку канала связи, удешевляет сеть, но ограничивает скорость передачи данных между двумя узлами.

Каждой технологии МАС-уровня соответствует несколько вариантов (спецификаций) протоколов физического уровня.

Ethernt (802.3) - соответствуют спецификации: 10Base-T, 10Base-FB, 10Base-FL, 10Base-5, 10Base-2 ит.д. Скорость до 10 Мбит/c.

Fast Ethernet (802.3u) - соответствуют сецификации: 100Base-T4, 100Base-TX, 100Base-FX и т.д. Скорость до 100 Мбит/c.

Gigabit Ethernet (802.3z и 802.3ab) - соответствуют спецификации: 1000BASE-T, 1000BASE-X и т.д. Скорость до 1000 Мбит/c.

10Gigabit Ethernet (802.3ае) - соответствуют спецификации: 10GBASE-CX4, 10GBASE-LR и т.д. Скорость до 10000 Мбит/c.

Token Ring (802.5) - в качестве физической среды используется экранированная витая пара STP, с помощью которой все станции сети соединяются в кольцевую структуру. В отличие от технологии Ethernet в сетях с передачей маркера (Token Ring) реализуется не случайный, а детерминированный доступ к среде с помощью кадра специального формата – маркера (token). Сети Token Ring используют два различных алгоритма функционирования, позволяющих передавать данные по кольцу со скоростями либо 4 Мбит/c, либо 16 Мбит/c, несовместима с Ethernet и в настоящее время почти полностью вытеснена технологией Ethernet.

Связь Подуровень LLC и подуровня MAC, формат кадра Ethernet

Кадр LLC уровня передается на МАС-уровень, где инкапсулируется в кадр соответствующей технологии данного уровня. При этом флаги кадра LLC отбрасываются. Технология Ethernet предусматривает четыре формата кадров: 802.3/LLC, Raw 802.3/Novell 802.3, Ethernet DIX/Ethernet II, Ethernet SNAP

Рассмотрим основной тип Ethernet DIX/Ethernet II

Perambule Destination Address Sourse Address Type/Length Data FCS

Perambule (Перамбула) - используется для синхронизации, состоит из 8 байт.

DA (Destination Address) - mac адрес узла назначения, состоит из 8 байт.

SA (Source Address) - mac адрес узла источника, состоит из 8 байт.

Type/Length - длину или тип, числовое значение этого поля определяет его смысл: если значение меньше 1500, то это поле длины, а если больше это тип.

Data — данные, могут быть от 46 до 1500 байт.

FCS (Frame Check Sequence) - контрольная сумма, состоит из 4-х байтов, служит для обнаружения ошибок в полученном кадре, использует алгоритм проверки на основе циклического кода.

# Общие сведения и формат кадров технологии Ethernet. Формат и типы MAC – адресов.

В сетях Ethernet на канальном уровне используются кадры 4-х различных форматов. Это связано с длительной историей развития технологии Ethernet, насчитывающей период существования до принятия стандартов IEEE 802, когда подуровень LLC не выделялся из общего протокола и, соответственно, заголовок LLC не применялся.

Различия в форматах кадров могут приводить к несовместимости в работе аппаратуры и сетевого программного обеспечения, рассчитанного на работу только с одним стандартом кадра Ethernet. Однако сегодня практически все сетевые адаптеры, драйверы сетевых адаптеров, мосты/коммутаторы и маршрутизаторы умеют работать со всеми используемыми на практике форматами кадров технологии Ethernet, причем распознавание типа кадра выполняется автоматически.

**Формат кадра**

Первоначальный Version I (больше не применяется).

Ethernet Version 2 или Ethernet-кадр II, ещё называемый DIX (аббревиатура первых букв фирм-разработчиков DEC, Intel, Xerox) — наиболее распространена и используется по сей день. Часто используется непосредственно протоколом Интернет.

Novell — внутренняя модификация IEEE 802.3 без LLC (Logical Link Control).

Кадр IEEE 802.3 LLC.

Кадр IEEE 802.3 LLC/SNAP.

Некоторые сетевые карты Ethernet, производимые компанией Hewlett-Packard, использовали при работе кадр формата IEEE 802.12, соответствующий стандарту 100VG-AnyLAN.

В качестве дополнения Ethernet-кадр может содержать тег IEEE 802.1Q для идентификации VLAN, к которой он адресован, а в нём IEEE 802.1p для указания приоритетности.

Разные типы кадра имеют различный формат и значение MTU.

**MAC-адреса**

При проектировании стандарта Ethernet было предусмотрено, что каждая сетевая карта (равно как и встроенный сетевой интерфейс) должна иметь уникальный шестибайтный номер (MAC-адрес), прошитый в ней при изготовлении. Этот номер используется для идентификации отправителя и получателя кадра, и предполагается, что при появлении в сети нового компьютера (или другого устройства, способного работать в сети) сетевому администратору не придётся настраивать MAC-адрес.

Уникальность MAC-адресов достигается тем, что каждый производитель получает в координирующем комитете IEEE Registration Authority диапазон из шестнадцати миллионов (224) адресов, и по мере исчерпания выделенных адресов может запросить новый диапазон. Поэтому по трём старшим байтам MAC-адреса можно определить производителя. Существуют таблицы, позволяющие определить производителя по MAC-адресу; в частности, они включены в программы типа arpalert.

MAC-адрес считывается один раз из ПЗУ при инициализации сетевой карты, в дальнейшем все кадры генерируются операционной системой. Все современные операционные системы позволяют поменять его. Для Windows, начиная, как минимум, с Windows 98, он менялся в реестре. Некоторые драйверы сетевых карт давали возможность изменить его в настройках, но смена работает абсолютно для любых карт.

Некоторое время назад, когда драйверы сетевых карт не давали возможность изменить свой MAC-адрес, а альтернативные возможности не были слишком известны, некоторые провайдеры Internet использовали его для идентификации машины в сети при учёте трафика. Программы из Microsoft Office, начиная с версии Office 97, записывали MAC-адрес сетевой платы в редактируемый документ в качестве составляющей уникального GUID-идентификатора.

**Типы MAC-адресов**

Есть три вида адресов:

Индивидуальный (unicast);

Когда вы передаете данные на индивидуальный мак адрес, то эти данные получает только один компьютер подключенный к сети. 30-9C-23-15-E8-8C

Групповой (multicast, первый бит старшего байта адреса равен 1);

Если передаем данные на групповой мак-адрес, то эти данные получают компьютеры, которые входят в группу. На этих компьютерах должен быть настроен прием данных по этому групповому MAC-адресу. Для того чтобы указать, что МАК адрес является групповым, используется первый бит старшего байта, и этот бит должен быть равен единице. 01-80-С2-00-00-08 Первый бит означает младший, поэтому 01.

Широковещательный (broadcast, все 1);

Это адрес состоящий из всех битовых единиц. В шестнадцатеричном виде он записывается вот так FF-FF-FF-FF-FF-FF. Когда данные отправляются на такой адрес их принимают все компьютеры в сети.

# Функционирование мостов. Отображение логических адресов (IP) на физические (МАС) –протоколы ARP.

Мост – это устройство, обеспечивающее взаимосвязь нескольких локальных сетей посредством передачи кадров из одной сети в другую. В отличие от концентраторов, которые проверяют электрические сигналы, мост проверяет только кадры. Мосты не повторяют шумы, ошибки или испорченные кадры. Мост выступает по отношению к каждой из соединяемых им сетей в качестве конечного узла. Он принимает кадр, сохраняет его в буферной памяти, анализирует адрес назначения кадра. В случае принадлежности кадра к сети, из которой он получен, мост на этот кадр не реагирует. Если необходимо передать кадр в другую сеть, мост должен получить доступ к ее разделяемой среде передачи данных в соответствии с теми же правилами, что и обычный узел.

Локальные мосты оборудуются портами для подключения к ЛВС. Типичными для такой среды носителями являются коаксиальный или волоконно-оптический кабель, а также витая пара проводов. Важным свойством локальных мостов является их способность соединять сети, использующие разные среды. Например, с их помощью можно подключить сеть на коаксиальном кабеле к сети с волоконно-оптическим кабелем или любую из них к сети на витой паре.

Глобальные мосты – это те мосты, порты которых согласуются со средами для передачи информации на большие расстояния. У глобальных мостов могут быть интерфейсы как для передачи на большие расстояния, так и локальные

В протоколе IP-адрес узла, то есть адрес компьютера или порта маршрутизатора, назначается произвольно администратором сети и прямо не связан с его локальным адресом, как это сделано, например, в протоколе IPX. Подход, используемый в IP, удобно использовать в крупных сетях и по причине его независимости от формата локального адреса, и по причине стабильности, так как в противном случае, при смене на компьютере сетевого адаптера это изменение должны бы были учитывать все адресаты всемирной сети Internet (в том случае, конечно, если сеть подключена к Internet'у).

Локальный адрес используется в протоколе IP только в пределах локальной сети при обмене данными между маршрутизатором и узлом этой сети. Маршрутизатор, получив пакет для узла одной из сетей, непосредственно подключенных к его портам, должен для передачи пакета сформировать кадр в соответствии с требованиями принятой в этой сети технологии и указать в нем локальный адрес узла, например его МАС-адрес. В пришедшем пакете этот адрес не указан, поэтому перед маршрутизатором встает задача поиска его по известному IP-адресу, который указан в пакете в качестве адреса назначения. С аналогичной задачей сталкивается и конечный узел, когда он хочет отправить пакет в удаленную сеть через маршрутизатор, подключенный к той же локальной сети, что и данный узел.

Для определения локального адреса по IP-адресу используется протокол разрешения адреса *Address Resolution Protocol, ARP*. Протокол ARP работает различным образом в зависимости от того, какой протокол канального уровня работает в данной сети - протокол локальной сети (Ethernet, Token Ring, FDDI) с возможностью широковещательного доступа одновременно ко всем узлам сети, или же протокол глобальной сети (X.25, frame relay), как правило не поддерживающий широковещательный доступ. Существует также протокол, решающий обратную задачу - нахождение IP-адреса по известному локальному адресу. Он называется реверсивный ARP - *RARP (Reverse Address Resolution Protocol)* и используется при старте бездисковых станций, не знающих в начальный момент своего IP-адреса, но знающих адрес своего сетевого адаптера.

В локальных сетях протокол ARP использует широковещательные кадры протокола канального уровня для поиска в сети узла с заданным IP-адресом.

Узел, которому нужно выполнить отображение IP-адреса на локальный адрес, формирует ARP запрос, вкладывает его в кадр протокола канального уровня, указывая в нем известный IP-адрес, и рассылает запрос широковещательно. Все узлы локальной сети получают ARP запрос и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел формирует ARP-ответ, в котором указывает свой IP-адрес и свой локальный адрес и отправляет его уже направленно, так как в ARP запросе отправитель указывает свой локальный адрес. ARP-запросы и ответы используют один и тот же формат пакета. Так как локальные адреса могут в различных типах сетей иметь различную длину, то формат пакета протокола ARP зависит от типа сети. На рисунке 4 показан формат пакета протокола ARP для передачи по сети Ethernet.

|  |  |  |
| --- | --- | --- |
| **Тип сети** | | **Тип протокола** |
| Длина локального адреса | Длина сетевого адреса | Операция |
| Локальный адрес отправителя (байты 0 - 3) | |  |
| Локальный адрес отправителя (байты 4 - 5) | | IP-адрес отправителя (байты 0-1) |
| IP-адрес отправителя (байты 2-3) | | Искомый локальный адрес (байты 0 - 1) |
| Искомый локальный адрес (байты 2-5) | |  |
| Искомый IP-адрес (байты 0 - 3) | |  |

Рис. 4 Формат пакета протокола ARP

В поле типа сети для сетей Ethernet указывается значение 1. Поле типа протокола позволяет использовать пакеты ARP не только для протокола IP, но и для других сетевых протоколов. Для IP значение этого поля равно 080016.

Длина локального адреса для протокола Ethernet равна 6 байтам, а длина IP-адреса - 4 байтам. В поле операции для ARP запросов указывается значение 1 для протокола ARP и 2 для протокола RARP.

Узел, отправляющий ARP-запрос, заполняет в пакете все поля, кроме поля искомого локального адреса (для RARP-запроса не указывается искомый IP-адрес). Значение этого поля заполняется узлом, опознавшим свой IP-адрес.

В глобальных сетях администратору сети чаще всего приходится вручную формировать ARP-таблицы, в которых он задает, например, соответствие IP-адреса адресу узла сети X.25, который имеет смысл локального адреса. В последнее время наметилась тенденция автоматизации работы протокола ARP и в глобальных сетях. Для этой цели среди всех маршрутизаторов, подключенных к какой-либо глобальной сети, выделяется специальный маршрутизатор, который ведет ARP-таблицу для всех остальных узлов и маршрутизаторов этой сети. При таком централизованном подходе для всех узлов и маршрутизаторов вручную нужно задать только IP-адрес и локальный адрес выделенного маршрутизатора. Затем каждый узел и маршрутизатор регистрирует свои адреса в выделенном маршрутизаторе, а при необходимости установления соответствия между IP-адресом и локальным адресом узел обращается к выделенному маршрутизатору с запросом и автоматически получает ответ без участия администратора.

1. Принцип работы коммутаторов. Типы и характеристики коммутаторов. Таблица MAC – адресов.

# Назначение сетевого уровня. Типы адресов (сетевой, узловой, широковещательный; частные и публичные, логический интерфейс loopback,).

**Сетевой уровень.**

Это уровень 3, который управляет адресацией устройств, отслеживает расположение устройств в сети.

Он определяет наилучший путь для перемещения данных из источника в место назначения в зависимости от состояния сети, приоритета обслуживания и других факторов.

Канальный уровень передачи данных отвечает за маршрутизацию и пересылку пакетов.

Маршрутизаторы – это устройства уровня 3, они указаны на этом уровне и используются для предоставления услуг маршрутизации в пределах межсетевого взаимодействия.

Протоколы, используемые для маршрутизации сетевого трафика, называются протоколами сетевого уровня. Примерами протоколов являются IPV4 и Ipv6.

**Сетевой**

IP-адрес, состоящий из 4 байт, например, 109.26.17.100. Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации специального подразделения Internet (Network Information Center, NIC), если сеть должна работать как составная часть Internet. Обычно провайдеры услуг Internet получают диапазоны адресов у подразделений NIC, а затем распределяют их между своими абонентами.

**Широковещательная рассылка и broadcast IP-адреса**. Суть взаимодействия понятна из названия. Если unicast можно описать как взаимодействия точка-точка, то broadcast, как точка-многоточка. Сразу стоит обратить внимание на то, что широковещательный трафик ограничен канальной средой узла, в котором он находится.

Частные и публичные адреса

Адреса всех пользователей сети Internet должны быть уникальными. Первоначально уникальность адресов обеспечивал центр Internet Network Information Center (Inter NIC), на смену которому пришла Администрация адресного пространства Интернет (Internet Assigned Numbers Authority - IANA). IANA управляет IP-адресами, чтобы не произошло дублирования общедоступных (публичных) адресов, распределяя их между пятью Региональными Интернет регистраторами (Regional Internet Regiestry - RIR): ARIN (Северная Америка), RIPE (Россия, Европа, страны СНГ, Ближняя Азия), APNIC (Азия и Австралия), LACNIC (Латинская и Южная Америка), AfriNIC (Африка). Таким образом, все общедоступные (публичные) адреса должны быть зарегистрированы регистратором RIR, который выделяет адреса сетевым операторам и провайдерам, а те, в свою очередь, выделяет адреса сетевым администраторам и отдельным пользователям.

В связи с быстрым ростом Internet, существует дефицит публичных адресов IPv4. Радикально решить проблему дефицита IP-адресов может созданная новая шестая версия (IPv6) адресации в IP-сетях. Для смягчения проблемы нехватки публичных адресов IPv4были разработаны новые схемы адресации, такие как бесклассовая междоменная маршрутизация (CIDR) и адресация на основе масок переменной длины (VLSM).

Кроме того, проблему нехватки публичных адресов может в некоторой мере ослабить использование частных адресов (Private IP addresses). Сети с частными адресами, не подключенные к Internet, могут иметь любые адреса, лишь бы они были уникальны внутри частной сети. Выход в Интернет-пакетов с частными адресами блокируется маршрутизатором. Документ RFC 1918 устанавливает три блока частных адресов для использования внутри частных сетей

**Loopback** интерфейсы и loopback адреса – это не отдельный вид трафик, зачем я его сюда добавил? Да просто потому что могу, а почему бы и нет, создавать отдельную тему, чтобы рассказать про loopback просто не вижу смысла. Если говорить про loopback интерфейс, то это интерфейс, которого нет физически, но есть в голове у узла или маршрутизатора, этот интерфейс будет доступен другим физическим устройствам до тех пор, пока жив хотя бы один физический интерфейс узла, на котором создан loopback. Про loopback IP-адреса мы уже говорили, когда разбирались с видами IP-адресов.

# Характеристика IP протокола. Формат IPv4 пакета, назначение основных полей заголовка.

Основным протоколом является IP, который имеет две версии: IPv4 и IPv6. Основные характеристики протокола IPv4:

* Размер адреса узла - 4 байта
* В заголовке есть поле TTL
* Нет гарантии при доставке, что будет правильная последовательность
* Пакетная передача данных.
* Если превысится максимальный размер для пакета, тогда обеспечивается его фрагментация.

Версия состоящее из четырех бит поле, которое содержит в себе номер версии IP протокола (4 или 6).

Длина заголовка - состоящее их 4х бит поле, которое определяет размер заголовка пакета.

Тип обслуживания поле, которое состоит из 1 байта; на сегодняшний день не используется. Его заменяют на два других:

1. **DSCP**, которое делит трафик на классы обслуживания, размер его составляет 6 бит.
2. **ECN** - поле, состоящее из 2 бит, используется в случае, если есть перегрузка при передаче трафика.

Смещение фрагмента используется в случае фрагментации пакета, поле которого равно 13 бит. Должно быть кратно 8.

"**Время жизни**" поле, длиной в 1 байт, значение устанавливает создающий IP-пакет узел сети, поле, состоящее из 1 байта

Транспорт поле, размером в один байт.

Доп. данные заголовка поле, которое имеет произвольную длину в зависимости от содержимого и используется для спец. задач.

Данные выравнивания. Данное поле используется для выравнивания заголовка пакета до 4 байт.

IP уникальный адрес. Адреса протокола четвёртой версии имеют длину 4 байта, а шестой 16 байт. IP адреса делятся на классы (A, B, C). Сети, которые получаются в результате взаимодействия данных классов, различаются допустимым количеством возможных адресов сети. Для классов A, B и C адреса распределяются между идентификатором (номером) сети и идентификатором узла сети.

Классовая IP адресация — это метод IP-адресации, который не позволяет рационально использовать ограниченный ресурс уникальных IP-адресов, т.к. невозможно использование различных масок подсетей. В классовом методе адресации используется фиксированная маска подсети, поэтому класс сети (см. выше) всегда можно идентифицировать по первым битам.

Бесклассовая IP адресация (Classless Inter-Domain Routing — CIDR) — это метод IP-адресации, который позволяет рационально управлять пространством IP адресов. В бесклассовом методе адресации используются маски подсети переменной длины (variable length subnet mask — VLSM).

Возможные значения маскок подсети при бесклассовом методе адресации (широко применяется в современных сетях).



Версия (Version)

Самое первое поле в заголовке IP пакета – это версия, под него выделено четыре бита, для протокола IPv4 здесь всегда неизменное значение – 4. Хочу заметить, что в IPv4 четверка не связана с количество октетов в IP-адресе, просто такое совпадение.

Размер заголовка (Internet Header Length)

Поле размер заголовка нужно для того, чтобы маршрутизатор или конечный узел понимали: где заканчивается заголовок и начинаются данные. Также мы помним, что поле Опции не является обязательным, собственно, из-за этого и появилась необходимость в поле «Размер заголовка», под это поле выделено четыре бита и оно служит для указания количества слов в заголовке. Таким образом получается, что минимальный размер заголовка IP-пакет равен пяти словам, каждое слово 32 бита, следовательно, обычный заголовок без дополнительных опций равен 160 бит или 20 байт. Максимальное количество слов в заголовке равно пятнадцати.

Тип обслуживания (Type of Service), DSCP и ECN

На самом деле на данный момент поля Type of Service в IP заголовке нет, изначально поле ToS использовалось для указания приоритетов при обработке трафика, сейчас это поле заменено на два:

DSCP (Differentiated Services Code Point), под которое выделено 6 бит, это поле используется для разделения трафика на классы обслуживания. Сейчас мы не будем сильно вдаваться в подробности, поскольку на самом деле это очень сложная тема, в которой даже инженеры с опытом могут вполне себе неплохо плавать.

Не трудно посчитать, что для второго поля у нас остается два бита, и это поле называется ECN (Explicit Congestion Notification) или указатель перегрузки. Как понятно из названия, это поле может пригодится в тех ситуациях, когда [пропускная способность канала связи](https://zametkinapolyah.ru/kompyuternye-seti/korotko-o-edinicax-izmereniya.html) меньше, чем трафик, который в текущий момент передается по каналу ([виды взаимодействия в компьютерных сетях](https://zametkinapolyah.ru/kompyuternye-seti/sovmestnye-resursy.html)). Это поле будет задействовано только в том случае, когда передающая и принимающая сторона умеют с ним работать.

Размер пакета (Total Length)

Это поле позволяет обрабатывающему устройству понять полный размер пакета, то есть заголовок плюс данные. Минимальный размер IP-пакета равен 20 байт, то есть это заголовок без опций и данных, а максимальный размер равен 65535 байт. Тут стоит заметить, что одним из параметров канала связи является максимально возможный размер пакета ([субъективные и объективные характеристики компьтюерной сети](https://zametkinapolyah.ru/kompyuternye-seti/osnovnye-xarakteristiki-kompyuternoj-seti.html)), который по этому каналу можно передавать. И, например, бывают ситуации, когда в канале связи можно передавать пакеты, размером, скажем (точные цифры сейчас не очень важны) 1000 байт, а узел генерирует пакеты размером 2000 байт, в этом случае перед отправкой данных в канал связи пакеты будут фрагментироваться, то есть разбиваться на более мелкие, это поведение по умолчанию в протоколе IP.

Идентификатор (Identification)

Чаще всего это поле используется в тех ситуация, когда пакет фрагментируется, чтобы принимающая сторона понимала, как из полученных кусочков правильно собрать пакет. У фрагментированных пакетов значение в этом поле должны быть одинаковыми.

Флаги (Flags)

Под поле флаги выделено три бита, этих три бита используются для контроля над фрагментацией пакетов. Немного отвлечемся непосредственно от флагов и заметим, что нумерация бит в поле начинается с нуля, крайний левый бит старший, а крайний правый – младший. Тогда у нас в поле Флаги получается следующая картина:

нулевой бит зарезервирован и должен быть всегда равен нулю;

если значение первого бита ноль, то допускается фрагментация пакетов, если единица (бит DF или Do not Fragment), то устройства компьютерной сети не будут выполнять фрагментацию;

второй бит служит для того, чтобы конечные узлы понимали, где начинается последовательность фрагментированных пакетов, а где она заканчивается, если значение этого бита равно единице (MF More Fragments), то узел понимает, что этот пакет не последний и нужно ждать еще пакеты, чтобы собрать изначально разделенный пакет.

Смещение фрагмента (Fragment Offset)

Это поле используется в тех случаях, когда выполняется фрагментация пакетов, размер этого поля равен 13 бит. Нетрудно посчитать, что максимально возможное значение, которое можно записать в этом поле равно два в тринадцатой степени или 65528, смещение задается в байтах, то есть это поле говорит узлу или маршрутизатору на сколько байт нужно выполнять смещение от нуля (это для пакетов без фрагментации, в этом случае все тринадцать бит имеют значение 0) до 65528 байт (в этом случае все тринадцать бит выставлены в единицу). В том случае, если фрагментация выполняется, то первый пакет последовательности также имеет смещение равное нулю. У числовых значений, записываемых в поле «Смещение фрагмента» есть одно ограничение — это значение должно быть всегда кратно восьми.

# Назначение маски подсети. Классовая и бесклассовая IP-адресация. Маски переменной длины.

Маска подсети — битовая маска для определения по IP-адресу адреса подсети и адреса узла (хоста, компьютера, устройства) этой подсети. В отличие от IP-адреса маска подсети не является частью IP-пакета.

Благодаря маске можно узнать, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети.

Например, узел с IP-адресом 12.34.56.78 и маской подсети 255.255.255.0 находится в сети 12.34.56.0 с длиной префикса 24 бита. В случае адресации IPv6 адрес 2001:0DB8:1:0:6C1F:A78A:3CB5:1ADD с длиной префикса 32 бита (/32) находится в сети 2001:0DB8::/32.

Каждый IP-адрес состоит из двух частей. Как узлы определяют, где сетевая часть, а где адрес узла? Для этого используется маска подсети.

При настройке IP узлу присваивается не только IP-адрес, но и маска подсети. Как и IP-адрес, маска состоит из 32 бит. Она определяет, какая часть IP-адреса относится к сети, а какая – к узлу.

Маска сравнивается с IP-адресом побитно, слева направо. В маске подсети единицы соответствуют сетевой части, а нули — адресу узла.

Отправляя пакет, узел сравнивает маску подсети со своим IP-адресом и адресом назначения. Если биты сетевой части совпадают, значит, узлы источника и назначения находятся в одной и той же сети, и пакет доставляется локально. Если нет, отправляющий узел передает пакет на интерфейс локального маршрутизатора для отправки в другую сеть.

В домашних офисах и небольших компаниях чаще всего встречаются следующие маски подсети: 255.0.0.0 (8 бит), 255.255.0.0 (16 бит) и 255.255.255.0 (24 бита). В маске подсети 255.255.255.0 (десятичный вариант), или 11111111.11111111.1111111.00000000 (двоичный вариант) 24 бита идентифицируют сеть, а 8 — узлы в сети.

Чтобы вычислить количество возможных сетевых узлов, нужно взять число два (2) в степени количества отведенных для них бит (2 ^ 8 = 256). Из полученного результата необходимо вычесть 2 (256-2). Дело в том, что состоящая из одних единиц (1) отведенная узлам часть IP-адреса предназначена для адреса широковещательной рассылки и не может принадлежать одному узлу. Часть, состоящая только из нулей, является идентификатором сети и тоже не может быть присвоена конкретному узлу. Возвести число 2 в степень без труда можно с помощью калькулятора, который есть в любой операционной системе Windows.

Классовая IP адресация — это метод IP-адресации, который не позволяет рационально использовать ограниченный ресурс уникальных IP-адресов, т.к. невозможно использование различных масок подсетей. В классовом методе адресации используется фиксированная маска подсети, поэтому класс сети (см. выше) всегда можно идентифицировать по первым битам.

Бесклассовая IP адресация (Classless Inter-Domain Routing — CIDR) — это метод IP-адресации, который позволяет рационально управлять пространством IP адресов. В бесклассовом методе адресации используются маски подсети переменной длины (variable length subnet mask — VLSM).

Возможные значения маскок подсети при бесклассовом методе адресации (широко применяется в современных сетях):

Термин маска подсети переменной длины (variable-length subnet mask — VLSM) означает, что одна сеть может быть сконфигурирована с различными масками. Основная идея применения VLSM3 заключается в предоставлении большей гибкости при разбиении сети на несколько подсетей, т.е. для оптимального распределения допустимого количества хостов в различных подсетях. Без VLSM для всей сети может использоваться только одна маска подсети. Тогда количество хостов в подсетях будет строго ограничено. Если же вы выберете маску, которая предоставит нужное количество подсетей, то, возможно, вам будет недостаточно допустимого количества хостов для каждой подсети. Та же ситуация справедлива и для хостов, т.е. маска, обеспечивающая достаточное количество хостов, ограничивает вас в числе подсетей. Маски переменной длины предоставляют возможность выделять подсети с различным количеством хостов в них, что позволяет сетевому администратору более эффективно использовать доступное адресное пространство.

Допустим для примера, что вам выделена сеть класса С с адресом 192.214.11.0, и

требуется разделить ее на три подсети. В одной подсети должно быть около 100 хостов, а в двух других — около 50 хостов в каждой. Исключая два адреса, 0 (номер сети) и 255 (широковещательный адрес для сети) вам теоретически доступно 256 адресов хостов для сети класса С, т.е. с 192.214.11.0 до 192.214.11.255. Как видите разбить такую сеть на подсети с требуемым количеством хостов без использования VLSM невозможно.

Чтобы определить параметры подсети в сети 192.214.11.0, сначала необходимо определить маску сети, которая для обычной сети класса С будет представлена в виде 255.255.255.0 (все биты равны 1 в первых трех октетах). Для разделения сети класса С с адресом 192.214.11.0 на подсети можно использовать несколько масок вида 255.255.255.Х. Маска, начиная со старшего (самого левого) бита, должна иметь непрерывный ряд единиц и оканчиваться нулями.

# Протокол DHCPv4. Основные настраиваемые параметры DHCPv4 сервера.

DHCP - это протокол TCP/IP, автоматизирующий присвоение IP-адресов. (Название "автоматическое присвоение IP-адресов", Automatic IP Address Assignment, может, и лучше отражает суть, но AIAA больше похоже не на сокращение, а на вопль, издаваемый сетевым администратором от безысходности). Для использования протокола TCP/IP в сети администратор должен задать для каждого из компьютеров по меньшей мере три параметра - IP-адрес, маску подсети и адрес используемого по умолчанию шлюза. При этом каждый компьютер должен иметь уникальный IP-адрес. Кроме того, присвоенный адрес должен находиться в диапазоне подсети, к которой подключено устройство. В большой сети иногда бывает трудно определить, к какой же из подсетей подключен тот или иной компьютер. Однако DHCP "знает", из какой подсети приходит запрос на получение IP-адреса, и сделает за вас все как надо. Если в сети используются Windows Internet Naming Service (WINS) и Domain Name Service (DNS), то на каждом из клиентских компьютеров администратору необходимо также указать IP-адреса WINS и DNS-серверов.

16. Администратор может сконфигурировать каждую из систем вручную или попросить сделать это пользователей, предоставив им необходимые данные. Однако последний подход слишком рискован. Самый простой и безопасный способ - сконфигурировать один или несколько DHCP-серверов так, чтобы они автоматически присваивали IP-адреса каждому компьютеру в сети. Для этого вам достаточно сконфигурировать сервер, ввести диапазоны адресов, настроить несколько дополнительных параметров и периодически осуществлять мониторинг.

DHCP-сервер в ИКС позволяет:

1. Добавить связь IP и MAC адресов (используется ARP-таблица).

2. Задать срок аренды адреса, по истечении которого ip, закрепленные за клиентским MAC, могут использоваться другими устройствами.

3. Автоматически создавать DNS-записи для каждого компьютера в сети и обращаться к ним по имени, а не по ip-адресу.

4. Отслеживать количество используемых и свободных адресов по каждой сети в DHCP.

5. Закрепить ip-адрес за определенным пользователем или устройством.

6. Отслеживать пользователей, получивших ip-адрес от DHCP-сервера.

7. Указать глобальный поисковый домен или поисковый домен для каждой локальной сети.

8. Экспортировать данные журнала за определенный период или удалить их.

9. Задать WINS.

10. Сконфигурировать путь до TFTP-сервера.

# Ограничения протокола IPv4. Протокол IPv6. Формат заголовка IPv6.

На протяжении многих лет протокол IPv4 периодически обновлялся для решения новых задач. Тем не менее, даже в результате изменений IPv4 по-прежнему имеет три основных недостатка.

• Нехватка IP-адресов. IPv4 может предложить лишь ограниченное количество уникальных общедоступных IP-адресов. Несмотря на то, что существует примерно 4 миллиарда IPv4-адресов, возросшее число новых устройств, в которых используется протокол IP, а также потенциальный рост менее развитых регионов привели к необходимости дополнительно увеличить количество адресов.

• Расширение таблицы интернет-маршрутизации. Таблица маршрутизации используется маршрутизаторами для определения оптимальных путей пересылки данных. По мере увеличения количества серверов (узлов), подключённых к Интернету, также растет число сетевых маршрутов. Эти маршруты IPv4 потребляют значительное количество памяти и ресурсов процессоров интернет-маршрутизаторов.

• Нехватка сквозных соединений. Преобразование сетевых адресов (NAT) представляет собой технологию, которая обычно применяется в сетях IPv4. NAT позволяет различным устройствам совместно использовать один публичный IP-адрес. При этом, поскольку публичный IP-адрес используется совместно, IP-адрес узла внутренней сети скрыт. Это может представлять проблему при использовании технологий, для которых необходимы сквозные подключения.

• Нехватка IP-адресов. IPv4 может предложить лишь 4.3 миллиарда IP-адресов.  CIDR, VLSM, NAT помогли продлить жизнь IPv4.  Отсутствие сквозных соединений. • NAT представляет проблему для сквозных соединений. • NAT увеличивает нагрузку на маршрутизвторы и замедляет процесс передачи пакетов.  Расширение таблицы интернет-маршрутизации. • Разбиение на подсети ведет к росту количества сетей и увеличению таблиц маршрутизации • Обработка больших таблиц требует больше ресурсов маршрутизаторов

Краткое описание IPv6  Увеличенное пространство адресов  340 ундециллионов IPv6-адресов 340 000 000 000 000 000 000 000 000 000 000 000 000  Улучшенная обработка пакетов за счет более простого заголовка  Отсутствие необходимости в использовании NAT  Автоконфигурация адресов  Поддержка больших IPv6 пакетов  Встроенная система безопасности Presentation\_ID © Корпорация Cisco Systems, 2014. Все права защищены. Конфиденциальная информация корпорации Cisco 7 Повышение производительности маршрутизаторов  Перенесение функции фрагментации с маршрутизаторов на конечные узлы.  Уменьшение таблиц маршрутизации.  Отказ от обработки не обязательных полей IPv6 заголовка и введение специального поля «Метка потока.»  Использование маршрутизации от источника. • Отпадает необходимость просмотра таблиц маршрутизации на промежуточных маршрутизаторах.

В интернет-протоколе нового поколения IPv6 для создания адресной маршрутизации используется 128-битная система записи. В IPv6-адресе записи представляют собой восемь 16-битных блоков, разделенных двоеточиями: 2dfc:0:0:0:0217:cbff:fe8c:0. Общее количество ip-адресов, возможных для распределения, может составить в общей сложности 2128 (приблизительно 340 282 366 920 938 000 000 000 000 000 000 000 000). Повсеместное использование данного стандарта позволит полностью решить задачу нехватки сетевых адресов в обозримом будущем.

С целью упрощения записи адреса в протоколе IPv6 используется вариант сжатия кода, когда смежные последовательности нулевых блоков заменяются парами символов двоеточия. Например, адрес групповой рассылки FFEA:0:0:0:0:CA28:1012:4254 в сжатой форме будет представлен в укороченном виде FFEA::CA28:1012:4254. Данный механизм упрощает процесс записи, хранения и обработки кода.

По правилам протокола IPv6 назначение сетевых адресов происходит автоматически и уникализируется за счет идентификации на уровне MAC-адреса конкретной единицы оборудования, для которой необходим выход в публичную сеть. Другими словами, каждый домашний компьютер, смартфон, холодильник или стиральная машина с функцией подключения к внешним устройствам получает собственный «белый» ip-адрес для коннекта с другими хостами через интернет. Доступна также произвольная генерация кодов путем администрирования с использованием маршрутизаторов.

формат заголовка пакета 1Ру6

Так же как и в 1Р'4 пакет протокола 1Ру6 имеет заголовок и поле данных. КРС-2460[1] (ранее — ЕРС-1883[2]) определяет формат заголовка

IPv6, длина которого составляет 40 байт. Структура заголовка приведена на рис. 5.7.

Версия протокола IP указывается в ноле Версия ( Version) длиной 4 бита.

Для работы с классами трафика и приоритетами передаваемых пакетов используется ноле Класс трафика (Traffic Class) длиной 8 бит.

Для идентификации специально обрабатываемого впоследствии передаваемого потока данных используется поле Метка потока (Flow Label) длиной 20 бит. Под специальной обработкой потока в данном случаеВ понимается наложение ограничений, например, на полосу пропусканияВ или величину задержки при передаче через промежуточные сетевые узлы.В Обычно в качестве метки потока выступает некоторое псевдослучайноеВ число, воспринимаемое сетевыми шлюзами как хеш-ключ, указывающий,В как именно должен быть обработан этот поток.

Рис. 5.7. Формат заголовка пакета IPv6 (RFC-2460)

После заголовка пакета следует поле данных, размер которого в байтах задается в ноле Размер поля данных (Payload Length) длиной 16 бит. В случае, если размер поля данных превышает 65 535 байт, то значение в полеВ Размер поля данных становится равным нулю, собственно размер поля данных указывается в иоле Сверхдлина (Jumbo Payload) дополнительного заголовка опций Нор-Ьу-Нор, о котором речь пойдет позже.

Протокол IPv6 может иметь дополнительные заголовки, предназначенные для выполнения определенных функций. Идентификатор типа дополнительного заголовка указывается в поле Следующий заголовок (NextВ Header) длиной 8 бит.

Аналогом поля времени жизни протокола IPv4 в IPv6 служит поле Максимальное число транзитных узлов (Нор Limit) длиной 8 бит. В нем задается предельный срок, в течение которого пакет может перемещаться по сети.В Значение этого поля уменьшается на 1, когда пакет проходит через промежуточный узел сети (например, шлюз или хост). При достижении нулевогоВ значения этого поля пакет уничтожается.

Адреса отправителя и получателя указываются в соответствующих полях IP-адрес источника (Source IP Address) и IP-адрес назначения (Destination IP Address) длиной по 128 бит.

# Типы адресов IPv6. Формат глобального индивидуального адреса.

IPv6-адреса можно классифицировать по способу адресации: одноадресные (Unicast), Anycast и групповые (Multicast)[1].

• Unicast адреса идентифицируют только один сетевой интерфейс. Протокол IPv6 доставляет пакеты, отправленные на такой адрес, на конкретный интерфейс.

• Anycast адреса назначаются группе интерфейсов, обычно принадлежащих различным узлам. Пакет, отправленный на такой адрес, доставляется на один из интерфейсов данной группы, как правило наиболее близкий к отправителю с точки зрения протокола маршрутизации.

• Multicast адрес также используется группой узлов, но пакет, отправленный на такой адрес, будет доставлен каждому узлу в группе.

В IPv6 не реализованы широковещательные адреса. Традиционная роль широковещательной рассылки реализована с помощью групповой рассылки на адрес ff02::1, однако использование этой группы не рекомендуется.

Индивидуальный адрес служит для определения интерфейса устройства под управлением протокола IPv6. Пакет, который отправляется на индивидуальный адрес, будет получен интерфейсом, присвоенным для этого адреса. Как и в случае с протоколом IPv4, IPv6-адрес должен быть индивидуальным. IPv6-адрес назначения может быть как индивидуальным, так и групповым.

Существует шесть типов индивидуальных IPv6-адресов:

Глобальный индивидуальный адрес

Глобальный индивидуальный адрес мало чем отличается от публичного IPv4-адреса. Эти адреса, к которым можно проложить маршрут по Интернету, являются уникальными по всему миру. Глобальные индивидуальные адреса могут быть настроены статически или присвоены динамически. В динамическом назначении IPv6-адреса устройством имеются некоторые важные отличия по сравнению с динамическим назначением IPv4-адреса.

Локальный адрес канала

Локальные адреса канала используются для обмена данными с другими устройствами по одному локальному каналу. В протоколе IPv6 термин «канал» означает подсеть. Локальные адреса каналов ограничены одним каналом. Они должны быть уникальны только в рамках этого канала, поскольку вне канала к ним нельзя проложить маршрут. Другими словами, маршрутизаторы не смогут пересылать пакеты, имея локальный адрес канала источника или назначения.

Логический интерфейс loopback

Loopback-адрес используется узлом для отправки пакета самому себе и не может быть назначен физическому интерфейсу. Как и на loopback-адрес IPv4, для проверки настроек TCP/IP на локальном узле можно послать эхо-запрос на loopback-адрес IPv6. Loopback-адрес IPv6 состоит из нулей, за исключением последнего бита, который выглядит как ::1/128 или просто ::1 в сжатом формате.

Неопределённый адрес

Неопределённый адрес состоит из нулей и в сжатом формате представлен как ::/128 или просто :: Он не может быть назначен интерфейсу и используется только в качестве адреса источника в IPv6-пакете. Неопределённый адрес используется в качестве адреса источника, когда устройству еще не назначен постоянный IPv6-адрес или когда источник пакета не относится к месту назначения.

Уникальный локальный адрес

Уникальные локальные IPv6-адреса имеют некоторые общие особенности с частными адресами RFC 1918 для IPv4, но при этом между ними имеются и значительные различия. Уникальные локальные адреса используются для локальной адресации в пределах узла или между ограниченным количеством узлов. Эти адреса не следует маршрутизировать в глобальном протоколе IPv6. Уникальные локальные адреса находятся в диапазоне от FC00::/7 до FDFF::/7.

В случае с IPv4 частные адреса объединены с преобразованием сетевых портов и адресов (NAT/PAT) для обеспечения преобразования адресов из частных в публичные. Это делается из-за недостатка адресного пространства IPv4. На многих сайтах также используют частный характер адресов RFC 1918, чтобы обеспечить безопасность или защитить сеть от потенциальных угроз. Однако такая мера никогда не была целью использования данных технологий, и организация IETF всегда рекомендовала предпринимать правильные меры предосторожности при работе маршрутизатора в Интернете. Хотя протокол IPv6 обеспечивает особую адресацию для сайтов, он не предназначен для того, чтобы скрывать внутренние устройства под управлением IPv6 от Интернета IPv6. IETF рекомендует ограничивать доступ к устройствам с помощью наилучших мер безопасности.

Примечание. Исходная спецификация IPv6 определяет локальные адреса для тех же целей с помощью диапазона префикса FEC0:: /10. В спецификации были обнаружены некоторые неточности, и локальные адреса сайтов были запрещены IETF в пользу уникальных локальных адресов.

Встроенный IPv4

Последними из рассматриваемых типов индивидуальных адресов являются встроенные IPv4-адреса. Использование этих адресов способствует переходу с протокола IPv4 на IPv6.

# Правила сокращенной записи IPv6-адресов.

Адрес протокола IPv6 состоит из 128 бит и записывается обычно в шестнадцатеричном виде. Адрес разбиваются на блоки по 16 бит (хекстеты) и каждый блок представляется четырьмя шестнадцатеричными цифрами. Хекстеты разделяются знаком двоеточия. Таким образом, адрес получается достаточно длинным – он состоит из 32 шестнадцатеричных цифр и 7 знаков двоеточия.

Пример IPv6 адреса 2001:0DB0:0000:123A:0000:0000:0000:0030. Запись весьма длинная, чтобы немного упростить процесс записи адресов, используют сокращенную запись. Для того чтобы сократить данный адрес надо последовательно применить два правила.

Правило 1

В каждом хекстете (группе из 4-х цифр) ведущие нули удаляются. Например, во втором хекстете 0DB0 заменяется на DB0. То есть ноль слева удаляется, ноль справа мы не трогаем. Если хекстет состоит из одних нулей, то он заменяется на один нуль. Таким образом адрес 2001:0DB0:0000:123A:0000:0000:0000:0030 преобразуется в 2001:DB0:0:123A:0:0:0:30. А, например, адрес loopback 0000:0000:0000:0000:0000:0000:0000:0001 заменяется на 0:0:0:0:0:0:0:1.

Правило 2

Это правило применяется только после первого. В адрес выбирается одна самая длинная группа, состоящая из полностью нулевых хекстетов, то есть самая длинная последовательность «:0:0:0:» и заменяется на два двоеточия «::». Эту замену можно произвести только один раз и только с самой длинной последовательностью, так как, если бы мы, например, сделали такую замену в двух местах адреса, то потом нельзя было бы восстановить, сколько именно хекстетов мы заменили в первом и во втором случае. Важный момент: нельзя заменять одну группу из :0: на ::, правило два применимо только если есть более одной нулевой группы.

Для примера возьмём адрес из предыдущей замены 2001:DB0:0:123A:0:0:0:30. Самая длинная последовательность из полностью пустых хекстетов – это «:0:0:0:», она начинается сразу после хекстета «123A». Есть ещё последовательность из одного пустого хекстета (между «DB0» и «123A»), но эта – длиннее, так что заменять будем её. Адрес станет совсем небольшим: 2001:DB0:0:123A::30 конечно, длиннее IPv4 адреса, но гораздо короче исходного.

Получение исходного адреса по сокращённой записи

Эта процедура достаточно тривиальна, если мы уже умеем сокращать адреса.

Сначала надо посчитать, сколько хекстетов в адресе осталось. В нашем случае, в адресе 2001:DB0:0:123A::30 осталось 5 хекстетов. Мы знаем, что адрес должен состоять из восьми хекстетов – значит вместо «::» возвращаем три недостающих нулевых, получаем 2001:DB0:0:123A:0:0:0:30. Теперь в каждой группе, где меньше четырёх цифр дописываем слева такое количество нулей, чтобы в группе стало четыре цифры. В результате получим исходный адрес 2001:0DB0:0000:123A:0000:0000:0000:0030.

Примеры

Теперь, чтобы закрепить понимание, приведём несколько примеров сокращения адресов. Сокращать будем по правилам в два этапа.

1. FF80:0000:0000:0000:0123:1234:ABCD:EF12 → FF80:0:0:0:123:1234:ABCD:EF12 → FF80::123:1234:ABCD:EF12

2. FF02:0000:0000:0000:0000:0001:FF00:0300 → FF02:0:0:0:0:1:FF00:300 → FF02::1:FF00:300

3. 2001:0DB8:0000:1111:0000:0000:0000:0200 → 2001:DB8:0:1111:0:0:0:200 → 2001:DB8:0:1111::200

4. 0000:0000:0000:0000:0000:0000:0000:0001 → 0:0:0:0:0:0:0:1 → ::1

5. 0000:0000:0000:0000:0000:0000:0000:0000 → 0:0:0:0:0:0:0:0 → ::

Адрес loopback выглядит в сокращённой записи особенно элегантно ::1. Даже если вы не пользуетесь IPv6, но работаете на одной из современных операционных, систем, у вас наверняка установлен этот протокол. Это легко проверить, пропинговав loopback.

# Автоматическое назначение IPv6 адресов SLAAC. Варианты назначения узловой части IPv6-адреса. Использование механизма EUI-64.

SLAAC — это способ, который позволяет устройству получить свой префикс, длину префикса и адрес шлюза по умолчанию от маршрутизатора IPv6 без помощи DHCPv6-сервера. При использовании SLAAC для получения необходимой информации устройства полагаются на сообщения «Объявления маршрутизатора ICMPv6».

IPv6-маршрутизаторы периодически отправляют сообщения «Объявления маршрутизатора ICMPv6» всем устройствам в сети под управлением IPv6. По умолчанию маршрутизаторы Cisco отправляют такие сообщения каждые 200 секунд на адрес групповой передачи всем IPv6-узлам. IPv6-устройству, находящемуся в сети, не нужно ждать этих периодических сообщений. Устройство может отправить сообщение «Запрос маршрутизатора ICMPv6», который использует адрес групповой передачи всем IPv6-узлам. Когда маршрутизатор IPv6 получает такое сообщение, он сразу же отправляет в ответ объявление маршрутизатора.

IPv6-маршрутизация не включена по умолчанию. Чтобы маршрутизатор работал как IPv6-маршрутизатор, необходимо использовать команду глобальной конфигурации ipv6 unicast-routing.

Сообщение «Объявления маршрутизатора ICMPv6» содержит префикс, длину префикса и другие сведения IPv6-устройства. Кроме того, такое сообщение указывает IPv6-устройству, как ему получить информацию по адресации. Сообщение «Объявления маршрутизатора» может выглядеть в одном из следующих 3 вариантов.

Вариант 1: только SLAAC. Устройство должно использовать префикс, длину префикса и шлюз по умолчанию, которые содержатся в сообщении «Объявления маршрутизатора». Другая информация недоступна с DHCPv6-сервера;

Вариант 2: SLAAC и DHCPv6. Устройство должно использовать префикс, длину префикса и шлюз по умолчанию, которые содержатся в сообщении «Объявления маршрутизатора». На DHCPv6-сервере доступна и другая информация, например адрес DNS-сервера. Устройство получит эту дополнительную информацию в процессе поисков и запросов к DHCPv6-серверу. Этот процесс называется «DHCPv6 без запоминания состояний», поскольку DHCPv6-серверы не выделяют и не отслеживают какие-либо назначения IPv6-адресов, а предоставляют дополнительную информацию, например об адресе DNS-сервера;

Вариант 3: только DHCPv6. Устройство не должно использовать информацию из сообщения «Объявления маршрутизатора» для пополнения своей информации об адресации. Вместо этого устройство будет использовать обычные процессы поисков и запросов к DHCPv6-серверам для получения всей своей информации об адресации. Такая информация включает в себя индивидуальный адрес IPv6, длину префикса, адрес шлюза по умолчанию и адреса DNS-серверов. В этом случае DHCPv6-сервер работает как DHCP-сервер, который фиксирует данные аналогично DHCP-серверу для IPv4. DHCPv6-сервер выделяет и отслеживает IPv6-адреса, чтобы не назначать один и тот же IPv6-адрес на нескольких устройствах.

Вариант 1 (только SLACC): «Я предоставляю все необходимые вам данные (префикс, длину префикса и шлюз по умолчанию)».

Вариант 2 (SLACC и DHCPv6): «Вот моя информация, но вам понадобятся и другие данные, например, об DNS-адресах с сервера DHCPv6».

Вариант 3 (только DHCPv6): «Я не могу вам помочь. Запросите необходимую информацию у сервера DHCPv6».

Общие сведения о SLAAC

Автоматическая настройка адреса без отслеживания состояния (SLAAC) — это способ получения устройством глобального IPv6-адреса одноадресной рассылки без использования DHCPv6-сервера. В основе SLAAC лежит протокол ICMPv6. Протокол ICMPv6 аналогичен ICMPv4, но при этом он имеет дополнительные функциональные возможности и демонстрирует большую устойчивость к ошибкам. SLAAC использует ICMPv6-сообщения запроса маршрутизатора и объявления маршрутизатора, чтобы предоставить информацию об адресации и другую информацию о конфигурации, обычно предоставляемую DHCP-сервером.

Сообщение запроса маршрутизатора (RS) — если клиент настроен на получение информации об адресации автоматически с использованием SLAAC, он посылает на маршрутизатор сообщение RS. Сообщение RS отправляется на IPv6-адрес многоадресной рассылки FF02::2, который поддерживают все маршрутизаторы.

Сообщение объявления маршрутизатора (RA) — для предоставления информации об адресации маршрутизатор отправляет сообщения RA клиентам, настроенным на получение IPv6-адресов автоматически. Сообщение RA содержит префикс и длину префикса локального сегмента. Эта информация используется клиентом для создания собственного глобального индивидуального IPv6-адреса. Маршрутизатор передаёт сообщение RA периодически или в ответ на сообщение RS. По умолчанию маршрутизаторы Cisco отправляют сообщения RA каждые 200 секунд. Сообщения RA всегда отправляются на общий для всех узлов IPv6-адрес многоадресной рассылки FF02::1.

Как видно из термина, SLAAC не отслеживает состояние адреса. Служба без отслеживания состояния говорит о том, что ни один из серверов не поддерживает информацию о сетевом адресе. В отличие от сервера DHCP, сервер SLAAC не знает, какие IPv6-адреса используются, а какие доступны.

На рисунке 3 показан принцип работы SLACC + DHCPV6 без отслеживание состояния.



Рис.3. SLACC + DHCPV6 без отслеживания состояния

EUI-64, генерация случайным образом.

Поскольку SLAAC — это процесс без отслеживания состояния, перед использованием PC1 этого вновь созданного IPv6-адреса, необходимо проверить его уникальность., PC1 посылает по протоколу ICMPv6 сообщение запроса поиска соседа с собственным адресом в качестве IPv6-адреса назначения. Если другие устройства не отвечают сообщением запроса поиска соседа, значит, адрес является уникальным и может быть использован PC1. Если сообщение запроса поиска соседей получено PC1, значит, адрес не уникален, и операционная система должна установить новый идентификатор интерфейса для использования.

Этот процесс является частью процесса обнаружения соседних устройств ICMPv6 и известен как обнаружение адресов-дубликатов (DAD).

Настроен ли клиент на автоматическое получение информации об IPv6-адресации с использованием SLAAC, DHCPv6 или сочетанием обоих вариантов, зависит от настроек, содержащихся в сообщении RA. ICMPv6 сообщения RA содержат два флага, обозначающих, какой из вариантов должен быть использован клиентом.

Этими флагами являются флаг управляемой конфигурации адресов (M) и флаг другой конфигурации (O).

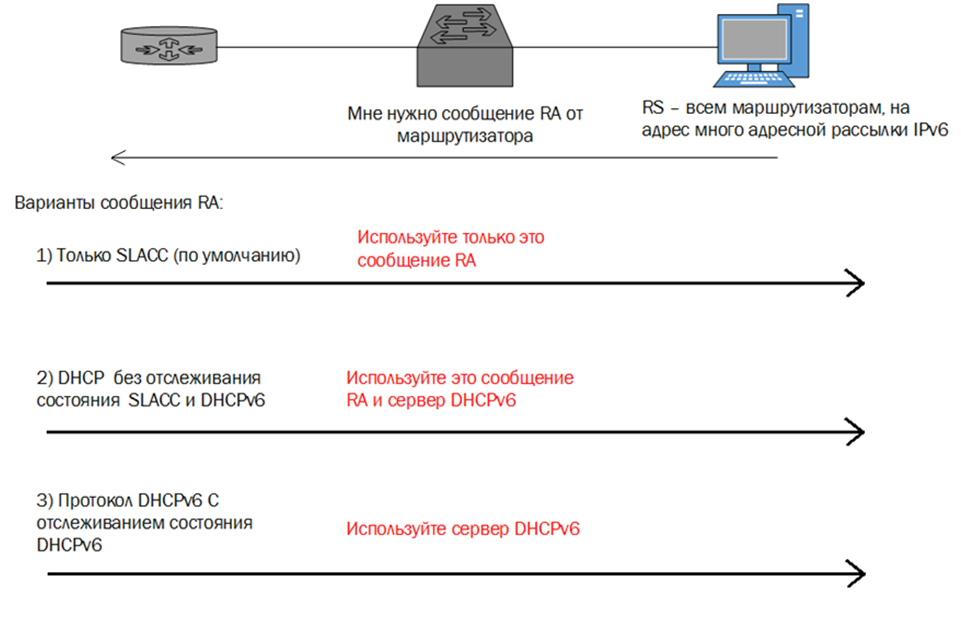


Рис.3.1. Варианты сообщений от RA

Функции SLAAC, DHCPV6+SLAAC, DHCPV6

Рассмотрим все три способа, изменение флагов.

1. SLAAC — Этот вариант указывает клиенту использовать только информацию из сообщения RA. Сюда входит информация о префиксе, длине префикса, DNS-сервере, MTU и информация о шлюзе по умолчанию. Далее клиент не получает никакой информации от сервера DHCPv6. Глобальный индивидуальный IPv6-адрес создаётся путём объединения префикса, полученного в сообщении RA, и идентификатора интерфейса, полученного с помощью EUI-64 или сгенерированного случайным образом.

Сообщения RA настроены на отдельном интерфейсе маршрутизатора. Для повторной активации режима SLAAC на интерфейсе, на котором мог быть установлен другой вариант работы, флаги M и O необходимо сбросить на их первоначальные значения, равные 0. Для этого применяются следующие команды режима конфигурации интерфейса:

2. DHCPV6+SLAAC — Для DHCPv6 без отслеживания состояния значение флага O установлено равным 1, а значение флага M остается со значением по умолчанию, равным 0. Значение флага O, равное 1, используется для информирования клиента о том, что на DHCPv6-сервере без отслеживания состояния доступна дополнительная информация о конфигурации.

Для того чтобы изменить сообщение RA, отправляемое на интерфейс маршрутизатора для указания использования DHCPv6 без отслеживания состояния, используйте следующие команды:

3. Протокол DHCPv6 с отслеживанием состояния (только DHCPv6)

Флаг M указывает, используется ли DHCPv6 с отслеживанием состояния. Флаг O не используется. Для того чтобы изменить значение флага М с 0 на 1 для объявления DHCPv6 с отслеживанием состояния, применяются следующие команды:

# Процесс принятия решения о пересылке пакетов сетевым узлом. Шлюз по умолчанию. Продвижение пакетов в сети.

Шлюз по умолчанию (англ. Default gateway) — в маршрутизируемых протоколах — 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). 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».

Устройство, обеспечивающее соединение разнородных сетей (обычно с различными протоколами передачи информации либо разной средой передачи). Он управляет передачей информации из одной сети в другую, что позволяет разделять потоки информации, идущие из одной сети в другую, и потоки, не выходящие за пределы одной сети. В сетях TCP/IP роль шлюза, как правило, выполняет маршрутизатор, соединяющий одну сеть с другой для их взаимодействия.

Решение о том, на какой интерфейс передать пришедший пакет, принимается на основании одного из трех методов продвижения пакетов:

При дейтаграмме передаче соединение не устанавливается, и все передаваемые пакеты продвигаются (передаются от одного узла сети другому) независимо друг от друга на основании одних и тех же правил. Процедура обработки пакета определяется только значениями параметров, которые он несет в себе, и текущим состоянием сети (напр, в зависти от ее нагрузки пакет может стоять в очереди на обслуживание большее или меньшее время). Однако никакая инфа об уже переданных пакетах сеть не хранится и в ходе обработки очередного пакета во внимание не принимается. Т.е. каждый отдельны пакет рассматривается сетью как совершенно независимая единица передачи – дейтаграмма.

Передача с установлением логического соединения распадается на так называемые сеансы, или логические соединения. Процедура обработки определяется не для отдельного пакета, а для всего множества пакетов, передаваемых в рамках каждого соединения. Для того чтобы реализовать дифференцированное обслуживание пакетов, принадлежащих разным соединениям, сеть должна, во-первых, присвоить каждому соед. идентификатор, во-вторых, запомнить параметры соединения, т.е. значения, определяющие процедуру обработки пакетов в рамках данного соединения. Эта инфа наз-ся инфой о состоянии соединения. Фиксированный маршрут не является обязательным параметром соединения. Пакеты, принадлежащие одному и тому же соединению, даже имеющие одни и те же адреса отправления и назначения, могут перемещаться по разным независимым друг от друга маршрутам.

Передача с установлением виртуального канала. Если в число параметров соединения входит маршрут, то все пакеты, передаваемые в рамках данного соединения, должны проходить по указанному пути. Такой единственный заранее проложенный фиксированный маршрут, соединяющий конечные узлы в сети с коммутацией пакетов, называют виртуальным каналом.

В одной и той же сетевой технологии могут быть задействованы разные способы обмена данными. Так, диаграммный протокол IP используется для передачи данных между отдельными сетями, составляющими Интернет. В то же время обеспечением надежной доставки данных между конечными узлами этой сети занимается протокол TCP, устанавливающий логические соединения без фиксации маршрута. И наконец, Интернет является примером сети, использующей технику виртуальных каналов, т.к. в состав Интернета входит немало сетей ATM и Frame Relay, поддерживающих вирт. каналы.

# Характеристики NAT. Терминология NAT. Статический NAT. Примеры использования.

NAT (от англ. Network Address Translation — «преобразование сетевых адресов») — это механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов. Также имеет названия IP Masquerading, Network Masquerading и Native Address Translation.

В терминологии NAT под «внутренней сетью» подразумевается набор сетей, задействованных в преобразовании. Термин «внешняя сеть» относится ко всем остальным сетям.

При использовании NAT, IPv4-адреса представляют разные точки назначения в зависимости от того, находятся ли они в частной или в публичной сети (Интернет), а также от того, является ли трафик входящим или исходящим.

В NAT предусмотрено 4 типа адресов:

• внутренний локальный адрес;

• внутренний глобальный адрес;

• внешний локальный адрес;

• внешний глобальный адрес.

При определении используемого типа адреса важно помнить, что терминология NAT всегда применяется с точки зрения устройства с преобразуемым адресом:

• Внутренний адрес — это адрес устройства, преобразуемый механизмом NAT.

• Внешний адрес — это адрес устройства назначения.

В рамках NAT по отношению к адресам также используется понятие локальности или глобальности:

• Локальный адрес — это любой адрес, появляющийся во внутренней части сети.

• Глобальный адрес — это любой адрес, появляющийся во внешней части сети.

Статическое преобразование сетевых адресов (NAT) выполняет взаимно однозначное преобразование внутренних IP-адресов во внешние. Это позволяет преобразовать IP-адрес внутренней сети во внешний IP-адрес.

Статический NAT позволяет устанавливать соединения как внутренним, так и внешним системам, например, хостам Internet. Этот тип преобразования особенно рекомендуется применять для организации общего доступа к системе, находящейся во внутренней сети. Для этого нужно создать правило NAT для преобразования фактического адреса системы во внешний адрес. Этот адрес будет доступен внешним пользователям. В этом случае никто не сможет получить информацию о внутренней сети для последующих атак извне.

Ниже перечислены особенности статического NAT:

• Это взаимно однозначное преобразование.

• Его можно инициировать как из внешней, так и из внутренней сети.

• Целевой адрес для преобразования может быть любым адресом.

• Целевой адрес для преобразования не может применяться в качестве интерфейса IP

• Нельзя применять NAT для преобразования портов.

# Динамический NAT. Преобразование адресов портов. Преимущества и недостатки NAT.

Динамический NAT использует пул публичных адресов и назначает их по принципу «первым пришел, первым обслужен». Когда внутреннее устройство запрашивает доступ к внешней сети, динамический NAT назначает доступный общедоступный IPv4-адрес из пула. Подобно статическому NAT, динамический NAT требует наличия достаточного количества общедоступных адресов для удовлетворения общего количества одновременных сеансов пользователя.

преобразование адресов портов (PAT), также называемое NAT с перегрузкой, сопоставляет множество частных IPv4-адресов одному или нескольким публичным IPv4-адресам. Именно этот метод реализуется большинством домашних маршрутизаторов. Интернет-провайдер назначает маршрутизатору один адрес, но несколько членов семьи могут одновременно получать доступ в Интернет. NAT с нагрузкой — это наиболее распространенный метод преобразования сетевых адресов.

С помощью данного метода множество адресов могут быть сопоставлены одному или нескольким адресам, поскольку каждый частный адрес также отслеживаются по номеру порта. Если устройство начинает сеанс TCP/IP, оно создаёт значение порта TCP или UDP для источника, чтобы уникальным образом определить сеанс. Когда маршрутизатор NAT получает пакет от клиента, он использует свой номер порта источника, чтобы уникальным образом определить конкретное преобразование NAT.

PAT гарантирует, что устройства будут использовать разные номера портов TCP для каждого сеанса взаимодействия с сервером в Интернете. При возвращении ответа от сервера номер порта источника, который становится номером порта назначения при обратной передаче, определяет, какому устройству маршрутизатор перешлет соответствующие пакеты. Процесс PAT также убеждается в том, что входящие пакеты действительно были запрошены, повышая таким образом степень безопасности сеанса.

Для управления анимацией используйте кнопки «Воспроизведение» и «Пауза» на рисунке.

Анимация иллюстрирует процесс преобразования адресов портов (PAT). Для того, чтобы различать преобразования, механизм PAT добавляет уникальные номера портов источника к внутреннему глобальному адресу.

преимущества

1. NAT помогает в сохранении адресного пространства IPv4, когда пользователь использует перегрузку NAT

2. NAT повышает надежность и гибкость соединений с глобальной сетью за счет развертывания нескольких исходных пулов, пула балансировки нагрузки и резервных пулов.

3. NAT имеет выдающийся метод сетевой адресации. Если используется глобальный IP-адрес, адресное пространство должно быть правильно назначено. Потому что при развитии сети может потребоваться много IP-адресов

4. NAT обеспечивает дополнительный уровень безопасности в сети, поскольку хост, встроенный в сеть NAT, недоступен другим сетевым устройствам в соответствии с предпочтениями пользователя.

Недостатки

1. Когда гость запрашивает удаленный доступ, он дважды проверит, относится ли соединение с роутера к NAT. Но некоторые гости установили соединение с другого хоста, если конкретный пользователь не отвечает на правильный хост, то он получит запрос, другой хост. Этот критерий приведет к снижению производительности сети

2. Если существует множество приложений и протоколов, основанных на сквозных функциях, то сеть пользователя не может быть доступна другим пользователям. Поскольку хост встроен в сеть NAT, которая недоступна, как обсуждалось выше

3. Если возникнет необходимость устранения неполадок в сети из удаленных районов, устранение неполадок будет сложным и приведет к потере сквозной прослеживаемости.

4. Применение протоколов туннелирования усложняет использование преобразованных значений NAT в заголовках IP, а также прерывает проверки целостности, выполняемые IPsec и протоколами левого туннелирования.

5. Службы, которым требуются установочные соединения UDP или TCP с глобальной стороны, могут подвергаться воздействию и иногда могут быть недоступны.

# Назначение транспортного уровня OSI. Понятия порта и сокета. Задачи и функции протокола UDP. Формат дейтаграммы.

**Транспортный уровень**

Основная статья: Транспортный уровень

Транспортный уровень (англ. transport layer) модели предназначен для обеспечения надёжной передачи данных от отправителя к получателю. При этом уровень надёжности может варьироваться в широких пределах. Существует множество классов протоколов транспортного уровня, начиная от протоколов, предоставляющих только основные транспортные функции (например, функции передачи данных без подтверждения приёма), и заканчивая протоколами, которые гарантируют доставку в пункт назначения нескольких пакетов данных в надлежащей последовательности, мультиплексируют несколько потоков данных, обеспечивают механизм управления потоками данных и гарантируют достоверность принятых данных. Например, UDP ограничивается контролем целостности данных в рамках одной датаграммы и не исключает возможности потери пакета целиком или дублирования пакетов, нарушение порядка получения пакетов данных; TCP обеспечивает надёжную непрерывную передачу данных, исключающую потерю данных или нарушение порядка их поступления или дублирования, может перераспределять данные, разбивая большие порции данных на фрагменты и наоборот, склеивая фрагменты в один пакет.

Порт является частью адреса в протоколах TCP и UDP. Он используется для того, чтобы помочь ОС определить, какое приложение должно получить полученные данные. ОС должна поддерживать порты для поддержки TCP и UDP, потому что порты являются неотъемлемой частью TCP и UDP.

Сокет является частью интерфейса, который ОС предоставляет приложениям, чтобы позволить им отправлять и получать сетевые данные. Большинство реализаций сокетов поддерживают множество протоколов помимо TCP и UDP, некоторые из которых не имеют понятия портов. ОС не должна поддерживать сокеты для поддержки TCP или UDP; это может обеспечить другой интерфейс для приложений для использования. Сокет - это просто один из способов отправки и получения данных через определенный порт.

Протокол UDP (User Datagram Protocol, RFC-768) является одним из основных протоколов, расположенных непосредственно над IP. Он предоставляет прикладным процессам транспортные услуги, немногим отличающиеся от услуг протокола IP. Протокол UDP обеспечивает доставку дейтограмм, но не требует подтверждения их получения. Протокол UDP не требует соединения с удаленным модулем UDP ("бессвязный" протокол). К заголовку IP-пакета UDP добавляет поля порт отправителя и порт получателя, которые обеспечивают мультиплексирование информации между различными прикладными процессами, а также поля длина UDP-дейтограммы и контрольная сумма, позволяющие поддерживать целостность данных. Таким образом, если на уровне IP для определения места доставки пакета используется адрес, на уровне UDP - номер порта.

В RFC-768 говорится, что поле "Порт отправителя" является опционным, что вроде бы, позволяет его не заполнять. Действительно, если UDP-дейтограммы используются для передачи цифровой ТВ-программы через Интернет, номер порта отправителя получателю знать не обязательно. Но за 40 лет, прошедших с написания RFC-769 перечень приложений, использующих протокол UDP существенно расширился. Например, для многопользовательских видеоконференций стало важно, в какое из открытых окон следует адресовать содержимое UDP-дейтограммы, а это может зависеть от номера порта отправителя.

Область использования UDP

Примерами сетевых приложений, использующих UDP, являются NFS (Network File System), TFTP (Trivial File Transfer protocol, RFC-1350), RPC (Remote Procedure Call, RFC-1057) и SNMP (Simple Network Management Protocol, RFC-1157). Малые накладные расходы, связанные с форматом UDP, а также отсутствие необходимости подтверждения получения пакета, делают этот протокол наиболее популярным при реализации приложений мультимедиа, но главное его место работы - локальные сети и мультимедиа.

Хотя протокол UDP не гарантирует доставки, по умолчанию предполагается, что вероятность потери пакета достаточно мала.

Прикладные процессы и модули UDP взаимодействуют через UDP-порты. Эти порты нумеруются, начиная с нуля. Прикладной процесс, предоставляющий некоторые услуги (сервер), ожидает сообщений, направленных в порт, специально выделенный для этих услуг. Программа-сервер ждет, когда какая-нибудь программа-клиент запросит услугу.

Длина дейтаграммы. Это полная длина IP-дейтаграммы (заголовок плюс данные) в байтах. Поскольку размер этого поля равен 16 бит, теоретически максимальный размер IP-дейтаграммы может составлять 65 535 байт. Однако размер дейтаграмм редко превосходит 1500 байт и обычно ограничивается значением 576 байт.

□ Идентификатор, флаги, смещение фрагмента. Эти три поля имеют отношение к так называемой IP-фрагментации. Этот вопрос мы подробно рассмотрим чуть позже. Интересно отметить, что новая версия протокола IP (IPv6) запрещает фрагментацию в маршрутизаторах.

Время жизни. Поле времени жизни (Time То Live, TTL) позволяет гарантировать, что дейтаграммы не будут вечно циркулировать в сети (например, из-за существующей в течение долгого времени маршрутной петли). Значение этого поля уменьшается на единицу на каждом маршрутизаторе. Когда значение поля TTL достигает нуля, маршрутизатор отбрасывает дейтаграмму.

# Задачи и функции протокола TCP. Назначение основных полей заголовка сегмента.

На транспортном уровне стека TCP/IP используются два основных протокола: TCP и UDP. Общее представление о функциях транспортного уровня можно получит в соответствующей статьей. В данном тексте речь пойдёт о протоколе TCP (Transmission Control Protocol), который используется для обеспечения надёжной доставки данных на транспортном уровне.

Существуют общие задачи транспортного уровня, с которыми справляется как TCP, так и UDP. Основных задач, собственно, две: сегментация данных, приходящих с уровня приложений и адресация приложений (передающего и принимающего) при помощи портов. Подробнее об этом можно прочесть в статье, посвященной транспортному уровню.

Помимо этого, TCP обеспечивает:

• Надёжную доставку сегментов.

• Упорядочивание сегментов при получении.

• Работу с сессиями.

• Контроль за скоростью передачи.

Рассмотрим эти возможности более детально

Надёжная доставка сегментов

Под надёжной доставкой подразумевается автоматическая повторная пересылка недошедших сегментов. Каждый сегмент маркируется при помощи специального поля — порядкового номера (sequence number). После отправки некоторого количества сегментов, TCP на отправляющем узле ожидает подтверждения от получающего, в котором указывается порядковый номер следующего сегмента, который адресат желает получить. В случае, если такое подтверждение не получено, отправка автоматически повторяется. После некоторого количества неудачных попыток, TCP считает, что адресат не доступен, и сессия разрывается.

Упорядочивание сегментов при получении

Как несложно догадаться, каждый сегмент на нижний уровнях TCP/IP обрабатывается индивидуально. То есть, как минимум, он будет запакован в индивидуальный пакет. Пакеты идут по сети и промежуточные маршрутизаторы в общем случае уже ничего не знают о том, что запаковано в эти пакеты. Часто пакеты с целью балансировки нагрузки могут идти по сети разными путями, через разные промежуточные устройства, с разной скоростью. Таким образом получатель, декапсулировав их, может получить сегменты не в том порядке, в котором они отправлялись.

Работа с сессиями

Перед началом передачи полезных данных, TCP позволяет убедиться в том, что получатель существует, слушает нужный отправителю порт и готов принимать данные для этого устанавливается сессия при помощи механизма трёхстороннего рукопожатия (three-way handshake), о котором можно прочесть в соответствующей статье. Далее, в рамках сессии передаются полезные пользовательские данные. После завершения передачи сессия закрывается, тем самым получатель извещается о том, что данных больше не будет, а отправитель извещается о том, что получатель извещён.

Контроль за скоростью передачи

Контроль за скоростью передачи позволяет корректировать скорость отправки данных в зависимости от возможностей получателя. Например, если быстрый сервер отправляет данные медленному телефону, то сервер будет передавать данные с допустимой для телефона скоростью.

Заголовок TCP сегмента имеет следующую структуру:

• Source port и Destination port — это соответственно номера портов получателя и отправителя, идентифицирующие приложений на отправляющем и принимающем узлах.

• Sequence number и Acknowledgment number — это порядковый номер сегмента и номер подтверждения, которые используются для надёжной доставки. Например, если отправитель шлёт сегмент с SN 100, то получатель может ответить на него ACK 101 SN200, что означает: «Я получил твой сегмент с номером 100 и жду от тебя 101-го, кстати, у меня своя нумерация. Мои номера начинаются с 200» Отправитель, в свою очередь, может ответить SN101 ACK201, что означает «Я получил от тебя сегмент с номером 200, могу принять следующий 201-ый, а вот тебе мой 101-ый сегмент, которого ты ждёшь». Ну и так далее.

• Header length — Это четырёхбитное поле, содержащее в себе длину заголовка TCP сегмента.

• Reserved — 6 зарезервированных на всякий случай бит.

• Control — поле с флагами, которые используются в процессе обмена информацией и описывают дополнительное назначение сегмента. Например, флаг FIN используется для завершения соединений, SYN и ACK — для установки.

• Window — содержит размер окна, о чём было сказано выше.

• Checksumm — контрольная сумма заголовка и данных.

• Urgent — признак важности (срочности) данного сегмента.

• Options — дополнительное необязательное поле, которое может использоваться, например, для тестирования протокола.

• В разделе данных содержатся собственно данные, полученные от протокола уровня приложений, либо их кусок, если данные пришлось разбивать.

# Последовательность передачи информации по протоколу TCP (процесс 3-х этапного «рукопожатия»).

Процесс начала сеанса TCP, также называемый «рукопожатие» (англ. handshake), состоящий из трёх шагов:

1. Клиент, который намеревается установить соединение, посылает серверу сегмент с номером последовательности и флагом SYN.

Дальнейший алгоритм:

• Сервер получает сегмент, запоминает номер последовательности и пытается создать сокет (буферы и управляющие структуры памяти) для обслуживания нового клиента;

• В случае успеха сервер посылает клиенту сегмент с номером последовательности и флагами SYN и ACK, и переходит в состояние SYN-RECEIVED;

• В случае неудачи сервер посылает клиенту сегмент с флагом RST.

2. Если клиент получает сегмент с флагом SYN, то он запоминает номер последовательности и посылает сегмент с флагом ACK.

Дальнейший алгоритм:

• Если он одновременно получает и флаг ACK (что обычно и происходит), то он переходит в состояние ESTABLISHED;

• Если клиент получает сегмент с флагом RST, то он прекращает попытки соединиться;

• Если клиент не получает ответа в течение 10 секунд, то он повторяет процесс соединения заново.

3. Если сервер в состоянии SYN-RECEIVED получает сегмент с флагом ACK, то он переходит в состояние ESTABLISHED.

• В противном случае после тайм-аута он закрывает сокет и переходит в состояние CLOSED.

Процесс называется «трёхэтапным рукопожатием» (англ. three way handshake), так как несмотря на то, что возможен процесс установления соединения с использованием четырёх сегментов (SYN в сторону сервера, ACK в сторону клиента, SYN в сторону клиента, ACK в сторону сервера), на практике для экономии времени используется три сегмента.

# Протокол ТСР. Механизмы обеспечения надежности TCP. Управление потоком TCP.

Transmission Control Protocol (TCP, протокол управления передачей) — один из основных протоколов передачи данных интернета, предназначенный для управления передачей данных. Пакеты в TCP называются сегментами.

В стеке протоколов TCP/IP выполняет функции транспортного уровня модели OSI.

Механизм TCP предоставляет поток данных с предварительной установкой соединения, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета, гарантируя тем самым, в отличие от UDP, целостность передаваемых данных и уведомление отправителя о результатах передачи.

Реализации TCP обычно встроены в ядра ОС. Существуют реализации TCP, работающие в пространстве пользователя.

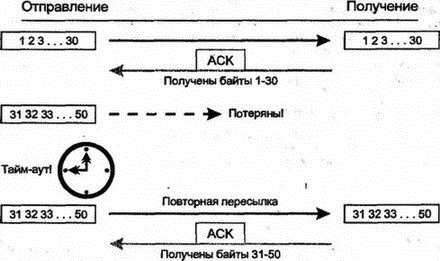
Когда осуществляется передача от компьютера к компьютеру через Интернет, TCP работает на верхнем уровне между двумя конечными системами, например, браузером и веб-сервером. TCP осуществляет надёжную передачу потока байтов от одного процесса к другому. TCP реализует управление потоком, управление перегрузкой, рукопожатие, надёжную передачу

Для обеспечения надежной пересылки данных в TCP используются нумерация (numbering) и подтверждение (acknowledgment — ACK). Схема нумерации TCP несколько необычна: *каждый* пересылаемый по соединению *октет* рассматривается как имеющий порядковый номер. Заголовок сегмента TCP содержит порядковый номер *первого октета данных этого сегмента*.

От приемника требуется подтверждение получения данных. Если ACK не приходит за интервал тайм-аута, данные передаются повторно. Этот способ называется *позитивным подтверждением с ретрансляцией* (positive acknowledgment with retransmission).

Получатель данных TCP проводит строгий контроль входящих порядковых номеров, чтобы проверить последовательность получения данных и отсутствие потерянных частей. Поскольку ACK случайным образом может быть потерян или задержан, к получателю могут поступить дублированные сегменты. Порядковые номера позволяют определить дублирование данных, которые далее отбрасываются.

На рис. 10.6 показан упрощенный взгляд на тайм-аут и повторную пересылку в TCP.



**Рис. 10.6.** Тайм-аут и повторная пересылка в TCP

# Принцип организации DNS. Рекурсивные и итеративные запросы.

Работа DNS достаточно проста, но из-за незнания её основ возникает основная масса проблем и вопросов при переносе существующего доменного имени и регистрации нового.

Остановимся немного подробней на описании самой схемы.

 Когда пользователь запускает веб-браузер и вводит название домена сайта, его ПК отправляет запрос к DNS-серверу интернет-провайдера для получения IP-адреса, на котором находится домен (1).

 Если DNS-серверы провайдера не обнаруживают в своем кэше информации о запрашиваемом сайте, то отправляют запрос на корневые DNS-серверы (2).

 Корневой DNS-сервер ищет в своей базе данных информацию о серверах имен хостинг-провайдера, на которых присутствует этот сайт. Далее, он сообщает их кэширующему DNS-серверу провайдера (3).

 После того, как кэширующий DNS-сервер интернет-провайдера получает информацию о серверах имен хостинг-провайдера он опрашивает любой из них (4) и, в случае получения положительного результата получения IP-адреса (5), помещает в кэш. Кэширование используется для того, чтобы снизить как нагрузку на интернет-каналы, так и для ускорения получения результата запроса.

 После этого DNS-сервер провайдера передает IP-адрес браузеру пользователя, совершившему запрос сайта (6).

 И уже после этого браузер, получив IP-адрес запрашиваемого сайта, переходит на сам сайт (7 и 8).

Важно, что обновление информации о сервера имен провайдера происходит не мгновенно, а через некоторое определенное (для каждого DNS-сервера, в зависимости от настроек и провайдера данные значения могут варьироваться) время.

Так же если кто-либо из пользователей вашего интернет-провайдера заходил на сайт и после этого сайт изменил IP-адрес или сервера имен, то в базе данных кеширующего DNS-сервера провайдера останется старая информация до тех пор, пока кеш не обновится. И до этого момента при запросе сайта будет предоставляться устаревшая информация о месторасположении (IP-адрес). Хотя пользователям с других провайдеров сайт может открываться уже с нового IP-адреса.

В основном если у вас возникла вышеописанная ситуация, то волноваться не стоит. Следует подождать некоторое время для обновления информации на корневых DNS-серверах и DNS-серверах провайдера.

А пока вы ожидаете, на всякий случай можете самостоятельно продиагностировать корректность настроек своего домена, либо обратиться в техническую поддержку.

В DNS имеются следующие типы запросов: итеративный (он же прямой), обратный и рекурсивный.

Итеративный (он же прямой, он же нерекурсивный) запрос посылает доменное имя DNS серверу и просит вернуть либо IP адрес этого домена, либо имя DNS сервера, авторитативного для этого домена. При этом, сервер DNS не опрашивает другие серверы для получения ответа. Так работают корневые и TLD серверы.

Рекурсивный запрос посылает DNS серверу доменное имя и просит возвратить IP адрес запрошенного домена. При этом сервер может обращаться к другим DNS серверам.

Обратный запрос посылает IP и просит вернуть доменное имя.

Любой DNS-server должен отвечать на итеративные запросы. Возможно настроить DNS отвечать и на рекурсивные запросы. Если DNS не настроен отвечать на рекурсивные запросы, он обрабатывает их как итеративные.

Обычно, в локальной сети стоит DNS-сервер, обрабатывающий рекурсивные запросы, а также, скорее всего, он настроен на кэширование запросов, что экономит трафик и снижает нагрузку на сеть. Схему взаимодействия клиента и DNS серверов можно представить следующей картинкой:

# Прикладной уровень модели OSI. Электронная почта. Принципы организации и функционирования протокола SMTP и POP3, IMAP

Прикладной уровень (уровень приложений; англ. application layer) — верхний уровень модели, обеспечивающий взаимодействие пользовательских приложений с сетью:

• позволяет приложениям использовать сетевые службы:

o удалённый доступ к файлам и базам данных,

o пересылка электронной почты;

• отвечает за передачу служебной информации;

• предоставляет приложениям информацию об ошибках;

• формирует запросы к уровню представления.

• Работа электронной почты основывается на использовании электронных почтовых ящиков. В момент отправления электронное письмо маршрутизируется с одного сервера на другой сервер (такие серверы также называют почтовыми узлами, или релеями), и в результате доходит до почтового сервера получателя. Точнее говоря, сообщение отправляется на почтовый сервер, в задачу которого входит транспортировка сообщений (называющийся MTA, то есть сервер электронной почты) до MTA получателя. В Интернете MTA связываются друг с другом при помощи протокола SMTP, поэтому, логично, что они называются SMTP-серверами (или иногда серверами исходящей почты).

• Затем MTA получателя доставляет электронное письмо на сервер входящей почты (называющийся MDA, то есть агент доставки электронной почты), который хранит письмо в ожидании его приема пользователем. Есть два основных протокола извлечения почты из MDA:

POP3 (Протокол POP), более старый из двух, использующийся, чтобы извлечь письмо и, в определенных случаях, оставить его копию на сервере;

IMAP (Протокол IMAP), использующийся для координирования статуса сообщений (прочитано, удалено, перемещено) между многочисленными почтовыми клиентами. При использовании IMAP копия каждого письма сохраняется на сервере, чтобы эта задача по синхронизации могла быть выполнена.

• Серверы входящей почты получили названия POP-серверы и IMAP-серверы, в зависимости от используемого протокола.

• Используя аналогию с обычной почтой, можно сказать, что MTA выступают в качестве «почтовых отделений» (сортировка почты и доставка почтальоном), в то время как MDA выполняют роль почтовых ящиков, которые хранят сообщения (насколько позволяет их объем), пока получатель не проверит свой ящик.

• Чтобы предотвратить проверку почты посторонними лицами, MDA защищен именем пользователя (логин) и паролем.

• Извлечение почты осуществляется при помощи программы, которая называется MUA (клиент электронной почты).

• Если MUA установлен на компьютере пользователя,– это клиент электронной почты (такой как Mozilla Thunderbird, Microsoft Outlook, Eudora Mail, Incredimail или Lotus Notes).

• Если же в качестве MUA выступает веб-интерфейс, использующийся для взаимодействия с сервером входящей почты, он называется веб-почтой.

• Открытый релей

• По умолчанию или в силу исторических причин нет необходимости аутентифицировать себя, чтобы отправить письмо, а это значит, что подделать свой собственный адрес при отправке сообщения очень легко.

• По этой причине почти все Интернет-провайдеры ставят шлюзы на свои SMTP-серверы, чтобы только их пользователи могли их использовать, или, точнее, только машины, чей IP-адрес принадлежит домену Интернет-провайдера. По этой причине пользователи должны изменять настройки исходящего сервера в своих почтовых клиентах каждый раз, когда они переезжают жить или работать в другое место.

• Если почтовый сервер (релей) не имеет подобной защиты, третьи лица могут отправлять письма. Такой сервер называется открытым релеем.

• Открытые релеи, как правило, используются спамерами, поскольку они позволяют замаскировать истинное происхождение сообщений. Чтобы защитить своих пользователей от получения писем с таких серверов, многие Интернет-провайдеры ведут постоянно обновляемые черные списки открытых релеев.

ростой протокол передачи почты (SMTP), используется для связи с удаленным сервером и последующей отправке сообщений с локального клиента на удаленный сервер, и в конечном итоге на сервер получателя сообщений. На вашем сервере электронной почты, этот процесс контролируется специальной службой (MTA). Стоит упомянуть, что SMTP используется исключительно для отправки сообщений.

Порты SMTP:

• Порт 25 — порт без шифрования

• Порт 465 — порт SSL/TLS, также известный как SMTPS

POP3 (протокол почтового отделения версия 3) часто используется для связи с удаленным сервером электронной почты и загрузки сообщений на локальный почтовый клиент с последующим удалением его на сервере, к примеру Outlook, Thunderbird, Windows Mail, Mac Mail и т.д. Однако обычно почтовые клиенты предлагают выбор — оставлять или нет копии сообщений на сервере. Если вы используете несколько устройств для отправки сообщений, то рекомендуется оставлять эту функцию включенной, в противном случае, на другом устройстве у вас не будет доступа к отправленным сообщениям, которые не были сохранены на удаленном сервере. Также стоит отметить, что POP3 — протокол работающий только в одном направлении, это означает, что данные берутся с удаленного сервера и отправляются на локальный клиент.

Порты POP3, по умолчанию являются такими:

Порт 110 — порт без шифрования

Порт 995 — порт SSL/TLS, также известный как POP3S

Шаг 2 — Различия между POP3 и IMAP, и какие порты у IMAP?

IMAP (протокол прикладного уровня для доступа к электронной почте), также как и POP3 используется для получения сообщений электронной почты на локальный клиент, однако, он имеет существенное отличие — загружаются только лишь заголовки электронных сообщений, сам текст письма остается на сервере. Данный протокол связи работает в две стороны, если происходят изменения на локальном клиенте, они передаются и на сервер. В последнее время IMAP стал более популярным, так как такие гиганты-провайдеры услуг электронной почты, как Gmail, стали рекомендовать использовать его вместо POP3.

Порты IMAP, по умолчанию являются такими:

• Порт 143 — порт без шифрования

• Порт 993 — порт SSL/TLS, также известный как IMAPS

# Протокол HTTP. Принципы организации и функционирования. Основные команды и их формат.

HTTP — широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (то есть документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам).

Аббревиатура HTTP расшифровывается как HyperText Transfer Protocol, «протокол передачи гипертекста». В соответствии со спецификацией OSI, HTTP является протоколом прикладного (верхнего, 7-го) уровня. Актуальная на данный момент версия протокола, HTTP 1.1, описана в спецификации RFC 2616.

Протокол HTTP предполагает использование клиент-серверной структуры передачи данных. Клиентское приложение формирует запрос и отправляет его на сервер, после чего серверное программное обеспечение обрабатывает данный запрос, формирует ответ и передаёт его обратно клиенту. После этого клиентское приложение может продолжить отправлять другие запросы, которые будут обработаны аналогичным образом.

Задача, которая традиционно решается с помощью протокола HTTP — обмен данными между пользовательским приложением, осуществляющим доступ к веб-ресурсам (обычно это веб-браузер) и веб-сервером. На данный момент именно благодаря протоколу HTTP обеспечивается работа Всемирной паутины.

1. Что и для чего

HTTP (англ. HyperText Transfer Protocol) – это протокол передачи гипертекста семейства протоколов TCP/IP. Другими словами, это соглашение между клиентом и сервером по формату запроса и ответа при обмене информацией и данными. Под клиентом, как правило, выступает обычный пользователь, а точнее его браузер. Под сервером, как правило, выступает сайт, а точнее программа-сервер, работающая на компьютере-сервере и принимающая запросы от клиентов.

Клиент обращается к серверу по URI (Uniform Resource Identifier). Это адрес файла на сервере. Это может быть также логический или абстрактный ресурс.

HTTP – протокол самого верхнего прикладного уровня согласно сетевой модели OSI (open systems interconnection basic reference model). Протокол для работы приложений.

2. Программное обеспечение

ПО для работы с HTTP разделяется на три класса программ:

• Серверы. Принимают, обрабатывают запросы и отвечают на них. Основные приложения: Apache, Internet Information Services (IIS), nginx.

• Клиенты. Отсылают запросы и интерпретируют ответы для пользователей. Основное приложение – это браузер. Также это могут быть менеджеры закачек, интернет-боты, ГИС…

• Посредники (прокси). Они просто пропускают через себя трафик. Представители: Squid, UserGate, Multiproxy, Naviscope, nginx.

3. Структура протокола

Каждое HTTP-сообщение состоит из трех частей, которые передаются в указанном порядке:

• Стартовая строка – определяет тип сообщения;

• Заголовки – характеризуют тело сообщения, параметры передачи и прочие сведения;

• Тело сообщения – непосредственно данные сообщения. Обязательно должно отделяться от заголовков пустой строкой.

3.1. Стартовые строки

Пример стартовой строки (starting line) запроса:

GET /wiki/HTTP HTTP/1.0

Здесь указаны метод запроса, URI (адрес страницы, который указывается ниже в заголовке), протокол и его версия. Пример стартовой строки ответа:

HTTP/1.0 200 OK

Здесь указывается протокол, его версия и код состояния HTTP. Кодов много. Они делятся на 5 категорий: 1XX – для информации; 2XX – для указания успешного соединения; 3XX – для перенаправления; 4XX – для указания ошибки клиента; 5XX – для указания ошибки сервера.

3.2. Заголовки

Заголовки HTTP (headers) – это строки, содержащие разделенную двоеточием пару параметр-значение. В них может указываться самая разная информация, относящаяся к запрашиваемому ресурсу. Все заголовки разделяются на четыре основных группы:

• Основные заголовки – должны включаться в любое сообщение клиента и сервера.

• Заголовки запроса – используются только в запросах клиента.

• Заголовки ответа – только для ответов от сервера.

• Заголовки сущности – сопровождают каждую сущность сообщения.

3.3. Тело сообщения

Тело HTTP-сообщения (message-body), если оно присутствует, используется для передачи тела объекта, связанного с запросом или ответом. В теле могут передаваться сущности разных форматов (текст, графика, видео…) как вместе, так и по отдельности. Так как HTTP-сообщение – это текст, то бинарные файлы передаются в закодированном виде.

4. Сессии

HTTP устанавливает отдельную TCP-сессию на каждый запрос. В более поздних версиях HTTP было разрешено делать несколько запросов в ходе одной TCP-сессии, но браузеры обычно запрашивают только страницу и включенные в нее сущности, а затем сразу разрывают TCP-сессию.

Для сохранения данных предыдущей сессии при новом соединении клиента с сервером существуют механизмы, которые могут сохранять данные сессии или на стороне клиента, или на стороне сервера. В случае клиентского хранения браузер может их прочитать и передать на сервер в составе HTTP-запроса. Чтобы эти данные нельзя было подменить по пути к серверу, совместно с HTTP используется протокол SSL (Secure Sockets Layer) и это расширение HTTP уже называется HTTPS (HyperText Transfer Protocol Secure).

OPTIONS - Запрос информации об опциях соединения (например, методах, типах документов, кодировках), которые поддерживает сервер для запрашиваемого ресурса.

Если идентификатор запрашиваемого ресурса - звездочка ("\*"), то запрос предназначен для обращения к серверу в целом.

GET - Позволяет получить информацию, связанную с запрашиваемым ресурсом.

HEAD - Идентична команде GET, за исключением того, что сервер не возвращает в ответе тело сообщения.

POST - Используется для запроса, при котором адресуемый сервер принимает данные, включенные в тело сообщения (объект) запроса, и отправляет их на обработку приложению, указанному как запрашиваемый ресурс.

PUT - Тело сообщения, которое передается в запросе, сохраняется на сервере, причем идентификатор запрашиваемого ресурса будет идентификатором сохраненного документа.

DELETE - Запрос на удаление ресурса, имеющего запрашиваемый идентификатор.

TRACE - Используется для тестирования или диагностики. Получатель запроса (сервер Web) отправляет полученное сообщение обратно клиенту как тело сообщения ответа.

После получения и интерпретации сообщения запроса, сервер отвечает сообщением HTTP ответа.

# Протокол передачи файлов FTP. Пассивный и активный режимы.

File Transfer Protocol является одним из первых способов обмена данными между компьютерами по сети TCP / IP. Очень удобен для загрузки и скачивания файлов большого объема. Протокол работает в модели клиент-сервер, где FTP-сервер и FTP-клиент выполняют операции передачи данных.

FTP-сервер настроен в сети и является местом хранения файлов. Пользователи получают к нему доступ при помощи специального программного обеспечения (FTP-клиента), чтобы начать процесс копирования или загрузки данных на сервер.

FTP-клиент

Несколько десятилетий назад FTP-клиенты были всего лишь приложениями интерфейса командной строки (CLI). Теперь они имеют свои простые в использовании, интуитивно понятные интерфейсы для облегчения и упрощения передачи файлов. Используются для настольных компьютеров, серверов и мобильных устройств. Доступны в качестве автономных приложений, web-клиентов и простых расширений web-браузеров. Например, самыми лучшими и бесплатными FTP-клиентами для ОС Windows являются FileZilla, FireFTP, Cyberduck, WinSCP, FreeFTP.

FTP-сервер

FTP-сервер способен поддерживать как активные, так и пассивные соединения с FTP-клиентом. При активном способе подключения клиент открывает порт и прослушивает его, пока сервер активно подключается к нему. При пассивном соединении сервер открывает порт и слушает пассивно, что позволяет клиентам самим подключаться к нему.

Пассивное соединение является более безопасным, а также более предпочтительным для IT-администраторов, поскольку подключение к данным осуществляется с клиента на сервер, а не наоборот. Это более надежный метод, который позволяет избежать входящих подключений из интернета к отдельным клиентам. В межсетевых развертываниях все подключения выполняются из интернета на сервер, а не с сервера обратно в интернет. Поэтому пассивный режим также известен как «брандмауэр-дружественный».

Вопросы безопасности

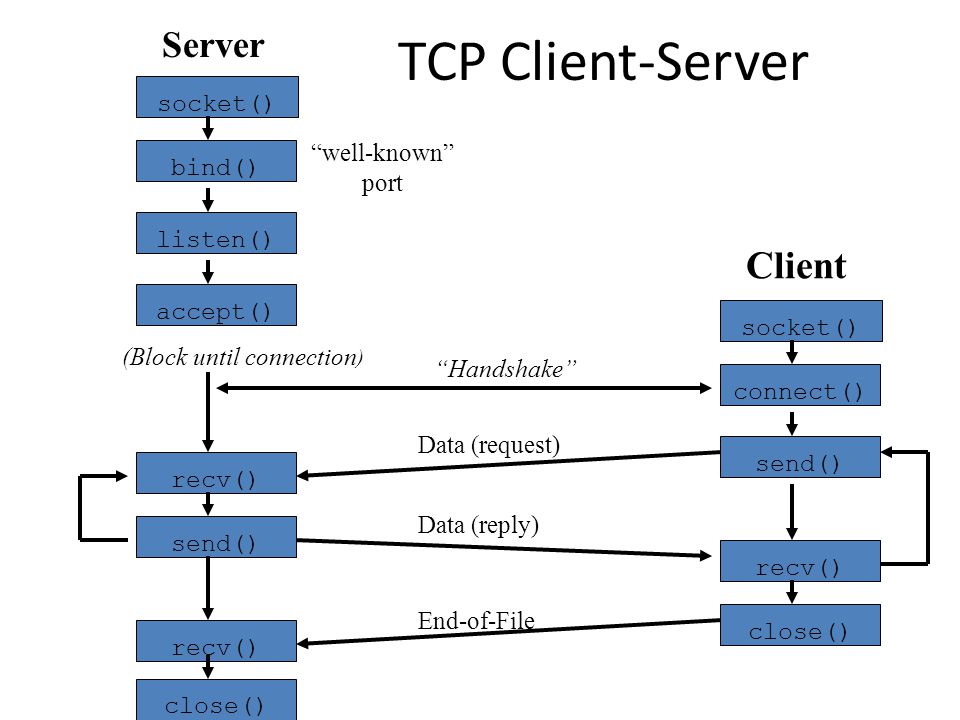
Простота использования протокола FTP является основной причиной его уязвимости. Возможность настройки доступа к серверам без достоверной аутентификации, хранение файлов в незашифрованном виде могут стать причиной перехвата передаваемых через интернет данных.

Еще одной причиной возникновения угрозы безопасности является существование множества анонимных FTP-серверов, предлагающих для бесплатного скачивания фильмы, музыку, графику. Установление сеанса связи с такими сайтами является максимально простым. Войти в систему можно, указав „ anonymous” в строке имени пользователя и адрес электронной почты в качестве пароля.

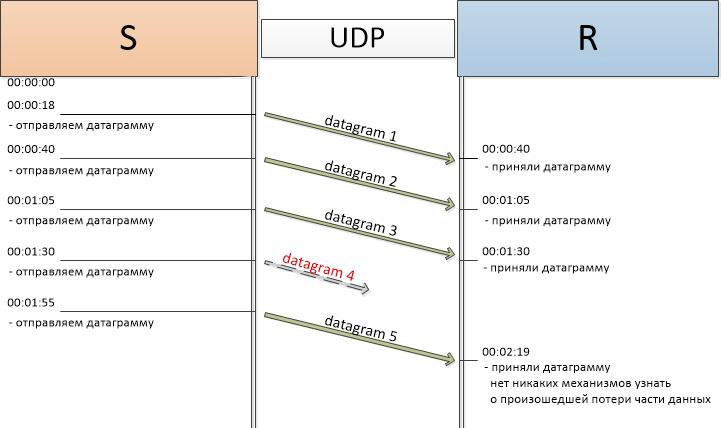
Для повышения безопасности передачи файлов при использовании FTP-соединения разработаны дополнительные протоколы шифрования: FTPS, SFTP, HTTPS с поддержкой криптографических протоколов Secure Sockets Layer (SSL), Secure Shell (SSH).

Главное отличие активного и пассивного режимов работы протокола FTP состоит в том, кто из связки клиент-сервер производит подключение для передачи данных, то есть, грубо говоря, кто к кому подключается. Также отличаются порты, на которые производится передача данных. При активном режиме работы, клиент производит управляющее соединение с сервером, а вот подключение для передачи данных производит уже сам сервер. При пассивном режиме работы подключение для передачи данных, равно как и управляющее соединение с сервером инициируется только клиентом. То есть, в активом режиме сервер подключается к клиенту для передачи данных, а в пассивном – клиент к серверу.

# Схема взаимодействия между клиентом и сервером по протоколу TCP (последовательность API - функций).



# Схема взаимодействия между клиентом и сервером по протоколу UDP (последовательность API - функций).



UDP не предпринимает никаких шагов по обнаружению потерь. Контроль ошибок передачи в UDP протоколе полностью возлагается на приложение.