Базовая эталонная модель взаимодействия открытых систем (модель OSI).

Эталонная модель OSI определяет семь уровней взаимодействия систем в сетях с коммутацией пакетов, дает им стандартные имена и указывает, какие функции должен выполнять каждый уровень.

- Прикладной уровень
- Уровень представления
- Сеансовый уровень
- Транспортный уровень
- Сетевой уровень
- Канальный уровень
- Физический уровень

□ Физический уровень (Physical layer) является самым нижним уровнем. Физический уровень (Physical layer) имеет дело с передачей битов по физическим каналам связи, таким, например, как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. На этом уровне определяются характеристики электрических сигналов, передающих дискретную информацию, например крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала. Кроме того, здесь стандартизуются типы разъемов и назначение каждого контакта.

□ На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (frames). Канальный уровень отвечает за формирование кадров (frame), физическую адресацию, разделение передающей среды, контроль ошибок.

□ Сетевой уровень (Network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать в общем случае различные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей. Сетевой уровень манипулирует пакетами (packet). Внутри сети доставка данных обеспечивается соответствующим канальным уровнем, а вот доставкой данных между сетями занимается сетевой уровень, который и поддерживает возможность правильного выбора маршрута передачи сообщения даже в том случае, когда характер структуры связей между составляющими сетями отличается от принятого в протоколах канального уровня.

Сети соединяются между собой специальными устройствами, называемыми маршрутизаторами. Маршрутизатор — это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. Чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое

количество транзитных передач между сетями, или хопов (hop — прыжок), каждый раз выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет.

□ Транспортный уровень (Transport layer) обеспечивает приложениям или верхним уровням стека — прикладному и сеансовому — передачу данных с той степенью надежности, которая им требуется. Выбор класса сервиса транспортного уровня определяется, с одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких, чем транспортный, уровней, а с другой стороны, этот выбор зависит от того, насколько надежной является система транспортировки данных в сети, обеспечиваемая уровнями, расположенными ниже транспортного — сетевым, канальным и физическим. Так, например, если качество каналов передачи связи очень высокое и вероятность возникновения ошибок, не обнаруженных протоколами более низких уровней, невелика, то разумно воспользоваться одним из облегченных сервисов транспортного уровня, не обремененных многочисленными проверками, квитированием и другими приемами повышения надежности. Если же транспортные средства нижних уровней изначально очень ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок.

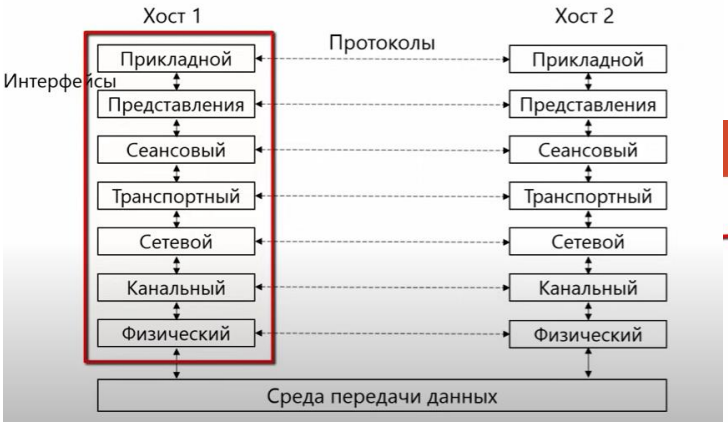
□ Сеансовый уровень (Session layer) обеспечивает управление взаимодействием: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала.

□ Представительный уровень (Presentation layer) имеет дело с формой представления передаваемой по сети информации, не меняя при этом ее содержания. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC. На этом уровне может выполняться шифрование и дешифрование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных служб.

□ Прикладной уровень (Application layer) — самый верхний уровень, являющийся, по сути, набором разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые web-страницы.

Единица данных, которой оперируют верхние три уровня, обычно называется сообщением (message).

Уровни модели OSI



Физический уровень

Передача битов по физическому каналу связи

Не вникает в смысл передаваемой информации

Задача: как представить биты информации в виде сигналов, передаваемых по среде

Канальный уровень

Передача сообщений по каналу связи

- Определение начала/конца сообщения в потоке бит

Обнаружение и коррекция ошибок

В широковещательной сети:

- Управление доступом к среде передачи данных
- Физическая адресация

Сетевой уровень

Объединяет сети, построенные на основе разных технологий

Задачи:

- Создание составной сети, согласование различий в сетях
- Адресация (сетевые или глобальные адреса)
- Определение маршрута пересылки пакетов в составной сети (маршрутизация)

Транспортный уровень

Обеспечивает передачу данных между **процессами** на хостах

Управление надежностью:

- Может предоставлять надежность выше, чем у сети
- Наиболее популярный сервис – защищенный от ошибок канал с гарантированным порядком следования сообщений

Сквозной уровень

- Сообщения доставляются от источника адресату
- Предыдущие уровни используют принцип **звеньев цепи**

Сеансовый уровень

Позволяет устанавливать сеансы связи

Задачи:

- Управление диалогом (очерёдность передачи сообщений)
- Управление маркерами (предотвращение одновременного выполнения критичной операции)
- Синхронизация (метки в сообщениях для возобновления передачи в случае сбоя)

Уровень представления

Обеспечивает согласование синтаксиса и семантики передаваемых данных

- Форматы представления символов
- Форматы чисел

Шифрование и дешифрование

Пример:

- Transport Layer Security (TLS) / Secure Sockets Layer (SSL)

Прикладной уровень

Набор приложений, полезных пользователям:

- Гипертекстовые Web-страницы
- Социальные сети
- Видео и аудио связь
- Электронная почта
- Доступ к разделяемым файлам
- и многое другое

Единицы передаваемых данных

Уровень	Название единицы
Прикладной	Сообщение
Представления	Сообщение
Сеансовый	Сообщение
Транспортный	Сегмент/Дейтаграмма
Сетевой	Пакет
Канальный	Кадр
Физический	Бит

Сетевое оборудование

Уровень модели OSI	Оборудование
Сетевой	Маршрутизатор
Канальный	Коммутатор, точка доступа
Физический	Концентратор

Модель OSI

Итоги

Модель взаимодействия открытых систем (Open Systems Interconnection, OSI)

- Эталонная модель организации компьютерных сетей
- Юридический стандарт организации ISO

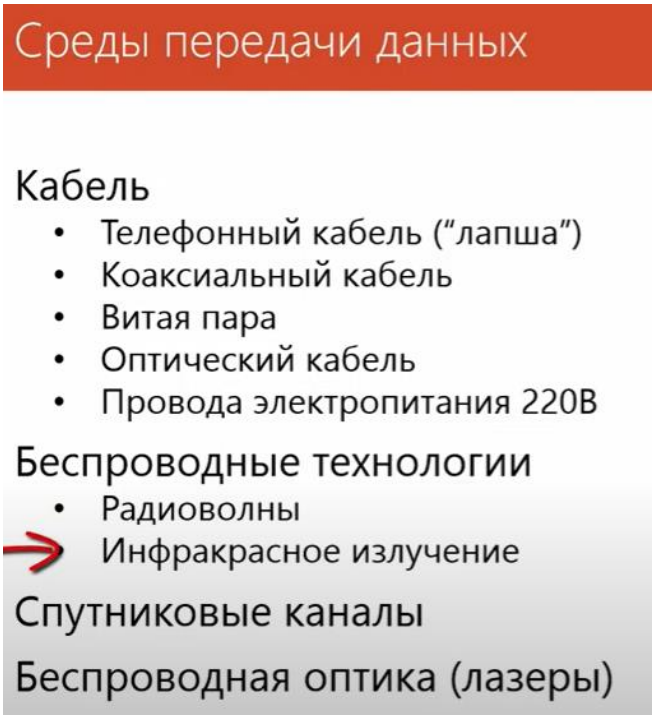
Включает 7 уровней организации сети и их назначение

- Протоколы не включены в модель

Не используются на практике

- «Общий язык» для описания компьютерных сетей

Виды каналов связи: оптический, медный, беспроводной. (с.163)



Линия связи состоит в общем случае из физической среды, по которой передаются информационные сигналы, аппаратуры передачи данных и промежуточной аппаратуры. Синонимом термина линия связи (line) является термин канал связи (channel).

Физическая среда передачи данных (medium) может представлять собой кабель, то есть набор проводов, изоляционных и защитных оболочек, соединительных разъемов, а также земную атмосферу или космическое пространство, через которые распространяются информационные сигналы. В современных телекоммуникационных системах информация передается с помощью электрического тока или напряжения, радиосигналов или световых сигналов — все эти физические процессы представляют собой колебания электромагнитного поля различной частоты и природы.

В зависимости от среды передачи данных линии связи разделяются на:

- проводные (воздушные);
- кабельные (медные и волоконно-оптические);
- радиоканалы наземной и спутниковой связи.

Проводные (воздушные) линии связи представляют собой провода без каких-либо изолирующих или экранирующих оплеток, проложенные между столбами и висящие в воздухе. По таким линиям связи традиционно передаются телефонные или телеграфные сигналы, но при отсутствии других возможностей эти линии используются и для передачи компьютерных данных. Скоростные качества и помехозащищенность этих линий оставляют желать много лучшего. Сегодня проводные линии связи быстро вытесняются кабельными.

Кабельные линии имеют достаточно сложную конструкцию. Кабель состоит из проводников, заключенных в несколько слоев изоляции: электрической,

электромагнитной, механической, а также, возможно, климатической. Кроме того, кабель может быть оснащен разъемами, позволяющими быстро выполнять присоединение к нему различного оборудования. В компьютерных (и телекоммуникационных) сетях применяются три основных типа кабеля: кабели на основе скрученных пар медных проводов, коаксиальные кабели с медной жилой, а также волоконно-оптические кабели (первые два типа кабелей называют также медными кабелями).

В зависимости от условий прокладки и эксплуатации кабели делятся на внутренние кабели (кабели зданий) и внешние кабели, которые, в свою очередь, подразделяются на подземные, подводные и кабели воздушной проводки.

Скрученная пара проводов называется витой парой (twisted pair). Скручивание проводов снижает влияние внешних и взаимных помех на полезные сигналы, передаваемые по кабелю. Для неответственных применений внутри здания иногда используются симметричные кабели из нескрученных пар — так называемая «лапша».

Волоконно-оптический кабель (optical fiber) состоит из тонких гибких стеклянных волокон (волоконных счетоводов), 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.

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 выходят за ее пределы, отражаясь от покрывающего слоя оболочки. В зависимости от распределения показателя преломления и от величины диаметра сердечника различают:

- многомодовое волокно со ступенчатым изменением показателя преломления;
- многомодовое волокно с плавным изменением показателя преломления;
- одномодовое волокно.

Понятие «мода» описывает режим распространения световых лучей во внутреннем сердечнике кабеля.

Волоконно-оптические кабели обладают отличными характеристиками всех типов: электромагнитными, механическими (хорошо гнутся, а в соответствующей изоляции обладают хорошей механической прочностью). Однако у них есть один серьезный недостаток — сложность соединения волокон с разъемами и между собой при необходимости наращивания (увеличения длины) кабеля.

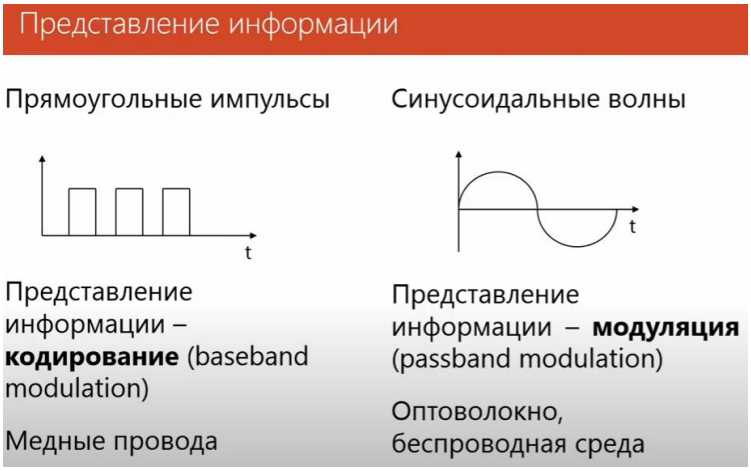
Сама стоимость волоконно-оптических кабелей ненамного превышает стоимость кабелей на витой паре, однако проведение монтажных работ с оптоволокном обходится намного дороже из-за трудоемкости операций и высокой стоимости применяемого монтажного оборудования.

Радиоканалы наземной и спутниковой связи образуются с помощью передатчика и приемника радиоволн. Существует большое разнообразие типов радиоканалов, отличающихся как используемым частотным диапазоном, так и дальностью канала. Диапазоны коротких, средних и длинных волн (КБ, СВ и ДВ), называемые также диапазонами амплитудной модуляции по типу используемого в них метода модуляции сигнала, обеспечивают дальнюю связь, но при невысокой скорости передачи данных. Более скоростными являются каналы, работающие на диапазонах ультракоротких волн, для которых характерна частотная модуляция, а также диапазонах сверхвысоких частот (СВЧ, или microwaves). В диапазоне СВЧ (свыше 4 ГГц) сигналы уже не отражаются ионосферой Земли, и для устойчивой связи требуется наличие прямой видимости между передатчиком и приемником. Поэтому такие частоты используют либо спутниковые каналы, либо радиорелейные каналы, где это условие выполняется.

В компьютерных сетях сегодня применяются практически все описанные типы

физических сред передачи данных, но наиболее перспективными являются волоконно-оптические кабели. На них сегодня строятся как магистрали крупных территориальных и городских сетей, так и высокоскоростные линии связи локальных сетей. Популярной средой является также витая пара, которая характеризуется отличным отношением качества к стоимости, а также простотой монтажа. Спутниковые каналы и радиосвязь используются чаще всего в тех случаях, когда кабельные линии связи применить нельзя — например, при прохождении канала через малонаселенную местность или же для связи с мобильными пользователями сети. Пока наиболее популярными являются мобильные телефонные сети, а мобильные компьютерные сети представлены сетями радио-Ethernet, имеющими несравнимо меньшее распространение. В мобильных сетях нового, так называемого третьего поколения (3d generation, 3G) предусматривается одновременная передача голоса и компьютерных данных, при этом каждый вид трафика считается одинаково важным.

Методы передачи данных. (с.188)



При передаче дискретных данных по каналам связи применяются два основных типа физического кодирования — на основе синусоидального несущего сигнала и на основе последовательности прямоугольных импульсов. Первый способ часто называется также модуляцией, или аналоговой модуляцией, подчеркивая тот факт, что кодирование осуществляется за счет изменения параметров аналогового сигнала. Второй способ обычно называют цифровым кодированием. Эти способы отличаются шириной спектра результирующего сигнала и сложностью аппаратуры, необходимой для их реализации.

При использовании прямоугольных импульсов спектр результирующего сигнала получается весьма широким. Это не удивительно, если вспомнить, что спектр идеального импульса имеет бесконечную ширину. Применение синусоиды приводит к спектру гораздо меньшей ширины при той же скорости передачи информации. Однако для синусоидальной модуляции требуется более сложная и дорогая аппаратура, чем для генерирования прямоугольных импульсов.

В настоящее время все чаще данные, изначально имеющие аналоговую форму — речь, телевизионное изображение, — передаются по каналам связи в дискретном виде, то есть в виде последовательности единиц и нулей. Процесс представления аналоговой информации в дискретной форме называется дискретной модуляцией. Термины «модуляция» и «кодирование» часто используют как синонимы.

При обмене данными между узлами сети используются три метода передачи данных:

- симплексная (однонаправленная) передача (телевидение, радио);
- полудуплексная (прием и передача информации осуществляются поочередно);
- дуплексная (двунаправленная), каждая станция одновременно передает и принимает данные.

Для передачи данных в сетях наиболее часто применяется последовательная передача. Широко используются следующие методы последовательной передачи: асинхронная и синхронная.

При асинхронной передаче каждый символ передается отдельной посылкой. Стартовые биты предупреждают приемник о начале передачи. Затем передается

символ. Для определения достоверности передачи используется бит четности (бит четности = 1, если количество единиц в символе нечетно, и 0 в противном случае. Последний бит «стопбит» сигнализирует об окончании передачи.

Преимущества: несложная отработанная система; недорогое (по сравнению с синхронным) интерфейсное оборудование.

Недостатки асинхронной передачи: третья часть пропускной способности теряется на передачу служебных битов (старт/стоповых и бита четности); невысокая скорость передачи по сравнению с синхронной; при множественной ошибке с помощью бита четности невозможно определить достоверность полученной информации.

Асинхронная передача используется в системах, где обмен данными происходит время от времени и не требуется высокая скорость передачи данных. Некоторые системы используют бит четности как символьный бит, а контроль информации выполняется на уровне протоколов обмена данными.

При использовании синхронного метода данные передаются блоками. Для синхронизации работы приемника и передатчика в начале блока передаются биты синхронизации. Затем передаются данные, код обнаружения ошибки и символ окончания передачи. При синхронной передаче данные могут передаваться и как символы, и как поток битов. В качестве кода обнаружения ошибки обычно используется циклический избыточный код обнаружения ошибок (CRC). Он вычисляется по содержимому поля данных и позволяет однозначно определить достоверность приятой информации.

Преимущества синхронного метода передачи информации: высокая эффективность передачи данных; высокие скорости передачи данных; надежный встроенный механизм обнаружения ошибок.

Цифровое кодирование. (с.195)

При цифровом кодировании дискретной информации применяют потенциальные и импульсные коды. В потенциальных кодах для представления логических единиц и пулей используется только значение потенциала сигнала, а его перепады, формирующие законченные импульсы, во внимание не принимаются. Импульсные коды позволяют представить двоичные данные либо импульсами определенной полярности, либо частью импульса — перепадом потенциала определенного направления.

Требования к методам цифрового кодирования

При использовании прямоугольных импульсов для передачи дискретной информации необходимо выбрать такой способ кодирования, который одновременно достигал бы несколько целей:

- имел при одной и той же скорости наименьшую ширину спектра результирующего сигнала;
- обеспечивал синхронизацию между передатчиком и приемником;
- обладал способностью распознавать ошибки;
- обладал низкой стоимостью реализации.

Синхронизация передатчика и приемника

Синхронизация передатчика и приемника необходима, чтобы приемник точно знал, в какой момент времени необходимо считывать информацию, поступающую по линии связи. Данная проблема в вычислительных сетях решается значительно сложнее, чем при обмене данными между близкорасположенными устройствами.

На относительно небольших расстояниях хорошо зарекомендовала себя схема синхронизации, построенная с использованием отдельной тактирующей линии связи.

Неравномерное распространение сигнала может привести к тому, что тактовый импульс может прийти раньше или позже, в результате чего информация будет потеряна. Это является недостатком данной схемы синхронизации.

Выделенные дополнительные линии, как правило, являются дорогостоящими, что ведет в целом к увеличению стоимости данного способа (метода) передачи.

В вычислительных сетях используются самосинхронизирующиеся коды, которые несут информацию принимающему устройству о том, в какой момент времени необходимо производить считывание очередной порции информации (в какой момент времени вести распознавание очередного бита).

Потенциальный код без возвращения к нулю (с.197)

NRZ. При передаче последовательности единиц сигнал не возвращается к нулю в течение такта. Метод NRZ прост в реализации, обладает хорошей распознаваемостью ошибок (из-за двух резко отличающихся потенциалов), но не обладает свойством самосинхронизации. При передаче длинной последовательности единиц или нулей

сигнал на линии не изменяется, поэтому приемник лишен возможности определять по входному сигналу моменты времени, когда нужно в очередной раз считывать данные.

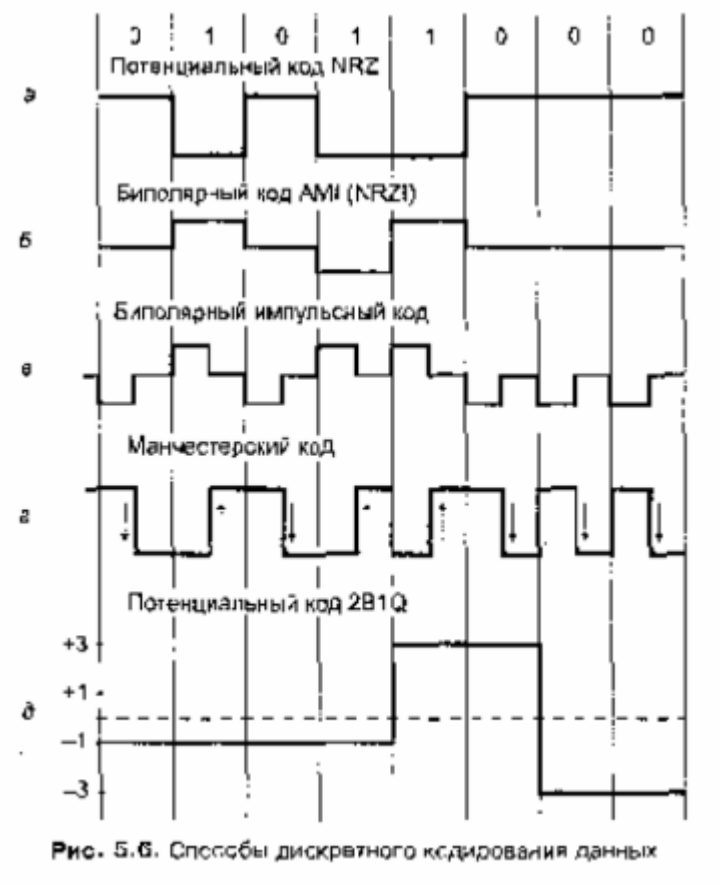


Рис. 5.6. Способы дискретного кодирования данных

Метод биполярного кодирования с альтернативной инверсией (AMI) (с.198)

Одной из модификаций метода NRZ является метод биполярного кодирования с альтернативной инверсией. В этом методе (рис. 5.6, б) используются три уровня потенциала — отрицательный, нулевой и положительный. Для кодирования логического нуля используется нулевой потенциал, а логическая единица кодируется либо положительным потенциалом, либо отрицательным, при этом потенциал каждой новой единицы противоположен потенциалу предыдущей.

Код AMI частично ликвидирует проблемы постоянной составляющей и отсутствия самосинхронизации, присущие коду NRZ. Это происходит при передаче длинных последовательностей единиц. В этих случаях сигнал на линии представляет собой последовательность разнополярных импульсов с тем же спектром, что и у кода NRZ, передающего чередующиеся нули и единицы.

Длинные же последовательности нулей так же опасны для кода AMI, как и для кода NRZ — сигнал вырождается в постоянный потенциал нулевой амплитуды.

В целом, для различных комбинаций битов на линии использование кода AMI приводит к более узкому спектру сигнала, чем для кода NRZ, а значит, и к более высокой пропускной способности линии. Код AMI предоставляет также некоторые возможности по распознаванию ошибочных сигналов. Так, нарушение строгого чередования полярности сигналов говорит о ложном импульсе или исчезновении с линии корректного импульса. Сигнал с некорректной полярностью называется

запрещенным сигналом. В коде AMI используются не два, а три уровня сигнала на линии.

Потенциальный код с инверсией при единице

Существует код, похожий на AMI, но только с двумя уровнями сигнала. При передаче нуля он передает потенциал, который был установлен в предыдущем такте (то есть не меняет его), а при передаче единицы потенциал инвертируется на противоположный. Этот код называется потенциальным кодом с инверсией при единице (NRZI). Он удобен в тех случаях, когда наличие третьего уровня сигнала весьма нежелательно, например в оптических кабелях, где устойчиво распознаются два состояния сигнала — свет и темнота.

Биполярный импульсный код

Кроме потенциальных кодов в сетях используются и импульсные коды, в которых данные представлены полным импульсом или же его частью — фронтом. Наиболее простым случаем такого подхода является биполярный импульсный код, в котором единица представлена импульсом одной полярности, а ноль — другой (рис. 5.6, в). Каждый импульс длится половину такта. Такой код обладает отличными самосинхронизирующими свойствами, но постоянная составляющая может присутствовать, например, при передаче длинной последовательности единиц или нулей. Кроме того, спектр у него ниже, чем у потенциальных кодов. Так, при передаче всех нулей или единиц частота основной гармоники кода будет равна N Гц, что в два раза выше основной гармоники кода NRZ и в четыре раза выше основной гармоники кода AMI при передаче чередующихся единиц и нулей. Из-за слишком широкого спектра биполярный импульсный код используется редко.

Манчестерский код

В локальных сетях до недавнего времени самым распространенным методом кодирования был так называемый манчестерский код (рис. 5.6, в). Он применяется в технологиях Ethernet и Token Ring.

В манчестерском коде для кодирования единиц и нулей используется перепад потенциала, то есть фронт импульса. При манчестерском кодировании каждый такт делится на две части. Информация кодируется перепадами потенциала, происходящими в середине каждого такта. Единица кодируется перепадом от низкого уровня сигнала к высокому, а ноль — обратным перепадом. В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько единиц или нулей подряд. Так как сигнал изменяется по крайней мере один раз за такт передачи одного бита данных, то манчестерский код обладает хорошими самосинхронизирующими свойствами. Полоса пропускания манчестерского кода уже, чем у биполярного импульсного. У него также нет постоянной составляющей, а основная гармоника в худшем случае (при передаче последовательности единиц или нулей) имеет частоту N Гц. В среднем ширина полосы манчестерского кода в полтора раза уже, чем у биполярного импульсного кода, а основная гармоника колеблется вблизи значения 3N/4. Манчестерский код имеет еще одно преимущество перед

биполярным импульсным кодом. В последнем для передачи данных используются три уровня сигнала, а в манчестерском — два.

Потенциальный код 2B1Q

На рис. 5.6, 3 показан потенциальный код с четырьмя уровнями сигнала для кодирования данных. Это код 2B1Q, название которого отражает его суть — каждые два бита (2B) передаются за один такт сигналом, имеющим четыре состояния (1Q). Паре бит 00 соответствует потенциал -2,5 В; паре бит 01 — потенциал -0,833 В; паре 11 — потенциал +0,833 В; а паре 10 — потенциал +2,5 В. При этом способе кодирования требуются дополнительные меры по борьбе с длинными последовательностями одинаковых пар битов, так как при этом сигнал превращается в постоянную составляющую. При случайном чередовании битов спектр сигнала в два раза уже, чем у кода NRZ, так как при той же битовой скорости длительность такта увеличивается в два раза. Таким образом, с помощью кода 2B1Q можно по одной и той же линии передавать данные в два раза быстрее, чем с помощью кода AMI или NRZI. Однако для его реализации мощность передатчика должна быть выше, чтобы четыре уровня четко различались приемником на фоне помех.

Управление каналом связи

Подуровни канального уровня

Подуровень управления логическим каналом (LLC)
- Отвечает за передачу данных (создание кадров, обработка ошибок и т.д.)
- Общий для разных технологий

Подуровень управления доступом к среде (MAC):
- Совместное использование разделяемой среды
- Адресация
- Специфичный для разных технологий
- Не является обязательным

Множественный доступ к каналу связи

Данные искажаются, если несколько компьютеров передают одновременно
- Коллизия

Управление доступом:
- Обеспечение использования канала только одним отправителем

Методы управления доступом:
- Рандомизированный – из N компьютеров выбирается один с вероятностью 1/N. (Ethernet, Wi-Fi).
- На основе правил использования. (Token Ring).

Технологии канального уровня

Ethernet

Wi-Fi

Token Ring

FDDI

ATM

100VG-AnyLAN

Обнаружение и коррекция ошибок (с.215)

Канальный уровень должен обнаруживать ошибки передачи данных, связанные с искажением битов в принятом кадре данных или с потерей кадра, и по возможности их корректировать.

Большая часть протоколов канального уровня выполняет только первую задачу — обнаружение ошибок, считая, что корректировать ошибки, то есть повторно передавать данные, содержавшие искаженную информацию, должны протоколы верхних уровней. Так работают такие популярные протоколы локальных сетей, как Ethernet, Token Ring, FDDI, а также протоколы глобальных сетей frame relay и ATM. Однако существуют протоколы канального уровня, например LLC2 для локальных сетей или HDLC для глобальных, которые самостоятельно решают задачу восстановления искаженных или потерянных кадров.

Очевидно, что протоколы должны работать наиболее эффективно в типичных условиях сети. Поэтому для сетей, в которых искажения и потери кадров являются очень редкими событиями, разрабатываются протоколы, не предусматривающие процедур устранения ошибок. Действительно, наличие процедур восстановления данных потребовало бы от конечных узлов дополнительных вычислительных затрат, которые в условиях надежной работы сети являлись бы избыточными.

Напротив, если в сети искажения и потери случаются часто, то желательно уже на канальном уровне использовать протокол с коррекцией ошибок, а не оставлять эту работу протоколам верхних уровней. Протоколы верхних уровней, например транспортного или прикладного, работая с большими тайм-аутами, восстановят потерянные данные с большой задержкой. В глобальных сетях первых поколений, например сетях X.25, которые работали через ненадежные каналы связи, протоколы канального уровня всегда выполняли процедуры восстановления потерянных и искаженных кадров.

Поэтому нельзя считать, что один протокол лучше другого потому, что он восстанавливает ошибочные кадры, а другой протокол — нет. Каждый протокол должен работать в тех условиях, для которых он разработан.

Методы обнаружения ошибок

Все методы обнаружения ошибок основаны на передаче в составе кадра данных избыточной служебной информации, по которой можно судить с некоторой степенью вероятности о достоверности принятых данных. Эту служебную информацию принято называть контрольной суммой, или последовательностью контроля кадра. Контрольная сумма вычисляется как функция от основной информации, причем необязательно только путем суммироваиия. Принимающая сторона повторно вычисляет контрольную сумму кадра по известному алгоритму и в случае ее совпадения с контрольной суммой, вычисленной передающей стороной, делает вывод о том, что данные были переданы через сеть корректно.

Существует несколько распространенных алгоритмов вычисления контрольной суммы, отличающихся вычислительной сложностью и способностью обнаруживать ошибки в данных.

Контроль по паритету представляет собой наиболее простой метод контроля данных. Метод заключается В суммировании по модулю 2 всех битов контролируемой информации. Например, для данных 100101011 результатом контрольного суммирования будет значение 1. Результат суммирования также представляет собой один бит данных, который пересылается вместе с контролируемой информацией. При искажении в процессе пересылки любого одного бита исходных данных результат суммирования будет отличаться от принятого контрольного разряда, что говорит об ошибке. Однако двойная ошибка будет неверно принята за корректные данные. Поэтому контроль по паритету применяется к небольшим порциям данных, как правило, к каждому байту, что дает коэффициент избыточности для этого метода 1/8. Метод редко применяется в вычислительных сетях из-за значительной избыточности и невысоких диагностических способностей.

Вертикальный и горизонтальный контроль по паритету представляет собой модификацию описанного выше метода. Его отличие состоит в том, что исходные данные рассматриваются в виде матрицы, строки которой составляют байты данных. Контрольный разряд подсчитывается отдельно для каждой строки и для каждого столбца матрицы. Этот метод обнаруживает большую часть двойных ошибокНа практике почти не применяется.

Циклический избыточный контроль является в настоящее время наиболее популярным методом контроля в вычислительных сетях. Метод основан на рассмотрении исходных данных в виде одного многоразрядного двоичного числа. Например, кадр стандарта Ethernet, состоящий из 1024 байт, будет рассматриваться как одно число, состоящее из 8192 бит. В качестве контрольной информации рассматривается остаток от деления этого числа на известный делитель R. Обычно в качестве делителя выбирается семнадцати- или тридцатитрехразряднос число, чтобы остаток от деления имел длину 16 разрядов (2 байт) или 32 разряда (4 байт). При получении кадра данных снова вычисляется остаток от деления на тот же делитель R, но при этом к данным кадра добавляется и содержащаяся в нем контрольная сумма. Если остаток от деления на R равен нулю, то делается вывод об отсутствии ошибок в полученном кадре, в противном случае кадр считается искаженным.

Этот метод обладает более высокой вычислительной сложностью, но его диагностические возможности гораздо выше, чем у методов контроля по паритету.

Метод CRC обнаруживает все одиночные ошибки, двойные ошибки и ошибки в нечетном числе битов. Метод обладает также невысокой степенью избыточности.

Методы восстановления искаженных и потерянных кадров

Методы коррекции ошибок в вычислительных сетях основаны на повторной передаче кадра данных в том случае, если кадр теряется и не доходит до адресата или приемник обнаружил в нем искажение информации. Чтобы убедиться в необходимости

повторной передачи данных, отправитель нумерует отправляемые кадры и для каждого кадра ожидает от приемника так называемой положительной квитанции — служебного кадра, извещающего о том, что исходный кадр был получен и данные в нем оказались корректными. Время этого ожидания ограничено — при отправке каждого кадра передатчик запускает таймер, и если по его истечении положительная квитанция не получена, кадр считается утерянным. Приемник в случае получения кадра с искаженными данными может отправить отрицательную квитанцию — явное указание на то, что данный кадр нужно передать повторно.

Существует два подхода к организации процесса обмена квитанциями: с простоями и с организацией «окна».

Метод с простоями (Idle Source) требует, чтобы источник, пославший кадр, ожидал получения квитанции (положительной или отрицательной) от приемника и только после этого посылал следующий кадр (или повторял искаженный). Если же квитанция не приходит в течение тайм-аута, то кадр (или квитанция) считается утерянным и его передача повторяется.

Второй метод называется методом «скользящего окна» (sliding window). В этом методе для повышения коэффициента использования линии источнику разрешается передать некоторое количество кадров в непрерывном режиме, то есть в максимально возможном для источника темпе, без получения на эти кадры положительных ответных квитанций. Метод скользящего окна более сложен в реализации, чем метод с простоями, так как передатчик должен хранить в буфере все кадры, на которые пока не получены положительные квитанции. Кроме того, требуется отслеживать несколько параметров алгоритма: размер окна w, номер кадра, на который получена квитанция, номер кадра, который еще можно передать до получения повой квитанции.

Коммутация и мультиплексирование.

Мы уже определили термин коммутация как процесс соединения абонентов сети через транзитные узлы. Этим же термином мы обозначаем и соединение интерфейсов в пределах отдельного транзитного узла. Коммутатором в широком смысле слова называется устройство любого типа, способное выполнять операции переключения потока данных с одного интерфейса на другой. Операция коммутации может быть выполнена в соответствии с различными правилами и алгоритмами.

Мультиплексирование и демультиплексирование

Как уже было сказано, прежде чем выполнить переброску данных на определенные для них интерфейсы, коммутатор должен понять, к какому потоку они относятся. Эта задача должна решаться независимо от того, поступает ли на вход коммутатора только один поток в "чистом" виде, или "смешанный" поток, который объединяет в себе несколько потоков. В последнем случае к задаче распознавания добавляется задача демультиплексирования.

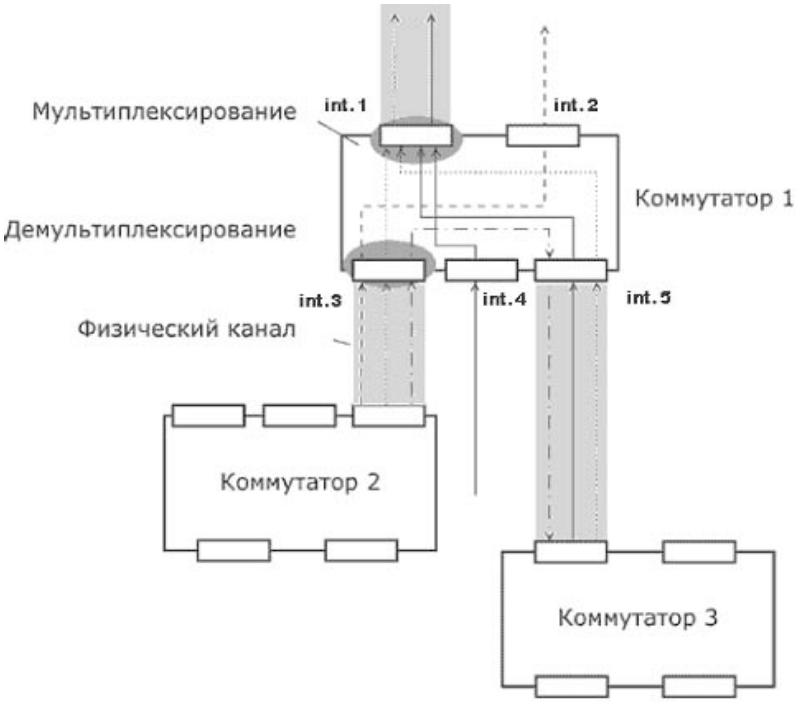
Задача демультиплексирования (demultiplexing) — разделение суммарного агрегированного потока, поступающего на один интерфейс, на несколько составляющих потоков.

Как правило, операцию коммутации сопровождает также обратная операция — мультиплексирование.

Задача мультиплексирования (multiplexing) — образование из нескольких отдельных потоков общего агрегированного потока, который можно передавать по одному физическому каналу связи.

Операции мультиплексирования/демультиплексирования имеют такое же важное значение в любой сети, как и операции коммутации, потому что без них пришлось бы все коммутаторы связывать большим количеством параллельных каналов, что свело бы на нет все преимущества неполносвязной сети.

На рис. 4 показан фрагмент сети, состоящий из трех коммутаторов. Коммутатор 1 имеет пять сетевых интерфейсов. Рассмотрим, что происходит на интерфейсе 1. Сюда поступают данные с трех интерфейсов — int 3, int.4 и int.5. Все их надо передать в общий физический канал, то есть выполнить операцию мультиплексирования. Мультиплексирование представляет собой способ обеспечения доступности имеющихся физических каналов одновременно для нескольких сеансов связи между абонентами сети.

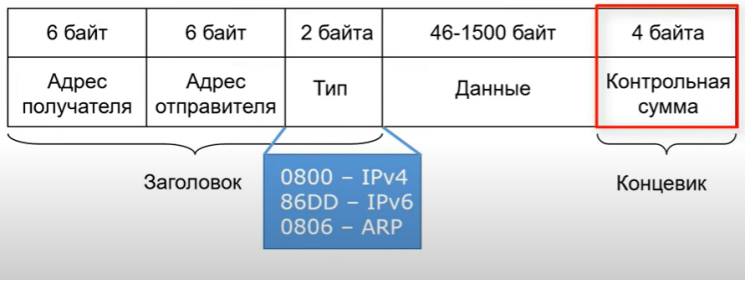


Существует множество способов мультиплексирования потоков в одном физическом канале, и важнейшим из них является разделение времени. При этом способе каждый поток время от времени (с фиксированным или случайным периодом) получает в свое распоряжение физический канал и передает по нему данные. Очень распространено также частотное разделение канала, когда каждый поток передает данные в выделенном ему частотном диапазоне.

Технология мультиплексирования должна позволять получателю такого суммарного потока выполнять обратную операцию — разделение (демультиплексирование) данных на составляющие потоки. На интерфейсе int.3 коммутатор выполняет демультиплексирование потока на три составляющих подпотока. Один из них он передает на интерфейс int. 1, другой на int.2, а третий на int.5. А вот на интерфейсе int.2 нет необходимости выполнять мультиплексирование или демультиплексирование — этот интерфейс выделен одному потоку в монопольное пользование. В общем случае на каждом интерфейсе могут одновременно выполняться обе задачи — мультиплексирование и демультиплексирование

Основы Ethernet (+с.316)

Формат кадра Ethernet



Ethernet - это традиционная технология, используемая для подключения устройств в проводной локальной сети (LAN) или глобальной сети (WAN), позволяющая им обмениваться данными друг с другом через протокол (набор правил или общий сетевой язык). Ethernet описывает, как сетевые устройства форматируют и передают данные, чтобы другие устройства в той же локальной сети или сегменте сети могли идентифицировать, получать и обрабатывать информацию. Кабель Ethernet - это физическая замкнутая проводка, по которой проходят данные.

Инженеры Xerox впервые разработали Ethernet в 1970-х годах. Первоначально Ethernet проходил через коаксиальные кабели. Сегодня в типичных локальных сетях Ethernet используются кабели с витой парой специального класса или оптоволоконные кабели. Ранний Ethernet подключал несколько устройств к сетевому сегменту через гирляндную цепочку или звездообразную топологию через концентратор (устройство уровня 1, отвечающее за передачу сетевых данных). Однако, если два устройства, совместно использующие концентратор, попытаются передать данные одновременно, пакеты данных могут столкнуться и вызвать проблемы с подключением. Чтобы уменьшить эти цифровые пробки, IEEE разработал протокол множественного доступа с контролем несущей (CSMA / CD) с обнаружением коллизий, который позволяет устройствам проверять, используется ли данная линия перед началом новой передачи.

Ethernet используется для подключения устройств в сети и до сих пор остается популярной формой сетевого подключения. Для локальных сетей, используемых определенными организациями (такими как корпоративные офисы, школы и больницы), Ethernet используется из-за его высокой скорости, безопасности и надежности.

Ethernet популярен, потому что он обеспечивает хороший баланс между скоростью, стоимостью и простотой установки. Эти преимущества в сочетании с широким распространением на компьютерном рынке и возможностью поддержки практически всех популярных сетевых протоколов делают Ethernet идеальной сетевой технологией для большинства пользователей компьютеров сегодня.

Институт инженеров по электротехнике и электронике разработал стандарт Ethernet, известный как IEEE 802.3. Этот стандарт определяет правила для настройки сети

Ethernet, а также определяет, как элементы в сети Ethernet взаимодействуют друг с другом. Придерживаясь стандарта IEEE, сетевое оборудование и сетевые протоколы могут эффективно обмениваться данными. Типы Ethernet Стандарт Fast Ethernet (IEEE 802.3u) был создан для сетей Ethernet, которым требуются более высокие скорости передачи. Этот стандарт повышает ограничение скорости Ethernet с 10 Мбит / с до 100 Мбит / с с минимальными изменениями в существующей структуре кабеля. Fast Ethernet обеспечивает более высокую пропускную способность для видео, мультимедиа, графики, просмотра веб-страниц и более надежное обнаружение и исправление ошибок.

Существует три типа Fast Ethernet:

100BASE-TX для использования с кабелем UTP уровня 5;

100BASE-FX для использования с оптоволоконным кабелем;

100BASE-T4, в котором используются два дополнительных провода для использования с кабелем UTP 3-го уровня.

Стандарт 100BASE-TX стал самым популярным из-за его тесной совместимости со стандартом 10BASE-T Ethernet.

Gigabit Ethernet: этот тип сети передает данные с еще более высокой скоростью, около 1000 Мбит/с или 1 Гбит/с. Гигабитная скорость - это модернизация Fast Ethernet, от которого постепенно отказываются. В этом типе сети все четыре пары в кабеле витой пары вносят свой вклад в скорость передачи данных. Он находит широкое применение в системах видеосвязи, в которых используются кабели CAT5e или другие современные кабели. Для расширенных сетей на расстоянии до 500 м можно использовать оптоволоконные кабели 1000Base SX для многомодовых систем, а также 1000Base LX для одномодовых систем. Наиболее важные различия между Gigabit Ethernet и Fast Ethernet включают дополнительную поддержку полнодуплексного режима на уровне MAC и скорости передачи данных.

10 Gigabit Ethernet - это самый быстрый и последний из стандартов Ethernet. IEEE 802.3ae определяет версию Ethernet с номинальной скоростью 10 Гбит / с, что делает его в 10 раз быстрее, чем Gigabit Ethernet. В отличие от других систем Ethernet, 10 Gigabit Ethernet полностью основан на использовании оптоволоконных соединений. В этом развивающемся стандарте происходит переход от дизайна ЛВС, который осуществляет широковещательную рассылку на все узлы, к системе, которая включает некоторые элементы глобальной маршрутизации. Он поддерживается кабелями витой пары CAT6a или CAT7, а также оптоволоконными кабелями. Используя оптоволоконный кабель, эта сетевая зона может быть увеличена примерно до 10 000 метров. Коммутатор Ethernet: для этого типа сети требуется коммутатор или концентратор. Также вместо кабеля витой пары в этом случае используется обычный сетевой кабель. Сетевые коммутаторы используются для передачи данных от одного устройства к другому, не прерывая работу других устройств в сети. Также в сети Ethernet могут использоваться различные типы кабелей.

Logical Link Control protocol (с.280)

LLC (Logical Link Control) — это протокол управления логическим каналом. Как только станция получит разрешение на соединение на уровне MAC, устанавливается логическое соединение между передающей данные станцией и принимающей эти данные станцией. Протокол LLC управляет данным логическим соединением.

Протокол LLC является своеобразным мостом между протоколами сетевого уровня и протоколами уровня MAC. Протоколы сетевого уровня передают через межуровневый интерфейс данные для протокола LLC — свой пакет (например, пакет IP, IPX или NetBEUI), адресную информацию об узле назначения, а также требования к качеству транспортных услуг, которое протокол LLC должен обеспечить. Протокол LLC помещает пакет протокола верхнего уровня в свой кадр, который дополняется необходимыми служебными полями, Далее через межуровневый интерфейс протокол LLC передает свой кадр вместе с адресной информацией об узле назначения соответствующему протоколу уровня MAC, который упаковывает кадр LLC в свой кадр (например, кадр Ethernet).

Протокол LLC записывает информацию, переданную сетевым протоколом, в свой пакет, дополняя его при этом служебной информацией. Далее пакет переходит на уровень MAC, где он преобразуется в кадр уровня MAC (например, в кадр Ethernet), дополненный определенными служебными заголовками, характерными для уровня MAC.

Различные компании использовали различные функции протоколов в своих технологиях. Это привело к необходимости включить в уровень LLC три типа процедур управления передачей данных, которые позволяют выбрать степень надежности передачи:

- LLC1 — процедура без установления соединения и без подтверждения;
- LLC2 — процедура с установлением соединения и с подтверждением;
- LLC3 — процедура без установления соединения, с подтверждением.

Стоит отметить, что протокол сетевого уровня может обратиться только к процедурам одного типа.

Процедура без установления соединения и без подтверждения

Это наименее надежный, но наиболее быстрый способ передачи данных. При этом способе передаче данных данные отправляются вслепую. Если узел назначения не может принять данные — например, он загружен или просто выключен, то данные отправляются «в никуда». Наш узел так и не узнает, получил ли данные узел назначения, поскольку процедура LLC1 не предусматривает подтверждения получения данных.

Данный способ передачи данных называется дейтаграммным (UDP, User Datagram Protocol). Кроме всего прочего он позволяет снизить загруженность канала, поскольку пакеты с подтверждением получения не отправляются.

Процедура с установлением соединения и с подтверждением

LLC2 — наиболее надежный способ передачи данных, поскольку сначала устанавливается логическое соединение с узлом назначения, а потом уже передаются данные, причем каждый переданный пакет подтверждается. Установление соединения позволяет исключить невозможность приема данных узлом назначения. Если узел назначения не может принять данные (например, он выключен), то передача будет прервана. Если узел не получил переданный пакет или пакет в результате передачи был поврежден, то пакет будет передан заново.

Процедура без установления соединения, с подтверждением

В некоторых, достаточно редких случаях, потеря времени на установление соединения просто неприемлема и/или просто не нужна, поскольку точно известно, что узел назначения включен и ожидает передачи данных. В то же время, необходимо знать, получил ли он от нас переданный пакет или нет. Тогда процедуры LLC1 и LLC2 не подходят — нужно использовать LLC3.

Протокол LLC обеспечивает для технологий локальных сетей нужное качество транспортной службы, передавая свои кадры либо дейтаграммным способом, либо с помощью процедур с установлением соединения и восстановлением кадров.

□ LLC предоставляет верхним уровням три типа процедур; процедуру без установления соединения и без подтверждения (LLC1); процедуру с установлением соединения и подтверждением (LLC2); процедуру без установления соединения, но с подтверждением (LLC3).

□ Логический канал протокола LLC2 является дуплексным, так что данные могут передаваться в обоих направлениях.

□ Протокол LLC в режиме LLC2 использует алгоритм скользящего окна.

□ Протокол LLC с помощью управляющих кадров имеет возможность регулировать поток данных, поступающих от узлов сети. Это особенно важно для коммутируемых сетей, в которых нет разделяемой среды, автоматически тормозящей работу передатчика при высокой загрузке сети.

□ В режиме LLC1 единственной функцией протокола LLC является демультиплексирование потока кадров, поступающих из сети, то есть распределения их по протоколам сетевого уровня в соответствии с адресом точки входа сервиса DSAP.

Virtual LAN (с.451) (канальный уровень)

VLAN — это технология, которая позволяет строить виртуальные сети с независимой от физических устройств топологией. например, можно объединить в одну сеть отдел компании, сотрудники которого работают в разных зданиях и подключены к разным коммутаторам. или наоборот, создать отдельные сети для устройств, подключённых к одному коммутатору, если этого требует политика безопасности.

Компьютеры в локальной сети соединяются между собой с помощью сетевого оборудования — коммутаторов. По умолчанию все устройства, подключённые к портам одного коммутатора, могут взаимодействовать, обмениваясь сетевыми пакетами. Любой компьютер может направить широковещательный пакет, адресованный всем устройствам в этой сети, и все остальные компьютеры, подключённые к коммутатору, получат его. Все слышат всех.

Большое количество широковещательных пакетов, отправляемых устройствами, приводит к снижению производительности сети, поскольку вместо полезных операций коммутаторы заняты обработкой данных, адресованных сразу всем.

Чтобы снизить влияние широковещательных рассылок на производительность, сеть разделяют на изолированные сегменты. При этом каждый широковещательный пакет будет распространяться только в пределах сегмента, к которому подключен компьютер-отправитель.

Добиться такого результата можно, подключив разные сегменты к разным физическим коммутаторам, не соединённым между собой, либо соединить их через маршрутизаторы, которые не пропускают широковещательные рассылки.

В основе технологии VLAN лежит стандарт IEEE 802.1Q. Он позволяет добавлять в Ethernet-трафик информацию о принадлежности передаваемых данных к той или иной виртуальной сети — теги VLAN. С их помощью коммутаторы и маршрутизаторы могут выделить из общего потока передаваемых по сети кадров те, что относятся к конкретному сегменту.

Технология VLAN даёт возможность организовать функциональный эквивалент нескольких LAN-сетей без использования набора из коммутаторов и кабелей, которые понадобились бы для их реализации в физическом виде. Физическое сетевое оборудование заменяется виртуальным. Отсюда термин Virtual LAN.

Используя виртуальные локальные сети, можно создавать конфигурации для решения различных задач:

- Объединить в единую сеть группы компьютеров, подключённых к разным коммутаторам.
- Разделить на разные сети компьютеры, подключённые к одному коммутатору.
- Разделить гостевую и корпоративную беспроводную сеть компании.
- Обеспечить взаимодействие территориально распределённых отделов компании как единого целого.

Преимущества VLAN

- Сокращение числа широковещательных запросов, которые снижают пропускную способность сети.
- Повышение безопасности каждой виртуальной сети. Работники одного отдела офиса не смогут отслеживать трафик отделов, не входящих в их VLAN, и не получат доступ к их ресурсам.
- Возможность разделять или объединять отделы или пользователей, территориально удаленных друг от друга. Это позволяет привлекать к рабочему процессу специалистов, не находящихся в здании офиса.
- Создать новую виртуальную сеть можно без прокладки кабеля и покупки коммутатора.
- Позволяет объединить в одну сеть компьютеры, подключенные к разным коммутаторам.
- Упрощение сетевого администрирования. При переезде пользователя VLAN в другое помещение или здание сетевому администратору нет необходимости перекоммутировать кабели, достаточно со своего рабочего места перенастроить сетевое оборудование. А в случае использования динамических VLAN регистрация пользователя в «своём» VLAN на новом месте выполнится автоматически.

FDDI

Технология FDDI (Fiber Distributed Data Interface)- оптоволоконный интерфейс распределенных данных - это первая технология локальных сетей, в которой средой передачи данных является волоконно-оптический кабель.

Технология FDDI во многом основывается на технологии Token Ring, развивая и совершенствуя ее основные идеи. Разработчики технологии FDDI ставили перед собой в качестве наиболее приоритетных следующие цели:

- Повысить битовую скорость передачи данных до 100 Мб/с.
- Повысить отказоустойчивость сети за счет стандартных процедур восстановления ее после отказов различного рода - повреждения кабеля, некорректной работы узла, концентратора, возникновения высокого уровня помех на линии и т.п.
- Максимально эффективно использовать потенциальную пропускную способность сети как для асинхронного, так и для синхронного трафиков.

Сеть FDDI строится на основе двух оптоволоконных колец, которые образуют основной и резервный пути передачи данных между узлами сети. Использование двух колец - это основной способ повышения отказоустойчивости в сети FDDI, и узлы, которые хотят им воспользоваться, должны быть подключены к обоим кольцам. В нормальном режиме работы сети данные проходят через все узлы и все участки кабеля первичного (Primary) кольца, поэтому этот режим назван режимом Thru - "сквозным" или "транзитным". Вторичное кольцо (Secondary) в этом режиме не используется.

В случае какого-либо вида отказа, когда часть первичного кольца не может передавать данные (например, обрыв кабеля или отказ узла), первичное кольцо объединяется со вторичным (рис. 31), образуя вновь единое кольцо. Этот режим работы сети называется Wrap, то есть "свертывание" или "сворачивание" колец. Операция свертывания производится силами концентраторов и/или сетевых адаптеров FDDI. Для упрощения этой процедуры данные по первичному кольцу всегда передаются против часовой стрелки, а по вторичному - по часовой. Поэтому при образовании общего кольца из двух колец передатчики станций по-прежнему остаются подключенными к приемникам соседних станций, что позволяет правильно передавать и принимать информацию соседними станциями.

В стандартах FDDI отводится много внимания различным процедурам, которые позволяют определить наличие отказа в сети, а затем произвести необходимую реконфигурацию. Сеть FDDI может полностью восстанавливать свою работоспособность в случае единичных отказов ее элементов. При множественных отказах сеть распадается на несколько не связанных сетей.

Рис. 3.17. Структура протоколов технологии FDDI

На рис. 3.17 приведено соответствие структуры протоколов технологии FDDI семиуровневой модели OSI. FDDI определяет протокол физического уровня и протокол подуровня доступа к среде (MAC) канального уровня. Как и во многих других технологиях локальных сетей, в технологии FDDI используется протокол подуровня управления каналом данных LLC, определенный в стандарте IEEE 802.2. Таким образом, несмотря на то что технология FDDI была разработана и стандартизована институтом ANSI, а не комитетом IEEE, она полностью вписывается в структуру стандартов 802.

Отличительной особенностью технологии FDDI является уровень управления станцией - Station Management (SMT). Именно уровень SMT выполняет все функции по управлению и мониторингу всех остальных уровней стека протоколов FDDI. В управлении кольцом принимает участие каждый узел сети FDDI. Поэтому все узлы обмениваются специальными кадрами SMT для управления сетью.

Физический уровень разделен на два подуровня: независимый от среды подуровень PHY (Physical), и зависящий от среды подуровень PMD (Physical Media Dependent). Работу всех уровней контролирует протокол управления станцией SMT (Station Management).

Уровень PMD обеспечивает необходимые средства для передачи данных от одной станции к другой по оптоволокну. В его спецификации определяются:

- Требования к мощности оптических сигналов и к многомодовому оптоволоконному кабелю 62.5/125 мкм.
- Требования к оптическим обходным переключателям (optical bypass switches) и оптическим приемопередатчикам.
- Параметры оптических разъемов MIC (Media Interface Connector), их маркировка.
- Длина волны в 1300 нанометров, на которой работают приемопередатчики.
- Представление сигналов в оптических волокнах в соответствии с методом NRZI.

Спецификация TP-PMD определяет возможность передачи данных между станциями по витой паре в соответствии с методом MLT-3. Спецификации уровней PMD и TP-PMD уже были рассмотрены в разделах, посвященных технологии Fast Ethernet.

Уровень PHY выполняет кодирование и декодирование данных, циркулирующих между MAC-уровнем и уровнем PMD, а также обеспечивает тактирование информационных сигналов. В его спецификации определяются:

- кодирование информации в соответствии со схемой 4B/5B;
- правила тактирования сигналов;
- требования к стабильности тактовой частоты 125 МГц;
- правила преобразования информации из параллельной формы в последовательную.

Уровень MAC ответственен за управление доступом к сети, а также за прием и обработку кадров данных. В нем определены следующие параметры:

- Протокол передачи токена.
- Правила захвата и ретрансляции токена.
- Формирование кадра.
- Правила генерации и распознавания адресов.
- Правила вычисления и проверки 32-разрядной контрольной суммы.

Уровень SMT выполняет все функции по управлению и мониторингу всех остальных уровней стека протоколов FDDI. В управлении кольцом принимает участие каждый узел сети FDDI. Поэтому все узлы обмениваются специальными кадрами SMT для управления сетью. В спецификации SMT определено следующее:

- Алгоритмы обнаружения ошибок и восстановления после сбоев.
- Правила мониторинга работы кольца и станций.
- Управление кольцом.
- Процедуры инициализации кольца.

Отказоустойчивость сетей FDDI обеспечивается за счет управления уровнем SMT другими уровнями: с помощью уровня PHY устраняются отказы сети по физическим причинам, например, из-за обрыва кабеля, а с помощью уровня MAC - логические отказы сети, например, потеря нужного внутреннего пути передачи токена и кадров данных между портами концентратора.

Беспроводные сети

Книга 3, с.585.

Беспроводные компьютерные сети — это технология, позволяющая создавать вычислительные сети, полностью соответствующие стандартам для обычных проводных сетей, без использования кабельной проводки. В качестве носителя информации в таких сетях выступают радиоволны СВЧ-диапазона.

Wi-Fi (англ. Wireless Fidelity — «беспроводная точность») — стандарт на оборудование Wireless LAN.

Беспроводная локальная сеть (Wireless Local Area Network; Wireless LAN; WLAN) — локальная сеть, построенная на основе беспроводных технологий. При таком способе построения сетей передача данных осуществляется через радиоэфир; объединение устройств в сеть происходит без использования кабельных соединений. Наиболее распространённым на сегодняшний день способом построения является Wi-Fi.

В стандарте IEEE 802.11 существует два базовых режима: инфраструктура и одноранговая сеть. В одноранговой сети, пользователи напрямую передают информацию друг другу. В инфраструктурном режиме, пользователи общаются через точку доступа, которая служит мостом к другим сетям, таким как интернет или LAN.

Так как беспроводные коммуникации менее защищённые по отношению к проводным, в 802.11 так же включены методы защиты, такие как WEP и WPA.

Инфраструктура

Большинство WLAN устроены по типу инфраструктура.

В режиме инфраструктуры, базовая станция является беспроводной точкой доступа или хабом, пользователи подключаются к хабу. Хаб обычно, но не всегда, имеет проводное подключение к интернету или другой сети.

Иногда WLAN может иметь несколько точек доступа, чтобы покрывать больший радиус, с одинаковым 'SSID' и защитой. В этом случае подключение к любой точке доступа подключает ко всей сети. Устройство пользователя будет пытаться подключиться к ближайшей точке доступа для лучшего качества соединения.

Одноранговая сеть

Одноранговая сеть - сеть, в которой станции общаются только пользователь-к-пользователю (англ. peer to peer). В такой сети нет базы и никто не даёт разрешения на сообщение. Это работает благодаря использованию Independent Basic Service Set (IBSS).

В группе Wi-Fi P2P, владелец группы действует как точка доступа, все остальные устройства - как клиенты. Существует два основных метода выбора владельца группы - выбор владельца пользователем и выбор путём договоров. При выборе путём договоров два устройства сравнивают своё "значение намеренности", устройство с большим значением становится лидером группы. "Значение намеренности" может

зависеть от количества устройств в его зоне покрытия, оставшегося заряда в устройстве и других характеристик.

P2P сеть позволяет беспроводным устройствам напрямую общаться с друг другом. Этот метод обычно используется между двумя компьютерами, чтобы образовать сеть.

Если сила сигнала по каким-то причинам важна, устройства в сети P2P могут ошибаться в определение силы сигнала из-за регистрации сигнала от ближайшего устройства.

В IEEE 802.11 объявлен физический слой и MAC на основе CSMA. В 802.11 предоставлен способ минимизации столкновений, связанных с нахождением двух пользователей в радиусе одной точки доступа, но вне радиуса доступа друг друга.

Мост

Мост может быть использован для соединения сетей, часто разных типов. Беспроводной Ethernet мост позволяет соединять устройства на проводной Ethernet сети с WLAN. Мост является точкой доступа для WLAN.

Структуризация сети: мосты и коммутаторы и их виды

Несмотря на появление новых дополнительных возможностей основной функцией концентраторов остается передача пакетов по общей разделяемой среде. Коллективное использование многими компьютерами общей кабельной системы в режиме разделения времени приводит к существенному снижению производительности сети при интенсивном трафике. Общая среда перестает справляться с потоком передаваемых кадров и в сети возникает очередь компьютеров, ожидающих доступа. Это явление характерно для всех технологий, использующих разделяемые среды передачи данных, независимо от используемых алгоритмов доступа (хотя наиболее страдают от перегрузок трафика сети Ethernet с методом случайного доступа к среде).

Поэтому сети, построенные на основе концентраторов, не могут расширяться в требуемых пределах - при определенном количестве компьютеров в сети или при появлении новых приложений всегда происходит насыщение передающей среды, и задержки в ее работе становятся недопустимыми. Эта проблема может быть решена путем логической структуризации сети с помощью мостов, коммутаторов и маршрутизаторов.

Мост (bridge), а также его быстродействующий функциональный аналог - коммутатор (switching hub), делит общую среду передачи данных на логические сегменты. Логический сегмент образуется путем объединения нескольких физических сегментов (отрезков кабеля) с помощью одного или нескольких концентраторов. Каждый логический сегмент подключается к отдельному порту моста/коммутатора (рис. 1.10). При поступлении кадра на какой-либо из портов мост/коммутатор повторяет этот кадр, но не на всех портах, как это делает концентратор, а только на том порту, к которому подключен сегмент, содержащий компьютер-адресат.

Разница между мостом и коммутатором состоит в том, что мост в каждый момент времени может осуществлять передачу кадров только между одной парой портов, а коммутатор одновременно поддерживает потоки данных между всеми своими портами. Другими словами, мост передает кадры последовательно, а коммутатор параллельно. (Для упрощения изложения далее в этом разделе будет использоваться термин "коммутатор" для обозначения этих обоих разновидностей устройств, поскольку все сказанное ниже в равной степени относится и к мостам, и к коммутаторам.) Следует отметить, что в последнее время локальные мосты полностью вытеснены коммутаторами. Мосты используются только для связи локальных сетей с глобальными, то есть как средства удаленного доступа, поскольку в этом случае необходимость в параллельной передаче между несколькими парами портов просто не возникает.

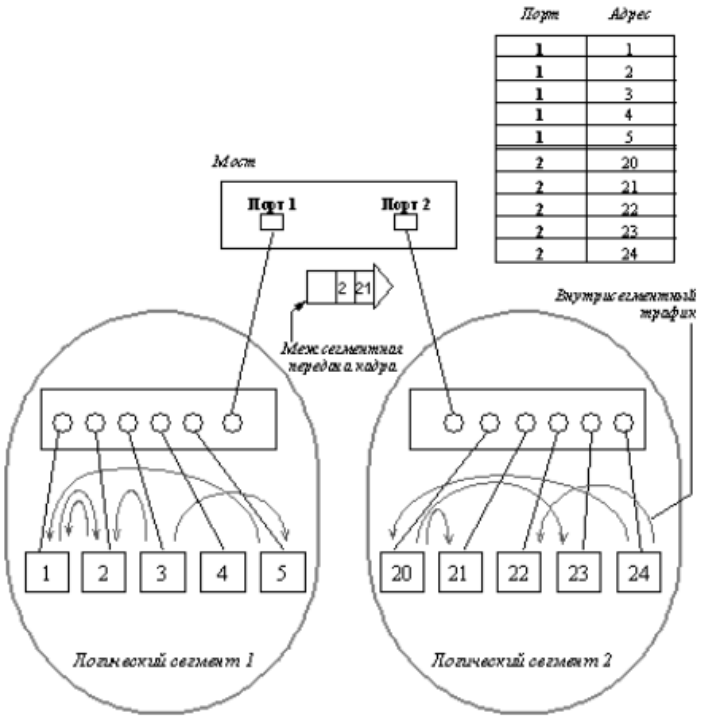


Рис. 1.10. Разделение сети на логические сегменты

При работе коммутатора среда передачи данных каждого логического сегмента остается общей только для тех компьютеров, которые подключены к этому сегменту непосредственно. Коммутатор осуществляет связь сред передачи данных различных логических сегментов. Он передает кадры между логическими сегментами только при необходимости, то есть только тогда, когда взаимодействующие компьютеры находятся в разных сегментах.

Деление сети на логические сегменты улучшает производительность сети, если в сети имеются группы компьютеров, преимущественно обменивающиеся информацией между собой. Если же таких групп нет, то введение в сеть коммутаторов может только ухудшить общую производительность сети, так как принятие решения о том, нужно ли передавать пакет из одного сегмента в другой, требует дополнительного времени.

Однако даже в сети средних размеров такие группы, как правило, имеются. Поэтому разделение ее на логические сегменты дает выигрыш в производительности - трафик локализуется в пределах групп, и нагрузка на их разделяемые кабельные системы существенно уменьшается.

Коммутаторы принимают решение о том, на какой порт нужно передать кадр, анализируя адрес назначения, помещенный в кадре, а также на основании информации о принадлежности того или иного компьютера определенному сегменту, подключенному к одному из портов коммутатора, то есть на основании информации о конфигурации сети. Для того, чтобы собрать и обработать информацию о конфигурации подключенных к нему сегментов, коммутатор должен пройти стадию "обучения", то есть самостоятельно проделать некоторую предварительную работу по изучению проходящего через него трафика. Определение принадлежности компьютеров сегментам возможно за счет наличия в кадре не только адреса

назначения, но и адреса источника, сгенерировавшего пакет. Используя информацию об адресе источника, коммутатор устанавливает соответствие между номерами портов и адресами компьютеров. В процессе изучения сети мост/коммутатор просто передает появляющиеся на входах его портов кадры на все остальные порты, работая некоторое время повторителем. После того, как мост/коммутатор узнает о принадлежности адресов сегментам, он начинает передавать кадры между портами только в случае межсегментной передачи. Если, уже после завершения обучения, на входе коммутатора вдруг появится кадр с неизвестным адресом назначения, то этот кадр будет повторен на всех портах.

Мосты/коммутаторы, работающие описанным способом, обычно называются прозрачными (transparent), поскольку появление таких мостов/коммутаторов в сети совершенно не заметно для ее конечных узлов. Это позволяет не изменять их программное обеспечение при переходе от простых конфигураций, использующих только концентраторы, к более сложным, сегментированным.

Существует и другой класс мостов/коммутаторов, передающих кадры между сегментами на основе полной информации о межсегментном маршруте. Эту информацию записывает в кадр станция-источник кадра, поэтому говорят, что такие устройства реализуют алгоритм маршрутизации от источника (source routing). При использовании мостов/коммутаторов с маршрутизацией от источника конечные узлы должны быть в курсе деления сети на сегменты и сетевые адаптеры, в этом случае должны в своем программном обеспечении иметь компонент, занимающийся выбором маршрута кадров.

За простоту принципа работы прозрачного моста/коммутатора приходится расплачиваться ограничениями на топологию сети, построенной с использованием устройств данного типа - такие сети не могут иметь замкнутых маршрутов - петель. Мост/коммутатор не может правильно работать в сети с петлями, при этом сеть засоряется зацикливающимися пакетами и ее производительность снижается.

Коммутация с полной буферизацией и «на лету». Spanning Tree Protocol.

На производительности коммутатора сказывается способ передачи пакетов - «на лету» или с буферизацией. Коммутаторы, передающие пакеты «на лету», вносят меньшие задержки передачи кадров на каждом промежуточном коммутаторе, поэтому общее уменьшение задержки доставки данных может быть значительным, что важно для мультимедийного трафика. Кроме того, выбранный способ коммутации оказывает влияние на возможности реализации некоторых полезных дополнительных функций, например трансляцию протоколов канального уровня. В табл. 4.2 дается сравнение возможностей двух способов коммутации.

Таблица 4.2. Возможности коммутаторов при коммутации «на лету» и с полной буферизацией

Функция	На лету	С буферизацией
Защита от плохих кадров	Нет	Да
Поддержка разнородных сетей (Ethernet, Token Ring, FDDI, ATM)	Нет	Да
Задержка передачи пакетов	Низкая (5–40 мкс) при низкой нагрузке, средняя при высокой нагрузке	Средняя при любой нагрузке
Поддержка резервных связей	Нет	Да
Функция анализа трафика	Нет	Да

Средняя величина задержки коммутаторов, работающих «на лету», при высокой нагрузке объясняется тем, что в этом случае выходной порт часто бывает занят приемом другого пакета, поэтому вновь поступивший пакет для данного порта все равно приходится буферизовать.

Коммутатор, работающий «на лету», может выполнять проверку некорректности передаваемых кадров, но не может изъять плохой кадр из сети, так как часть его байт (и, как правило, большая часть) уже переданы в сеть.

Так как каждый способ имеет свои достоинства и недостатки, в тех моделях коммутаторов, которым н нужно транслировать протоколы, иногда применяется механизм адаптивной смены режима работы коммутатора. Основной режим такого коммутатора - коммутация «на лету», но коммутатор постоянно контролирует трафик и при превышении интенсивности появления плохих кадров некоторого порога переходит на режим полной буферизации. Затем коммутатор может вернуться к коммутации «на лету».

Алгоритм покрывающего дерева - Spanning Tree Algorithm (STA) позволяет коммутаторам автоматически определять древовидную конфигурацию связей в сети при произвольном соединения портов между собой. Как уже отмечалось, для нормальной работы коммутатора требуется отсутствие замкнутых маршрутов в сети. Эти маршруты могут создаваться администратором специально для образования резервных связей или же возникать случайным образом, что вполне возможно, если сеть имеет многочисленные связи, а кабельная система плохо структурирована или документирована.

Поддерживающие алгоритм STA коммутаторы автоматически создают активную древовидную конфигурацию связей (то есть связную конфигурацию без петель) на множестве всех связей сети. Такая конфигурация называется покрывающим деревом - Spanning Tree (иногда ее называют основным деревом), и ее название дало имя всему алгоритму. Алгоритм Spanning Tree описан в стандарте IEEE 802.1D, том же стандарте, который определяет принципы работы прозрачных мостов.

Коммутаторы находят покрывающее дерево адаптивно, с помощью обмена служебными пакетами. Реализация в коммутаторе алгоритма STA очень важна для работы в больших сетях - если коммутатор не поддерживает этот алгоритм, то администратор должен самостоятельно определить, какие порты нужно перевести в заблокированное состояние, чтобы исключить петли. К тому же при отказе какого-либо кабеля, порта или коммутатора администратор должен, во-первых, обнаружить факт отказа, а во-вторых, ликвидировать последствия отказа, переведя резервную связь в рабочий режим путем активизации некоторых портов. При поддержке коммутаторами сети протокола Spanning Tree отказы обнаруживаются автоматически, за счет постоянного тестирования связности сети служебными пакетами. После обнаружения потери связности протокол строит новое покрывающее дерево, если это возможно, и сеть автоматически восстанавливает работоспособность.

Алгоритм Spanning Tree определяет активную конфигурацию сети за три этапа.

Сначала в сети определяется корневой коммутатор (root switch), от которого строится дерево. Корневой коммутатор может быть выбран автоматически или назначен администратором. При автоматическом выборе корневым становится коммутатор с меньшим значением MAC - адреса его блока управления.

Затем, на втором этапе, для каждого коммутатора определяется корневой порт (root port) - это порт, который имеет по сети кратчайшее расстояние до корневого коммутатора (точнее, до любого из портов корневого коммутатора).

И наконец, на третьем этапе для каждого сегмента сети выбирается так называемый назначенный порт (designated port) - это порт, который имеет кратчайшее расстояние от данного сегмента до корневого коммутатора. После определения корневых и назначенных портов каждый коммутатор блокирует остальные порты, которые не попали в эти два класса портов. Можно математически доказать, что при таком выборе активных портов в сети исключаются петли и оставшиеся связи образуют покрывающее дерево (если оно может быть построено при существующих связях в сети).

Понятие расстояния играет важную роль в построении покрывающего дерева. Именно по этому критерию выбирается единственный порт, соединяющий каждый коммутатор с корневым коммутатором, и единственный порт, соединяющий каждый сегмент сети с корневым коммутатором.

Для автоматического определения начальной активной конфигурации дерева все коммутаторы сети после их инициализации начинают периодически обмениваться специальными пакетами, называемыми протокольными блоками данных моста - BPDU (Bridge Protocol Data Unit), что отражает факт первоначальной разработки алгоритма STA для мостов.

Пакеты BPDU помещаются в поле данных кадров канального уровня, например кадров Ethernet или FDDI. Желательно, чтобы все коммутаторы поддерживали общий групповой адрес, с помощью которого кадры, содержащие пакеты BPDU, могли бы одновременно передаваться всем коммутаторам сети. Иначе пакеты BPDU рассылаются широковещательно.

После инициализации каждый коммутатор сначала считает себя корневым. Поэтому он начинает через интервал hello генерировать через все свои порты сообщения BPDU конфигурационного типа. В них он указывает свой идентификатор в качестве идентификатора корневого коммутатора (и в качестве идентификатора данного коммутатора также), расстояние до корня устанавливается в 0, а в качестве идентификатора порта указывается идентификатор того порта, через который передается BPDU. Как только коммутатор получает BPDU, в котором имеется идентификатор корневого коммутатора, со значением, меньшим его собственного, он перестает генерировать свои собственные кадры BPDU, а начинает ретранслировать только кадры нового претендента на звание корневого коммутатора.

При ретрансляции кадров каждый коммутатор наращивает расстояние до корня, указанное в пришедшем BPDU, на условное время сегмента, по которому принят данный кадр. Тем самым в кадре BPDU, по мере прохождения через коммутаторы, накапливается расстояние до корневого коммутатора.

Ретранслируя кадры, каждый коммутатор для каждого своего порта запоминает минимальное расстояние до корня, встретившееся во всех принятых этим портом кадрах BPDU. При завершении процедуры установления конфигурации покрывающего дерева (по времени) каждый коммутатор находит свой корневой порт - это порт, для которого минимальное расстояние до корня оказалось меньше, чем у других портов.

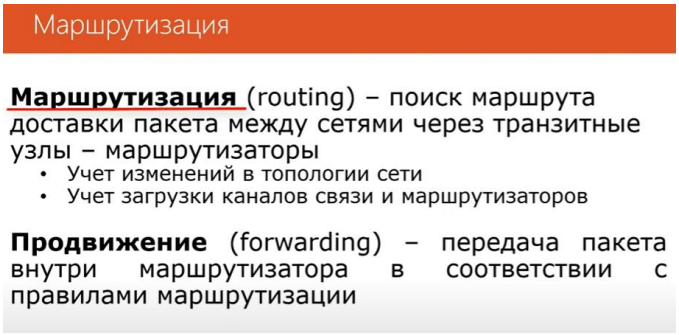
Кроме корневого порта коммутаторы распределенным образом выбирают для каждого сегмента сети назначенный порт. Для этого они исключают из рассмотрения свой корневой порт (для сегмента, к которому он подключен, всегда существует другой коммутатор, который ближе расположен к корню), а для всех своих оставшихся портов сравнивают принятые по ним минимальные расстояния до корня с расстоянием до корня своего корневого порта. Если у какого-либо своего порта принятые им расстояния до корня больше, чем расстояние маршрута, пролегающего через свой корневой порт, то это значит, что для сегмента, к которому подключен данный порт, кратчайшее расстояние к корневому коммутатору ведет именно через данный порт. Коммутатор делает все свои порты, у которых такое условие выполняется, назначенными.

Если в процессе выбора корневого порта или назначенного порта несколько портов оказываются равными по критерию кратчайшего расстояния до корневого коммутатора, то выбирается порт с наименьшим идентификатором.

Затем все порты, кроме корневого и назначенных, переводятся каждым коммутатором в заблокированное состояние. На этом построение покрывающего дерева заканчивается.

В процессе нормальной работы корневой коммутатор продолжает генерировать служебные кадры BPDU, а остальные коммутаторы продолжают их принимать своими корневыми портами и ретранслировать назначенными. Если у коммутатора нет назначенных портов, как у коммутаторов 2 и 4, то они все равно продолжают принимать участие в работе протокола Spanning Tree, принимая служебные кадры корневым портом. Если по истечении тайм-аута корневой порт любого коммутатора сети не получает служебный кадр BPDU, то он инициализирует новую процедуру построения покрывающего дерева, оповещая об этом другие коммутаторы BPDU уведомления о реконфигурации. Получив такой кадр, все коммутаторы начинают снова генерировать BDPU конфигурационного типа, в результате чего устанавливается новая активная конфигурация.

Идея Internetworking. Маршрутизация и функции маршрутизатора (с.468)



Понятие internetworking

Основная идея введения сетевого уровня состоит в следующем. Сеть в общем случае рассматривается как совокупность нескольких сетей и называется составной сетью или интерсетью (internetwork или internet). Сети, входящие в составную сеть, называются подсетями (subnet), составляющими сетями или просто сетями.

Подсети соединяются между собой маршрутизаторами. Компонентами составной сети могут являться как локальные, так и глобальные сети. Все узлы в пределах одной подсети взаимодействуют, используя единую для них технологию. локальные сети Ethernet, Fast Ethernet, Token Ring, FDDI и глобальные сети frame relay, X.25, ISDN. Каждая из этих технологий достаточна для того, чтобы организовать взаимодействие всех узлов в своей подсети, но не способна построить информационную связь между произвольно выбранными узлами, принадлежащими разным подсетям. Следовательно, для организации взаимодействия между любой произвольной парой узлов этой «большой» составной сети требуются дополнительные средства. Такие средства и предоставляет сетевой уровень.

Сетевой уровень выступает в качестве координатора, организующего работу всех подсетей, лежащих на пути продвижения пакета по составной сети. Для перемещения данных в пределах подсетей сетевой уровень обращается к используемым в этих подсетях технологиям.

Хотя многие технологии локальных сетей (Ethernet, Token Ring, FDDI, Fast Ethernet и др.) используют одну и ту же систему адресации узлов на основе MAC - адресов, существует немало технологий (X.25, ATM, frame relay), в которых применяются другие схемы адресации. Адреса, присвоенные узлам в соответствии с технологиями подсетей, называют локальными. Чтобы сетевой уровень мог выполнить свою задачу, ему необходима собственная система адресации, не зависящая от способов адресации узлов в отдельных подсетях, которая позволила бы на сетевом уровне универсальным и однозначным способами идентифицировать любой узел составной сети.

Естественным способом формирования сетевого адреса является уникальная нумерация всех подсетей составной сети и нумерация всех узлов в пределах каждой подсети. Таким образом, сетевой адрес представляет собой пару: номер сети (подсети) и номер узла.

Данные, которые поступают на сетевой уровень и которые необходимо передать через составную сеть, снабжаются заголовком сетевого уровня. Данные вместе с заголовком образуют пакет. Заголовок пакета сетевого уровня имеет унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, которые могут входить в объединенную сеть, и несет наряду с другой служебной информацией данные о номере сети, которой предназначается этот пакет. Сетевой уровень определяет маршрут и перемещает пакет между подсетями.

При передаче пакета из одной подсети в другую пакет сетевого уровня, инкапсулированный в прибывший канальный кадр первой подсети, освобождается от заголовков этого кадра и окружается заголовками кадра канального уровня следующей подсети. Информацией, на основе которой делается эта замена, являются служебные поля пакета сетевого уровня. В поле адреса назначения нового кадра указывается локальный адрес следующего маршрутизатора.

Основным полем заголовка сетевого уровня является номер сети-адресата.

Кроме номера сети заголовок сетевого уровня должен содержать и другую информацию, необходимую для успешного перехода пакета из сети одного типа в сеть другого типа. К такой информации может относиться, например:

- номер фрагмента пакета, необходимый для успешного проведения операций сборки-разборки фрагментов при соединении сетей с разными максимальными размерами пакетов;
- время жизни пакета, указывающее, как долго он путешествует по интерсети, это время может использоваться для уничтожения «заблудившихся» пакетов;
- качество услуги - критерий выбора маршрута при межсетевых передачах

Когда две или более сети организуют совместную транспортную службу, то такой режим взаимодействия обычно называют межсетевым взаимодействием (internetworking).

Принципы маршрутизации

Важнейшей задачей сетевого уровня является маршрутизация - передача пакетов между двумя конечными узлами в составной сети.

Маршрутизаторы имеют по нескольку портов (по крайней мере, по два), к которым присоединяются сети. Каждый порт маршрутизатора можно рассматривать как отдельный узел сети: он имеет собственный сетевой адрес и собственный локальный адрес в той подсети, которая к нему подключена.

В сложных составных сетях почти всегда существует несколько альтернативных маршрутов для передачи пакетов между двумя конечными узлами. Маршрут - это последовательность маршрутизаторов, которые должен пройти пакет от отправителя до пункта назначения.

Задачу выбора маршрута из нескольких возможных решают маршрутизаторы, а также конечные узлы. Маршрут выбирается на основании имеющейся у этих устройств информации о текущей конфигурации сети, а также на основании указанного критерия выбора маршрута.

Чтобы по адресу сети назначения можно было бы выбрать рациональный маршрут дальнейшего следования пакета, каждый конечный узел и маршрутизатор анализируют специальную информационную структуру, которая называется таблицей маршрутизации.

В первом столбце таблицы перечисляются номера сетей, входящих в интерсеть. В каждой строке таблицы следом за номером сети указывается сетевой адрес следующего маршрутизатора (более точно, сетевой адрес соответствующего порта следующего маршрутизатора), на который надо направить пакет, чтобы тот передвигался по направлению к сети с данным номером по рациональному маршруту.

Когда на маршрутизатор поступает новый пакет, номер сети назначения, извлеченный из поступившего кадра, последовательно сравнивается с номерами сетей из каждой строки таблицы. Строка с совпавшим номером сети указывает, на какой ближайший маршрутизатор следует направить пакет.

Поскольку пакет может быть адресован в любую сеть составной сети, может показаться, что каждая таблица маршрутизации должна иметь записи обо всех сетях, входящих в составную сеть. Но при таком подходе в случае крупной сети объем таблиц маршрутизации может оказаться очень большим, что повлияет на время ее просмотра, потребует много места для хранения и т. п. Поэтому на практике число записей в таблице маршрутизации стараются уменьшить за счет использования специальной записи - «маршрутизатор по умолчанию» (default). Действительно, если принять во внимание топологию составной сети, то в таблицах маршрутизаторов, находящихся на периферии составной сети, достаточно записать номера сетей, непосредственно подсоединенных к данному маршрутизатору или расположенных поблизости, на тупиковых маршрутах. Обо всех же остальных сетях можно сделать в таблице единственную запись, указывающую на маршрутизатор, через который пролегает путь ко всем этим сетям. Такой маршрутизатор называется маршрутизатором по умолчанию, а вместо номера сети в соответствующей строке помещается особая запись, например default.

Перед тем как передать пакет следующему маршрутизатору, текущий маршрутизатор должен определить, на какой из нескольких собственных портов он должен поместить данный пакет. Для этого служит третий столбец таблицы маршрутизации. Еще раз подчеркнем, что каждый порт идентифицируется собственным сетевым адресом.

Некоторые реализации сетевых протоколов допускают наличие в таблице маршрутизации сразу нескольких строк, соответствующих одному и тому же адресу сети назначения. В этом случае при выборе маршрута принимается во внимание столбец «Расстояние до сети назначения».

Наличе нескольких маршрутов к одному узлу делают возможным передачу трафика к этому узлу параллельно по нескольким каналам связи, это повышает пропускную способность и надежность сети.

Задачу маршрутизации решают не только промежуточные узлы - маршрутизаторы, но и конечные узлы - компьютеры. Средства сетевого уровня, установленные на конечном узле, при обработке пакета должны, прежде всего, определить, направляется ли он в другую сеть или адресован какому-нибудь узлу данной сети. Если номер сети назначения совпадает с номером данной сети, то для данного пакета не требуется решать задачу маршрутизации. Если же номера сетей отправления и назначения не совпадают, то маршрутизация нужна. Таблицы маршрутизации конечных узлов полностью аналогичны таблицам маршрутизации, хранящимся на маршрутизаторах.

Конечные узлы в еще большей степени, чем маршрутизаторы, пользуются приемом маршрутизации по умолчанию.

Основная функция маршрутизатора - чтение заголовков пакетов сетевых протоколов, принимаемых и буферизуемых по каждому порту, и принятие решения о дальнейшем маршруте следования пакета по его сетевому адресу, включающему, как правило, номер сети и номер узла.

Функции маршрутизатора могут быть разбиты на 3 группы в соответствии с уровнями модели OSI

Уровень интерфейсов

На нижнем уровне маршрутизатор, как и любое устройство, подключенное к сети, обеспечивает физический интерфейс со средой передачи, включая согласование уровней электрических сигналов, линейное и логическое кодирование, оснащение определенным типом разъема. В разных моделях маршрутизаторов часто предусматриваются различные наборы физических интерфейсов, представляющих собой комбинацию портов для подсоединения локальных и глобальных сетей. Интерфейсы для присоединения к глобальным сетям чаще всего определяют только некоторый стандарт физического уровня, над которым в маршрутизаторе могут работать различные протоколы канального уровня. Разница между интерфейсами локальных и глобальных сетей объясняется тем, что технологии локальных сетей работают по собственным стандартам физического уровня, которые не могут, как правило, использоваться в других технологиях, поэтому интерфейс для локальной сети представляет собой сочетание физического и канального уровней и носит название по имени соответствующей технологии - например, интерфейс Ethernet.

Интерфейсы маршрутизатора выполняют полный набор функций физического и канального уровней по передаче кадра, включая получение доступа к среде (если это необходимо), формирование битовых сигналов, прием кадра, подсчет его контрольной суммы и передачу поля данных кадра верхнему уровню, в случае если контрольная сумма имеет корректное значение.

Перечень физических интерфейсов, которые поддерживает та или иная модель маршрутизатора, является его важнейшей потребительской характеристикой. Маршрутизатор должен поддерживать все протоколы канального и физического уровней, используемые в каждой из сетей, к которым он будет непосредственно присоединен.

Кадры, которые поступают на порты маршрутизатора, после обработки соответствующими протоколами физического и канального уровней, освобождаются от заголовков канального уровня. Извлеченные из поля данных кадра пакеты передаются модулю сетевого протокола.

Уровень сетевого протокола

Сетевой протокол в свою очередь извлекает из пакета заголовок сетевого уровня и анализирует содержимое его полей. Прежде всего проверяется контрольная сумма, и если пакет пришел поврежденным, то он отбрасывается. Выполняется проверка, не превысило ли время, которое провел пакет в сети (время жизни пакета), допустимой величины. Если превысило - то пакет также отбрасывается. На этом этапе вносятся корректировки в содержимое некоторых полей, например, наращивается время жизни пакета, пересчитывается контрольная сумма.

На сетевом уровне выполняется одна из важнейших функций маршрутизатора - фильтрация трафика. Маршрутизатор, обладая более высоким интеллектом, нежели мосты и коммутаторы, позволяет задавать и может отрабатывать значительно более сложные правила фильтрации. Пакет сетевого уровня, находящийся в поле данных кадра, для мостов/коммутаторов представляется неструктурированной двоичной последовательностью. Маршрутизаторы же, программное обеспечение которых содержит модуль сетевого протокола, способны производить разбор и анализ отдельных полей пакета. Они оснащаются развитыми средствами пользовательского интерфейса, которые позволяют администратору без особых усилий задавать сложные правила фильтрации. Маршрутизаторы, как правило, также могут анализировать структуру сообщений транспортного уровня.

В случае если интенсивность поступления пакетов выше интенсивности, с которой они обрабатываются, пакеты могут образовать очередь.

К сетевому уровню относится основная функция маршрутизатора - определение маршрута пакета. По номеру сети, извлеченному из заголовка пакета, модуль сетевого протокола находит в таблице маршрутизации строку, содержащую сетевой адрес следующего маршрутизатора, и номер порта, на который нужно передать данный пакет, чтобы он двигался в правильном направлении. Если в таблице отсутствует запись о сети назначения пакета и к тому же нет записи о маршрутизаторе по умолчанию, то данный пакет отбрасывается.

Перед тем как передать сетевой адрес следующего маршрутизатора на канальный уровень, необходимо преобразовать его в локальный адрес той технологии, которая

используется в сети, содержащей следующий маршрутизатор. Для этого сетевой протокол обращается к протоколу разрешения адресов. Протоколы этого типа устанавливают соответствие между сетевыми и локальными адресами либо на основании заранее составленных таблиц, либо путем рассылки широковещательных запросов. Таблица соответствия локальных адресов сетевым адресам строится отдельно для каждого сетевого интерфейса. Протоколы разрешения адресов занимают промежуточное положение между сетевым и канальным уровнями.

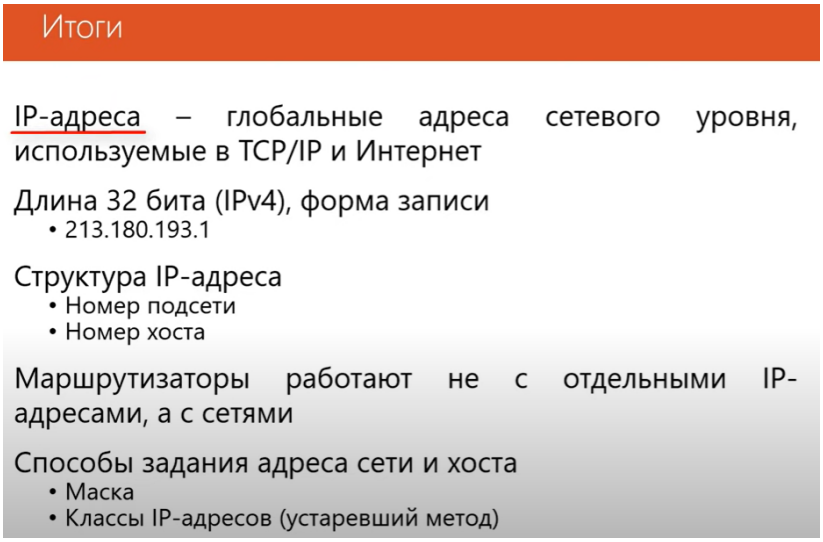
С сетевого уровня пакет, локальный адрес следующего маршрутизатора и номер порта маршрутизатора передаются вниз, канальному уровню. На основании указанного номера порта осуществляется коммутация с одним из интерфейсов маршрутизатора, средствами которого выполняется упаковка пакета в кадр соответствующего формата. В поле адреса назначения заголовка кадра помещается локальный адрес следующего маршрутизатора. Готовый кадр отправляется в сеть.

Уровень протоколов маршрутизации

Сетевые протоколы активно используют в своей работе таблицу маршрутизации, но ни ее построением, ни поддержанием ее содержимого не занимаются. Эти функции выполняют протоколы маршрутизации. На основании этих протоколов маршрутизаторы обмениваются информацией о топологии сети, а затем анализируют полученные сведения, определяя наилучшие по тем или иным критериям маршруты. Результаты анализа и составляют содержимое таблиц маршрутизации.

+Помимо перечисленных выше функций, на маршрутизаторы могут быть возложены и другие обязанности, например операции, связанные с фрагментацией.

Модель DOD. Классовая и бесклассовая адресация



Модель DOD или модель TCP/IP - модель сетевого взаимодействия, разработанная Министерством обороны США, практической реализацией которой является стек протоколов TCP/IP.

Стек протоколы TCP/IP - это набор сетевых протоколов передачи данных, используемых в сетях.

В отличие от модели OSI, модель DOD состоит из четырёх уровней (Рисунок 10):

- Уровень приложений
- Транспортный уровень
- Межсетевой уровень
- Уровень сетевого доступа

Уровень приложений или прикладной уровень. Верхний уровень модели, включающий протоколы, обрабатывающие данные пользователей и осуществляющие управление обменом данными между приложениями. На этом уровне стандартизируется представление данных.

Транспортный уровень, содержит протоколы для обеспечения целостности данных при сквозной передаче. Обеспечивает управление инициализацией и закрытием соединений.

Межсетевой уровень- уровень сети Интернет, содержит протоколы для маршрутизации сообщений в сети.

Уровень сетевого доступа. Нижний уровень модели. Содержит протоколы для физической доставки данных к сетевым устройствам. Этот уровень размещает данные в фрейме.

В сетях адресация осуществляется при помощи маски подсетей.

Маской подсети или маской сети - битовая маска, определяющая, 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 12.34.56.0/24 с длиной префикса 24 бита.

Длина префикса - количество двоичных единиц в маске подсети. Например, маской подсети 255.255.255.0 в двоичном виде 11111111 11111111 11111111 00000000, количество единиц 24 значит, длина префикса равняется 24.

IP- адресация разделяется на два вида:

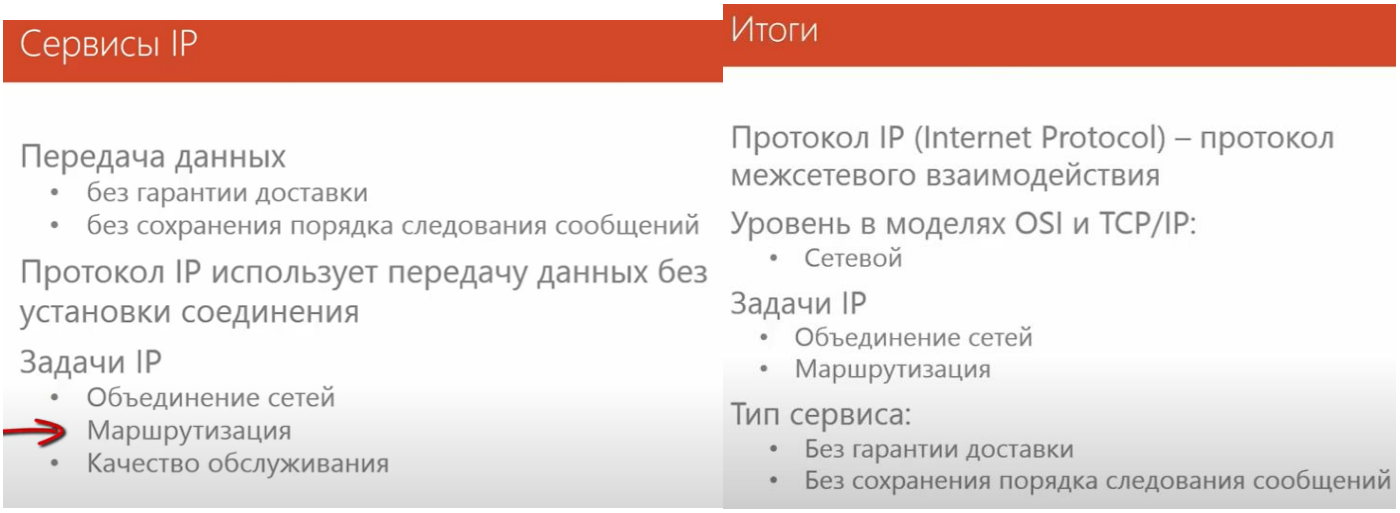
- Классовая адресация
- Бесклассовая адресация

Классовая адресация сетей - метод IP-адресации, работающий с одним классом сети. В классовой адресации все подсети в ходящие в локальную сеть должны иметь одну маску подсети. Если используется разные маску подсети то пакеты отправленные из подсети в другую будут уничтожаться, не доходя до получателя.

Бесклассовая адресация (англ. Classless Inter-Domain Routing (CIDR)) - метод IP-адресации, позволяющий гибко управлять пространством IP-адресов. В бесклассовой адресации к каждой подсети локальной сети возможно применение различных масок подсетей (Рисунок 12). Количество адресов подсети не равно количеству возможных узлов. Начальный адрес сети резервируется для идентификации подсети, последний - в качестве широковещательного адреса.

В википедии хорошо написано про адресацию.

Протоколы IP(v4,v6) и ARP (с.371) (с.487)



IP-адрес. имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например, 128.10.2.30 — традиционная десятичная форма представления адреса, а 10000000 00001010 00000010 00011110 — двоичная форма представления этого же адреса.

Адрес состоит из двух логических частей — номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая — к номеру узла, определяется значениями первых бит адреса. Значения этих бит являются также признаками того, к какому классу относится тот или иной IP-адрес.

Первая функция протокола IP заключается в том, чтобы дать уникальные имена узлам в компьютерной сети.

Вторая функция протокола IP заключает в том, чтобы предоставить услугу вышестоящему уровню, то есть транспортному уровню. Тут нужно заметить, что модель TCP/IP не предполагает взаимодействие с установлением соединения на сетевом уровне, вид взаимодействия определяется транспортным уровнем, следовательно сам протокол IP работает без установления соединения и как бы это странно не прозвучала, но услугой протокола IP для транспортного уровня является передача данных между сетями или же транспортировка.

Начнем с IPv4. Больше всего нас интересует IP-адрес, под него в IPv4 выделено четыре байта или октета, как известно, в байте 8 бит, то есть восемь двоичных значений (0 или 1), следовательно, максимально возможное десятичное число равно 255, минимально допустимое 0.

Обмен информацией в IPv4 происходит при помощи IP-пакетов, у данной версии протокола этот пакет делится на два больших поля: поле данных, в котором переносится полезная информация и заголовок, в котором заложен весь функционал протокола, заголовок пакета IPv4 содержит 14 полей, тринадцать из которых обязательные и одно опциональное. Вообще, протокол IPv4 дает нам в распоряжение 2 в 32 степени IP-адресов.

Осознание того, что 4.2 млрд адресов не хватит начало приходить в 90-ых годах, а в 1996 году появился протокол IPv6, который должен когда-нибудь заменить IPv4. Сравним IPv4 и IPv6. Во-первых, IPv6 делает такую технологию, как NAT в текущих условиях бесполезной (для кого-то это плюс, а для кого-то это минус, поскольку NAT не только позволяет «перебивать» много частных IP-адресов в один публичный, но является первой линией защиты вашей компьютерной сети).

Маршрутизация в IPv6 стала быстрее, чем в IPv4 даже несмотря на то, что адрес IPv6 значительно больше, дело все в том, что количество полей в IPv6 стало меньше, хотя сам заголовок оказался несколько длиннее, также немного изменен алгоритм обработки IPv6 пакетов маршрутизатором.

IPv4 использует 32-битные (четырёхбайтные) адреса, ограничивающие адресное пространство 4 294 967 296 (232) возможными уникальными адресами.

Традиционной формой записи IPv4-адреса является запись в виде четырёх десятичных чисел (от 0 до 255), разделённых точками. Через дробь указывается длина маски подсети.

IPv6 (англ. Internet Protocol version 6) — новая версия интернет-протокола (IP), призванная решить проблемы, с которыми столкнулась предыдущая версия (IPv4) при её использовании в Интернете, за счёт целого ряда принципиальных изменений. Протокол был разработан IETF. Длина адреса IPv6 составляет 128 бит, в отличие от адреса IPv4, длина которого равна 32 битам.

Согласно статистике Google на январь 2020 года, доля IPv6 в сетевом трафике составляла около 30 %. В России коммерческое использование операторами связи невелико (не более 4,5 % трафика). DNS-серверы многих российских регистраторов доменов и провайдеров хостинга используют IPv6.

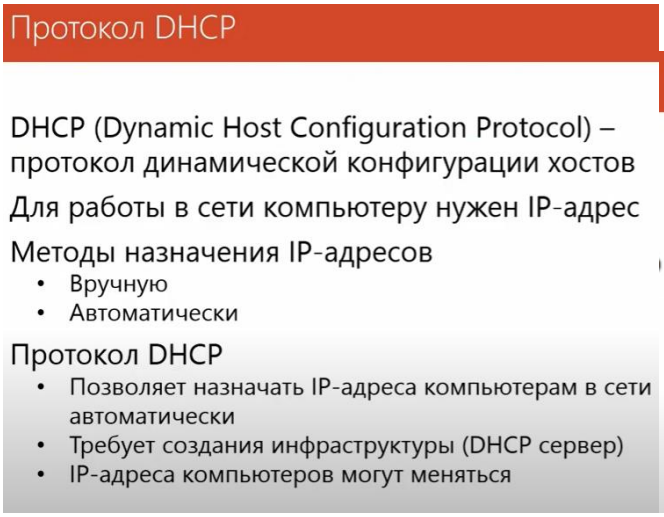
После того, как адресное пространство в IPv4 закончится, два стека протоколов — IPv6 и IPv4 — будут использоваться параллельно, с постепенным увеличением доли трафика IPv6, по сравнению с IPv4. Такая ситуация станет возможной из-за наличия огромного количества устройств, в том числе устаревших, не поддерживающих IPv6 и требующих специального преобразования для работы с устройствами, использующими только IPv6.

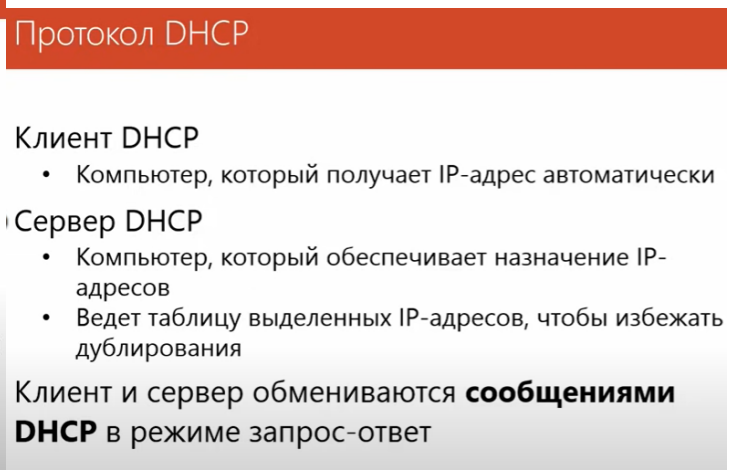
ARP:

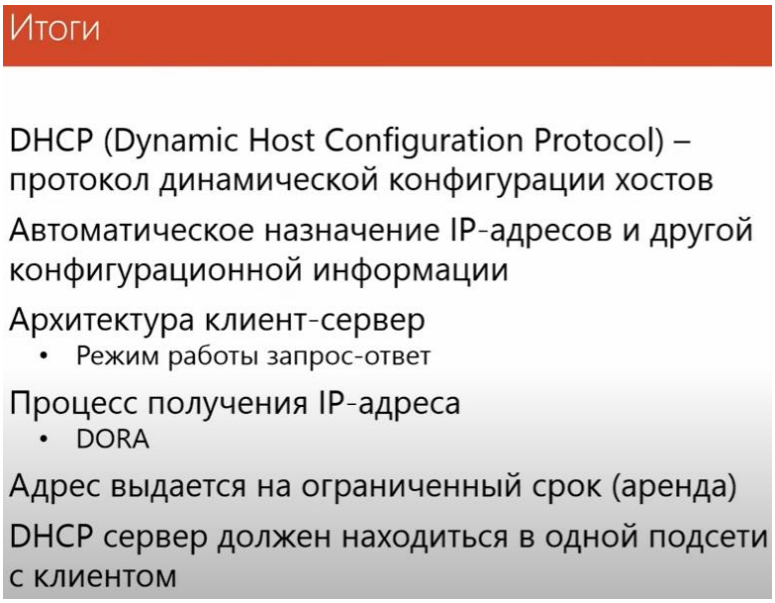
Протокол, используемый для сопоставления адресов сетевого уровня, адресам канального уровня в сетях множественного доступа. ARP описан стандартом RFC 826 в 1982 году. Также является названием сетевого инструмента для сопоставления адресов в большинстве операционных систем. ARP используется для преобразования сетевого адреса (например, IPv4) в физический адрес (например, Ethernet-адрес). Протокол был реализован во многих технологиях сетевого и канального уровней, таких как: IPv4, PUP, Frame Relay и ATM. В IPv6 сетях роль ARP играет NDP (Nework Discovery Protocol).

Представьте компьютеры A и B, расположенные в офисе, объединенные в локальную сеть друг с другом с помощью коммутатора. В данной сети не используются промежуточные шлюзы или роутеры. Компьютер A хочет отправить данные компьютеру B. С помощью DNS он определяет, что компьютеру B соответствует IP-адрес 192.168.0.55. Для передачи данных ему также необходим MAC-адрес компьютера B. Сначала компьютер A использует ARP-таблицу из кэша для поиска MAC-адреса для 192.168.0.55. Если запись с MAC-адресом была найдена, то на MAC-адрес 00:eb:24:b2:05:ac передается IP-пакет, инкапсулированный в фрейм канального уровня модели OSI. Если же запись в кэше не была найдена, то компьютер A отправляет широковещательное ARP-сообщение с запросом информации об IP-адресе 192.168.0.55. В ответ компьютер B отправляет пакет со своим IP и MAC-адресом. После выполнения данной последовательности начинается передача данных.

Протокол DHCP (с.497)







Для нормальной работы сети каждому сетевому интерфейсу компьютера и маршрутизатора, на которые будут приходить и с которых будут отправляться IP-пакеты, должен быть назначен IP-адрес. Назначение IP-адресов может происходить вручную в результате выполнения процедуры конфигурирования интерфейса, которая для компьютера сводится, например, к заполнению системы экранных форм. При этом администратор должен помнить, какие адреса из имеющегося множества он уже использовал для других интерфейсов, а какие — еще свободны. Даже при не очень большом размере сети эта работа представляет для администратора рутинную, а временами и утомительную процедуру. Протокол Dynamic Host Configuration Protocol (DHCP) освобождает администратора от этих проблем, автоматизируя процесс назначения IP-адресов.

DHCP может поддерживать автоматическое динамическое распределение адресов, а также более простые способы ручного и автоматического статического назначения адресов. Протокол DHCP работает в соответствии с моделью клиент-сервер. Во время старта системы компьютер, являющийся DHCP-клиентом, посылает в сеть широковещательный запрос на получение IP-адреса. DHCP-сервер откликается и посылает сообщение-ответ, содержащее IP-адрес и некоторые другие

конфигурационные параметры. Предполагается, что DHCPклиент и DIICP-сервер находятся в одной IP-сети.

При ручной процедуре назначения статических адресов активное участие принимает администратор, который сообщает DHCP-серверу информацию о жестком соответствии IP-адресов физическим адресам или другим идентификаторам клиентов. DHCP-сервер, пользуясь этой информацией, всегда выдает определенному клиенту один и тот же назначенный ему администратором адрес.

При автоматическом статическом способе DHCP-сервер самостоятельно, без вмешательства администратора, выбирает клиенту произвольный IP-адрес из пула наличных IP-адресов. Границы пула задает администратор при конфигурировании DHCP-сервера. Адрес дается клиенту из пула в постоянное пользование, то есть с неограниченным сроком аренды. Между идентификатором клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первого назначения DHCP-сервером IP-адреса клиенту. При всех последующих запросах сервер возвращает клиенту тот же самый IP-адрес.

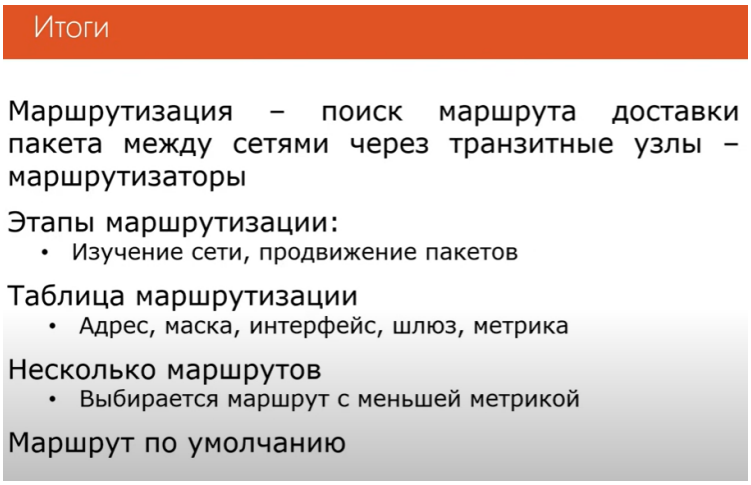
При динамическом распределении адресов DHCP-сервер выдаст адрес клиенту на ограниченное время, называемое временем аренды (lease duration), что дает возможность впоследствии повторно использовать этот IP-адрес для назначения другому компьютеру. Таким образом, помимо основного преимущества DHCP — автоматизации рутинной работы администратора по конфигурированию стека TCP/IP на каждом компьютере, динамическое разделение адресов в принципе позволяет строить IP-сеть, количество узлов в которой превышает количество имеющихся в распоряжении администратора IP-адресов.

DHCP обеспечивает надежный и простой способ конфигурации сети TCP/IP, гарантируя отсутствие дублирования адресов за счет централизованного управления их распределением. Администратор управляет процессом назначения адресов с помощью параметра «продолжительность аренды», который определяет, как долго компьютер может использовать назначенный IP-адрес, перед тем как снова запросить его от DHCP-сервера в аренду.

Рассмотрим работу протокола DHCP в ситуации, когда компьютер, являющийся DHCP-клиентом, удаляется из подсети. При этом назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс. Это свойство очень важно для мобильных пользователей. DHCP-сервер может назначить клиенту не только IP-адрес, но и другие параметры стека TCP/IP, необходимые для его эффективной работы, например, маску, IP-адрес маршрутизатора по умолчанию, IP-адрес сервера DNS, доменное имя компьютера и т. п.

Методы маршрутизации. Функции маршрутизатора

Сетевой уровень



Алгоритмы маршрутизации – КС – Куроуз, Росс, - с.413.

Протоколы маршрутизации – КС – Олифер, - с. 553.

Маршрут - это список узлов коммутации от узла-источника до узла-получателя. Маршрутизация - это набор процедур, позволяющих определить оптимальный маршрут по заданным параметрам на сети связи между парой узлов коммутации.

Маршрутизатор (router) – это устройство сетевого уровня эталонной модели OSI, использующее одну или более метрик для определения оптимального пути передачи сетевого трафика на основе информации сетевого уровня.

Маршрутизатор выбирает наилучший путь, т.е. путь с наименьшей метрикой. То, какой путь лучше определяется количественными показателями. В метрике может учитываться несколько показателей, например длина пути и время прохождения.

Маршрутизатор делит физическую среду передачи данных на части более эффективно, чем мост или коммутатор. Он может пересылать пакеты на конкретный адрес, выбирать лучший путь для прохождения пакета и многое другое. Чем сложнее и больше сеть, тем больше выгода от использования маршрутизаторов.

Маршрутизатор состоит:

- из сетевых адаптеров, зависящих от протоколов и определяющих правила взаимодействия с локальными и глобальными сетями (интерфейсные модули)
- из управляющего процессора определяющего маршрут и обновляющего информацию о топологии
- из основной магистрали

После поступления пакета на интерфейсный модуль анализируется адрес назначения. Сетевой адаптер принимает команды управляющего процессора для определения выходного порта. Затем пакет по основной магистрали передаётся в интерфейсный модуль, служащий для связи с адресуемым сегментом логической или глобальной сети.

Маршрутизаторы делятся на:

1) Устройства верхнего класса – для объеденения сетей предприятия. Данные устройства обладают высокой производительностью, поддерживают множество протоколов и интерфейсов, могут иметь до 50 портов локальных или глобальных сетей.

2) Устройства среднего класса – для менее крупных сетевых объединений масштаба предприятия. Их стандартная конфигурация включает 2-3 порта локальных сетей и от 4 до 8 портов глобальных сетей

3) Устройства нижнего класса – для локальных сетей, подразделений; они связывают небольшие офисы с сетью предприятия.

Функции маршрутизатора:

1) Сбор информации о других маршрутизаторах и хостах в сети. Для этого маршрутизатор в целях определения маршрута использует тот или иной протокол маршрутизации.

2) Сохранение полученной информации о маршрутах в таблице маршрутизации

3) Выбор наилучшего пакета для каждого конкретного пакета при этом осуществляется передача пакета с входного интерфейса на соответствующий выходной интерфейс.

Динамическая маршрутизация. Протокол RIP

Протокол маршрутизации RIP

Динамическая маршрутизация:
- Маршруты в сети определяются автоматически с помощью протоколов маршрутизации

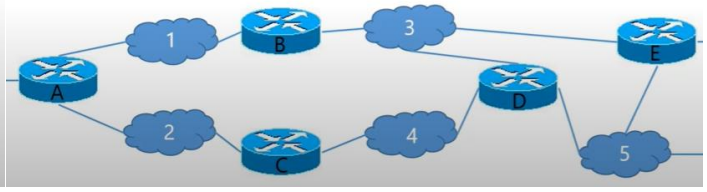
Версии протокола RIP:
- Реализация в BSD UNIX – 1982 г.
- RIPv1 – 1988 г., RFC 1058
- RIPv2 – 1994 г., RFC 2453
- RIPng – 1997 г., RFC 2080

Протокол RIP (Routing Information Protocol):
- Дистанционно-векторный протокол
- Используется алгоритм Беллмана – Форда
- Передача данных через UPD, порт 520

Вектор расстояния

Вектор расстояния (distance vector):
- Адрес сети, расстояние до сети

Расстояние в RIP – количество промежуточных маршрутизаторов
- Максимальное расстояние – 16 (бесконечность)



Работа протокола RIP

Дистанционно-векторный протокол:
- Децентрализованный распределенный алгоритм
- Итерационный расчет стоимости путей при неполной информации о сети

Начальный этап:
- В таблице маршрутизации только подключенные к маршрутизатору сети

Обмен векторами расстояний с соседними маршрутизаторами

Извлечение информации о новых сетях в сообщениях от соседей

Недостатки RIP

Ограничения метрики:
- Учитывается только количество маршрутизаторов, но не скорость каналов

Медленное обнаружение отказов:
- Маршрутизаторы обмениваются сообщениями с векторами расстояний каждые 30 секунд
- Если от маршрутизатора нет сообщений 180 секунд, он считается отказавшим

Протокол маршрутной информации (RIP) — это протокол, позволяющий сетевым устройствам маршрутизации примерно каждые 30 секунд обмениваться имеющимися у них маршрутами. Таким образом каждое устройство маршрутизации добавляет в свою сетевую таблицу маршрутизации новые устройства и маршруты.

Каждое устройство маршрутизации на ссылке маршрутизации называется устройством «транзитной связи»; в таблицах маршрутизации создаются маршруты, имеющие до 15 устройств транзитной связи. Если один пункт назначения имеет более одного маршрута, в таблицу маршрутизации добавляется маршрут с минимальной метрикой (количеством устройств транзитной связи).

В случае недоступности существующего маршрута через 5 минут он отмечается как требующий настройки «бесчисленный» (16 устройств транзитной связи). Затем для нескольких следующих обновлений он транслируется как таковой для других устройств маршрутизации, прежде чем будет удален из таблицы маршрутизации. Система также использует термины «split horizon» (расщепленный горизонт) и «poison reverse» (блокировка сбойного маршрута).

RIP — это простой метод автоматического разделения маршрута и обновления внутри небольших однородных сетей. Он позволяет объявлять альтернативные маршруты в случае неисправности существующего. В большой сети обмен маршрутной информацией каждые 30 секунд может создать избыточный трафик. Кроме того,

таблица маршрутизации, ведущаяся в каждой системе, ограничена 100 маршрутами (включая статические и внутренние).

Протокол RIP является наиболее распространенным протоколом маршрутизации сетей TCP/IP. Несмотря на его простоту, определенную использованием дистанционно-векторного алгоритма, RIP успешно работает в небольших сетях с количеством промежуточных маршрутизаторов не более 15.

□ Для IP имеются две версии протокола RIP: первая и вторая. Протокол RIPvl не поддерживает масок. Протокол RIPv2 передает информацию о масках сетей, поэтому он в большей степени соответствует требованиям сегодняшнего дня.

□ RIP-маршрутизаторы при выборе маршрута обычно используют самую простую метрику — количество промежуточных маршрутизаторов между сетями, то есть хопов.

□ В сетях, использующих RIP и имеющих петлевидные маршруты, могут наблюдаться достаточно длительные периоды нестабильной работы, когда пакеты «зацикливаются» в маршрутных петлях и не доходят до адресатов. Для борьбы с этими явлениями в RlP-маршрутизаторах предусмотрено несколько приемов (Split Horizon, Hold Down, Triggered Updates), которые сокращают в некоторых случаях периоды нестабильности.

□ Дистанционно-векторные алгоритмы хорошо работают только в небольших сетях.

КС – Куроуз, Росс, - с.440.

Протокол OSPF

Протокол маршрутизации OSPF

Протокол OSPF (Open Shortest Path First):
- Протокол с учетом состояния канала (link-state protocol)
- Используется алгоритм Дейкстры (Shortest Path First)
- Передача данных через IP, код протокола 89

Особенности OSPF:
- Децентрализованный глобальный алгоритм
- Расчёт стоимости путей после получения полной информации о сети

Версии протокола OSPF:
- OSPFv1 – 1989 г., RFC 1131
- OSPFv2 – 1998 г., RFC 2328
- OSPFv3 – 2008 г., RFC 5340

Этапы работы OSPF

Изучение топологии сети:
- Маршрутизаторы изучают подключенные сети и ближайших соседей
- Информация о топологии распространяется по всей сети с помощью лавинной рассылки (flooding)

Расчет стоимости маршрутов в сети:
- Выполняется после того, как будет известна полная конфигурация сети
- Каждый маршрутизатор выполняет расчет самостоятельно

Обновление информации о конфигурации сети:
- Маршрутизаторы проверяют доступность соседей
- Рассылка информации об изменении конфигурации сети

Объявления о состоянии канала

Объявление о состоянии канала (link state advertisement):
- Все каналы маршрутизатора
- Состояние каналов
- Доступные сети

База данных состояния каналов (link state database):
- Ведется каждым маршрутизатором
- Содержит информацию о состоянии каналов во всей сети
- Формируется на основе объявлений о состоянии канала
- Базы данных состояния каналов соседних маршрутизаторов в OSPF должны быть синхронизированы

Итоги

Протокол OSPF (Open Shortest Path First):
- Современный протокол маршрутизации в сетях IP
- Стоимость маршрута зависит не только от количества маршрутизаторов
- Расчёт стоимости путей после получения полной информации о сети

Быстрая сходимость:
- Сообщения Hello

Более сложный протокол по сравнению с RIP:
- Высокие требования к памяти и вычислительной мощности маршрутизаторов

□ Протокол OSPF был разработан для эффективной маршрутизации IP-пакетов в больших сетях со сложной топологией, включающей петли. Он основан на алгоритме состояния связей, который обладает высокой устойчивостью к изменениям топологии сети.

□ Периоды нестабильной работы в OSPF-сетях продолжаются недолго, причем пакеты не «зацикливаются» в маршрутных петлях, а просто отбрасываются при невозможности их передать через неработоспособную связь.

□ При выборе маршрута OSPF-маршрутизаторы используют метрику, учитывающую пропускную способность составных сетей.

□ Протокол OSPF является первым протоколом маршрутизации для IP-сетей, который учитывает биты качества обслуживания (пропускная способность, задержка и надежность) в заголовке IP-пакета. Для каждого типа качества обслуживания строится отдельная таблица маршрутизации.

□ Протокол OSPF разрешает хранить в таблице маршрутизации несколько маршрутов к одной сети, если они обладают равными метриками, что дает возможность маршрутизатору работать в режиме баланса загрузки маршрутов.

□ Протокол OSPF обладает высокой вычислительной сложностью, поэтому чаще всего работает на мощных аппаратных маршрутизаторах.

OSPF (Open Shortest Path First), протокол маршрутизации на основе состояния канала, широко применяется в крупных корпоративных сетях. Протокол маршрутизации OSPF собирает информацию о состоянии канала от маршрутизаторов в сети и определяет информацию таблицы маршрутизации для пересылки пакетов. Это происходит путем создания карты топологии сети. В отличие от RIP, OSPF обменивается маршрутной информацией только при изменении топологии сети. Протокол OSPF лучше всего подходит для сложных сетей, состоящих из нескольких подсетей, работающих для упрощения администрирования сети и оптимизации трафика. Когда происходит изменение, он эффективно вычисляет кратчайший путь с минимальным сетевым трафиком.

Плюсы:

- Протокол маршрутизации OSPF полностью знает топологию сети, что позволяет маршрутизаторам рассчитывать маршруты на основе входящих запросов.
- Протокол OSPF не имеет ограничений по количеству переходов, в отличие от протокола RIP, который имеет не более 15 переходов. Таким образом, OSPF сходится быстрее, чем RIP, и обеспечивает лучшую балансировку нагрузки.
- OSPF выполняет многоадресную рассылку обновлений состояния каналов и отправляет обновления только при изменении в сети.

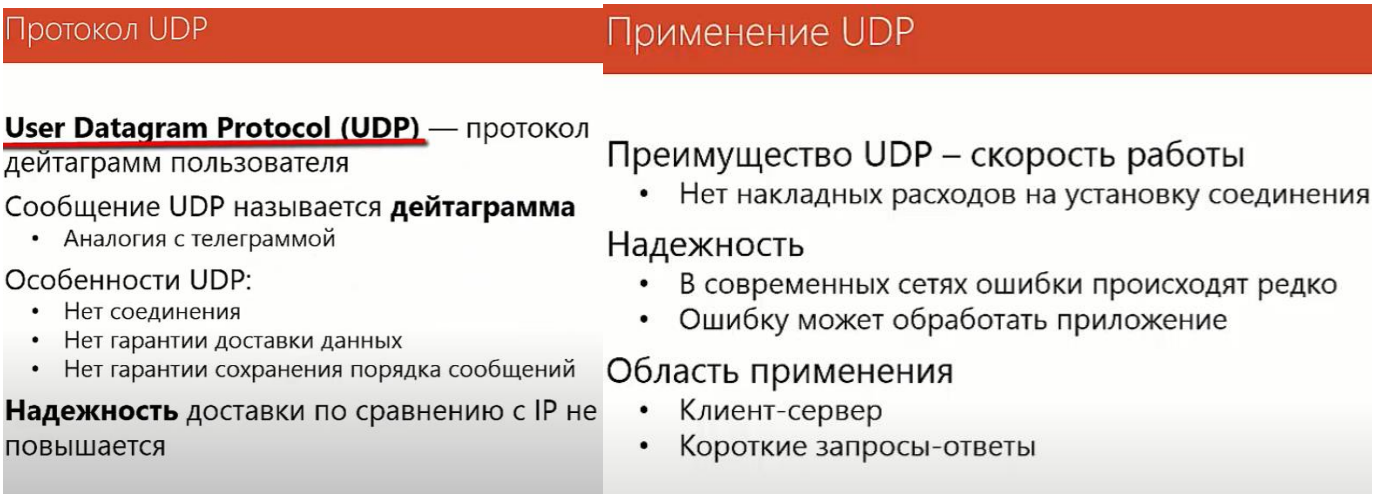
Минусы:

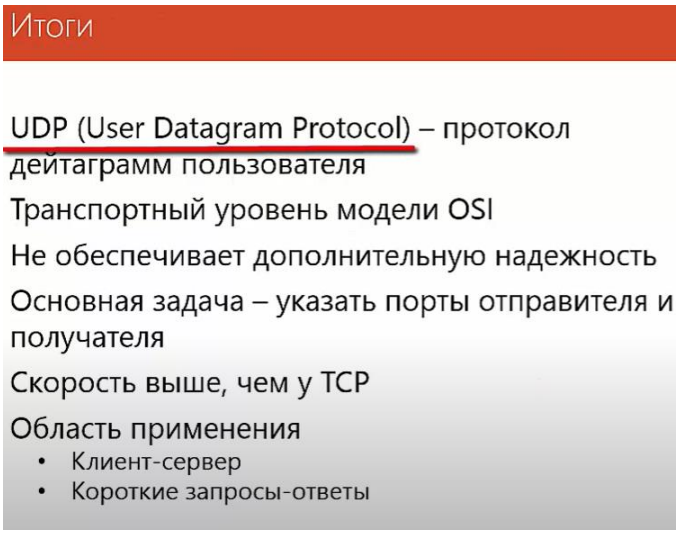
- Протокол OSPF требует глубоких знаний о сложных сетях, что делает его не таким легким для изучения, как некоторые другие протоколы.
- Маршрутизация OSPF не масштабируется при добавлении дополнительных маршрутизаторов в сеть. Отсутствие масштабируемости в протоколе OSPF делает его непригодным для маршрутизации через Интернет.
- Протокол OSPF поддерживает несколько копий маршрутной информации, увеличивая объем необходимой памяти.

КС – Куроуз, Росс, - с.445.

КС – Олифер, - с. 572.

Протокол UDP. Передача данных без установления соединения





КС – Куроуз, Росс, - с.232.

Протокол транспортного уровня без установления соединения.

UDP — User Datagram Protocol

В отличие от TCP UDP — очень быстрый протокол, поскольку в нем определен самый минимальный механизм, необходимый для передачи данных. Конечно, он имеет некоторые недостатки. Сообщения поступают в любом порядке, и то, которое отправлено первым, может быть получено последним. Доставка сообщений UDP вовсе не гарантируется, сообщение может потеряться, и могут быть получены две копии одного и того же сообщения. Последний случай возникает, если для отправки сообщений в один адрес использовать два разных маршрута.

UDP не требует открывать соединение, и данные могут быть отправлены сразу же, как только они подготовлены. UDP не отправляет подтверждающие сообщения, поэтому данные могут быть получены или потеряны. Если при использовании UDP требуется надежная передача данных, ее следует реализовать в протоколе более высокого уровня.

Однонаправленное (unicast) сообщение отправляется из одного узла только в один другой узел. Это также называется связью "точка-точка". Протокол TCP поддерживает

лишь однонаправленную связь. Если серверу нужно с помощью TCP взаимодействовать с несколькими клиентами, каждый клиент должен установить соединение, поскольку сообщения могут отправляться только одиночным узлам.

Широковещательная передача (broadcast) означает, что сообщение отправляется всем узлам сети. Групповая рассылка (multicast) - это промежуточный механизм: сообщения отправляются выбранным группам узлов.

UDP может использоваться для однонаправленной связи, если требуется быстрая передача, например для доставки мультимедийных данных, но главные преимущества UDP касаются широковещательной передачи и групповой рассылки.

Обычно, когда мы отправляем широковещательные или групповые сообщения, не нужно получать подтверждения из каждого узла, поскольку тогда сервер будет наводнен подтверждениями, а загрузка сети возрастет слишком сильно. Примером широковещательной передачи является служба времени. Сервер времени отправляет широковещательное сообщение, содержащее текущее время, и любой хост, если пожелает, может синхронизировать свое время с временем из широковещательного сообщения.

UDP — это быстрый протокол, не гарантирующий доставки. Если требуется поддержание порядка сообщений и надежная доставка, нужно использовать TCP. UDP главным образом предназначен для широковещательной и групповой передачи.

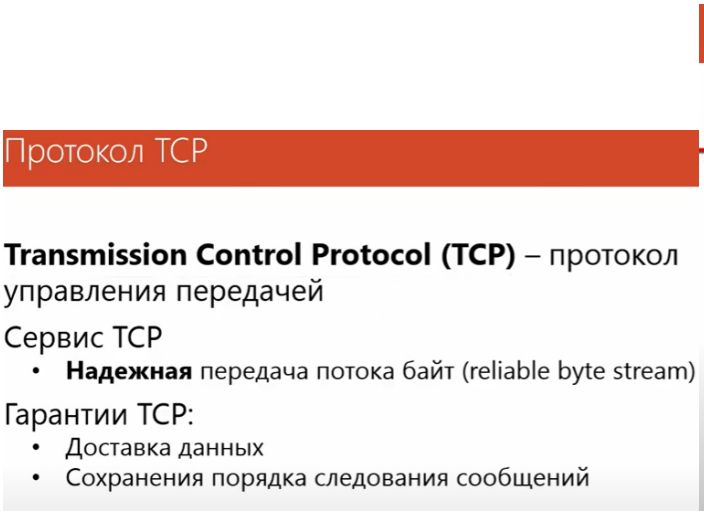
UDP (User Datagram Protocol, Протокол дейтаграмм пользователя) предназначен для обмена дейтаграммами между процессами компьютеров, входящих в единую сеть с коммутацией пакетов. В качестве протокола нижнего уровня UDP-протокол использует IP.

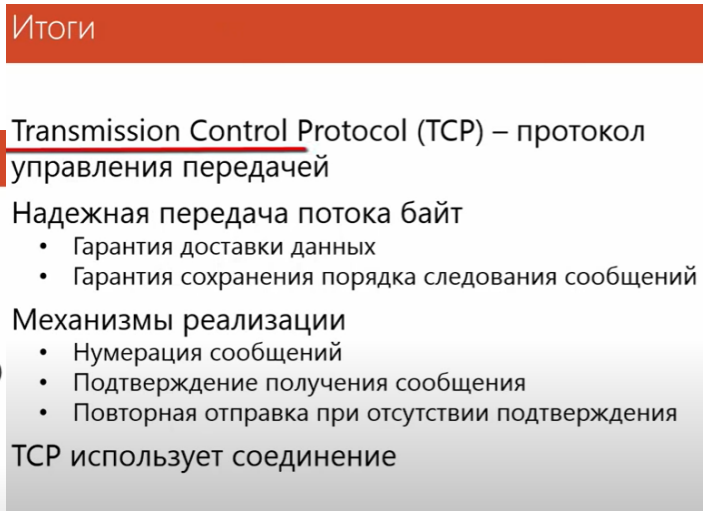
Протокол UDP предоставляет прикладным программам возможность отправлять сообщения другим приложениям, используя минимальное количество параметров протокола. Этот протокол не обеспечивает достоверность доставки пакетов, защиты дублирования данных или надежности от сбоев в передаче. За исключением параметров приложения - номеров портов отправителя и получателя пакета, UDP практически ничего не добавляет к IP-дейтаграмме.

Протокол UDP намного проще, чем TCP и полезен в ситуациях, когда мощные механизмы обеспечения надежности протокола TCP не требуются или будут только помехой для решения определенного рода задач, например, аутентификации пользователей.

Преимущество протокола UDP состоит в том, что он требует минимум установок и параметров для соединения двух процессов между собой.

Протокол TCP. Потоковая передача данных





КС – Куроуз, Росс, - с.269.

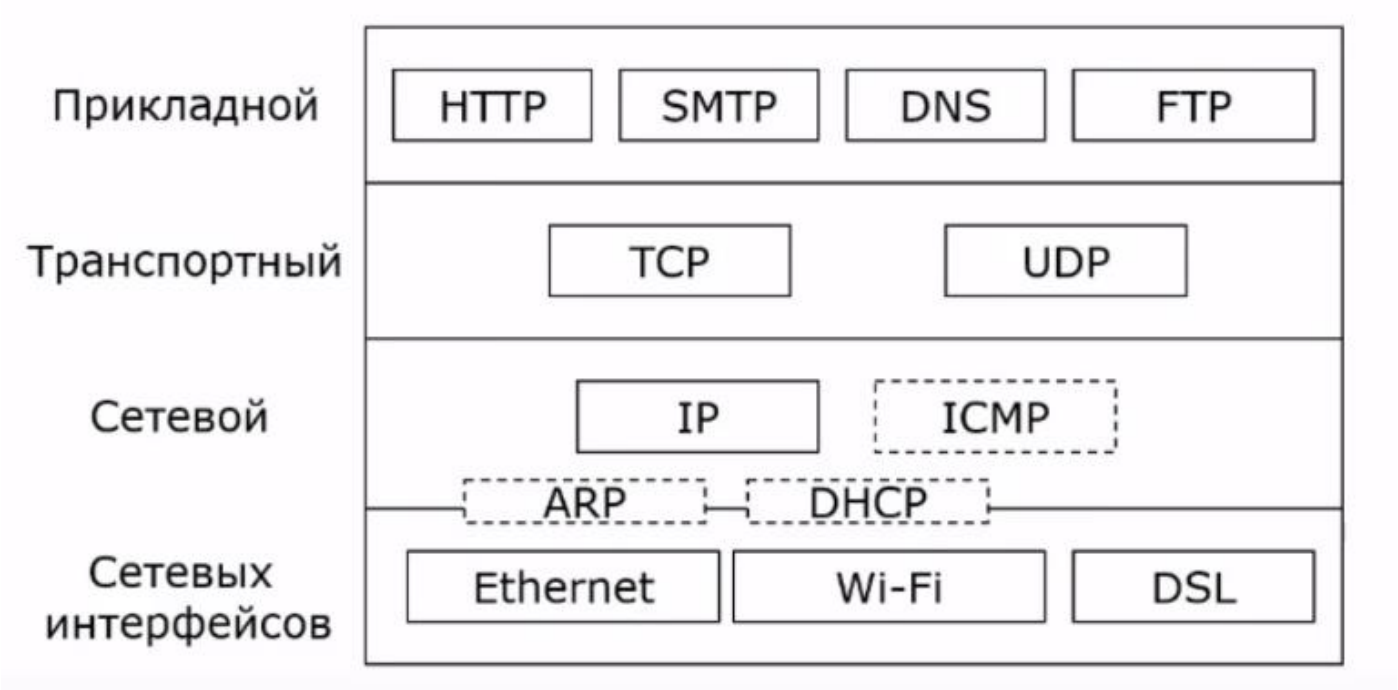
TCP – протокол с установлением логического соединения.

Обмен данными, ориентированный на соединения, может использовать надежную связь, для обеспечения которой протокол уровня 4 посылает подтверждения о получении данных и запрашивает повторную передачу, если данные не получены или искажены. Протокол TCP использует именно такую надежную связь. TCP используется в таких прикладных протоколах, как HTTP, FTP, SMTP и Telnet.

Протокол TCP требует, чтобы перед отправкой сообщения было открыто соединение. Серверное приложение должно выполнить так называемое пассивное открытие (passive open), чтобы создать соединение с известным номером порта, и, вместо того чтобы отправлять вызов в сеть, сервер переходит в ожидание поступления входящих запросов. Клиентское приложение должно выполнить активное открытие (active open), отправив серверному приложению синхронизирующий порядковый номер (SYN), идентифицирующий соединение. Клиентское приложение может использовать динамический номер порта в качестве локального порта.

Сервер должен отправить клиенту подтверждение (ACK) вместе с порядковым номером (SYN) сервера. В свою очередь клиент отвечает ACK, и соединение устанавливается.

После этого может начаться процесс отправки и получения сообщений. При получении сообщения в ответ всегда отправляется сообщение ACK. Если до получения ACK отправителем истекает тайм-аут, сообщение помещается в очередь на повторную передачу.


Прикладной
HTTP
SMTP
DNS
FTP
Транспортный
TCP
UDP
Сетевой
IP
ICMP
ARP
DHCP
Сетевых интерфейсов
Ethernet
Wi-Fi
DSL

Глобальные сети

Глобальные сети - Wide Area Networks (WAN) - объединяют территориально рассредоточенные компьютеры, которые могут находиться в различных городах и странах. Так как прокладка высококачественных линий связи на большие расстояния обходится очень дорого, в глобальных сетях часто используются уже существующие линии связи, изначально предназначенные совсем для других целей. Например, многие глобальные сети строятся на основе телефонных и телеграфных каналов общего назначения. Из-за низких скоростей таких линий связи в глобальных сетях (десятки килобит в секунду) набор предоставляемых услуг обычно ограничивается передачей файлов, преимущественно не в оперативном, а в фоновом режиме, с использованием электронной почты. Для устойчивой передачи дискретных данных по некачественным линиям связи применяются методы и оборудование, существенно отличающиеся от методов и оборудования, характерных для локальных сетей. Как правило, здесь применяются сложные процедуры контроля и восстановления данных, так как наиболее типичный режим передачи данных по территориальному каналу связи связан со значительными искажениями сигналов.