1) Понятие компьютерной сети

Под *компьютерной сетью* (КС) понимают совокупность различных технических средств (то есть самих компьютеров и другого оборудования), предназначенная для передачи компьютерной информации (то есть файлов и сообщений) на относительно большие расстояния (то есть за пределы компьютеров).

Любую КС можно рассматривать с двух точек зрения

1) Программной
2) Аппаратной

В основе любой КС лежит так называемая сеть передачи данных (СПД), которая может задействовать различные среды передачи данных (СрПД). Иногда в составе СПД выделяют базовую (опорную) СПД

Все устройства в составе СПД можно разделить на две четко разделяющиеся группы:

1) Оконечные - находятся по периметру СПД
2) Посредники - составляют ядро СПД

Весь трафик в СПД традиционно разделяют на три базовых типа:

1) Обычные компьютерные данные (data)
2) Голос (voice)
3) Видео (video)

Каждый тип обладает характерными особенностями. СПД, поддерживающие пересылку разнородного трафика, в нотации Cisco называют конвергированными (converged networks)

Особенности трафика обеспечиваются так называемым качеством обслуживания Quality of Service (QoS).

Традиционные виды компьютерных данных без исключения, по умолчанию, обслуживаются по принципу «Все делается для доставки пакетов, но при этом ничего не гарантируется» (best efforts), что, по сути, является отсутствием QoS. Гарантии «возникают» при работе с голосом и видео.

В рамках предоставляемой оборудованием СПД полосы пропускания (bandwidth) можно выделить реально задействованную ее часть (troughput) и полезную составляющую этой задействованной части (goodput) без учета служебного трафика.

2) Классификация компьютерных сетей

С одной стороны, выделяют:

1) Local Area Networks (LANs) - локальные КС (ЛКС)
2) Wide Area Networks (WANs) - глобальные КС (ГКС)
3) Metropolitan Area Networks (MANs) - городские КС

4) Personal Area Networks (PANs) - личные КС
5) Remote Access Services (RASes) - КС для подключения удаленных пользователей (teleworkers)
6) Industrial Networks - промышленные КС
7) Datacenter Networks КС - центров обработки данных

С другой:

1) Intranets - внутренние КС предприятий и организаций
2) Internets - КС публичного доступа

LAN выделяют прежде всего территориально - в современном понимании, охватывает территорию не более кампуса, но при этом подразумевают определенные технологии.

WAN выделяют прежде всего технологически и, в общем случае, может охватывать произвольную территорию.

MAN представляет собой промежуточный вариант между LAN и WAN

PAN позволяет подключить к компьютеру периферийные устройства

RAS существует в контексте WAN

Industrial и Datacenter Networks являются специализированными вариантами LAN

Intranet обычно выделяют по ведомственной принадлежности пользователей.

Практически все Internets сейчас интегрированы в одну сборную одноименную сеть.

Intranet почти всегда имеет связь с Internet

Кроме того, сети могут быть:

1) Изолированными (isolated)
2) Открытыми для прослушивания (open)

С точки зрения организации взаимодействия КС могут быть:

1) Сильносвязанными
2) Слабосвязанными

В случае сильносвязанной КС подразумевают наличие так называемой хост-ЭВМ (host) с одной стороны и терминала (terminal) - с другой. Хост является основным вычислительным компонентом. Под терминалами подразумевают исключительно устройства для ввода и отображения информации, следовательно, они без хоста бесполезны. Совокупность хоста и подключенных к нему терминалов принято называть рабочей станцией (workstation). Терминал администратора, обычно подключаемый особым образом, называют консолью (console). Мы имеем дело с хост-терминальной моделью.

В случае слабосвязанной КС подразумевают наличие сервера (server) с одной стороны и клиента (client) - с другой. Клиентские ЭВМ, обслуживающие запросы пользователей, являются активными компонентами. Сервер либо серверы, являющиеся пассивными

компонентами, в свою очередь, обслуживают запросы клиентов. Как клиенты, так и серверы могут работать независимо, связываясь по мере надобности. Мы имеем дело с клиент-серверной моделью.

3) Стандарты компьютерных сетей

Все стандарты, в том числе в области КС, делят на:

1) Международные (например, ISO/IEC)
2) Европейские (например, EN)
3) Американские (например, ANSI/TIA/EIA)

Стандарты лишь формализуют определенные требования в той или иной предметной области. Стандарты могут носить предварительный или временный характер. Могут включать дополнения и списки обнаруженных ошибок. Могут устаревать или замещаться другими стандартами.

Практическим (или теоретическим) воплощением стандарта является так называемая реализация.

Сертификация позволяет определить факт соответствия стандарту. В 1980 г при IEEE был создан специальный комитет по стандартизации КС, результатом работы которого стало множество стандартов 802.x. Сейчас наибольший интерес представляют:

1) 802.3. – Ethernet
2) 802.11. – Wi-Fi
3) 802.16. – WiMax

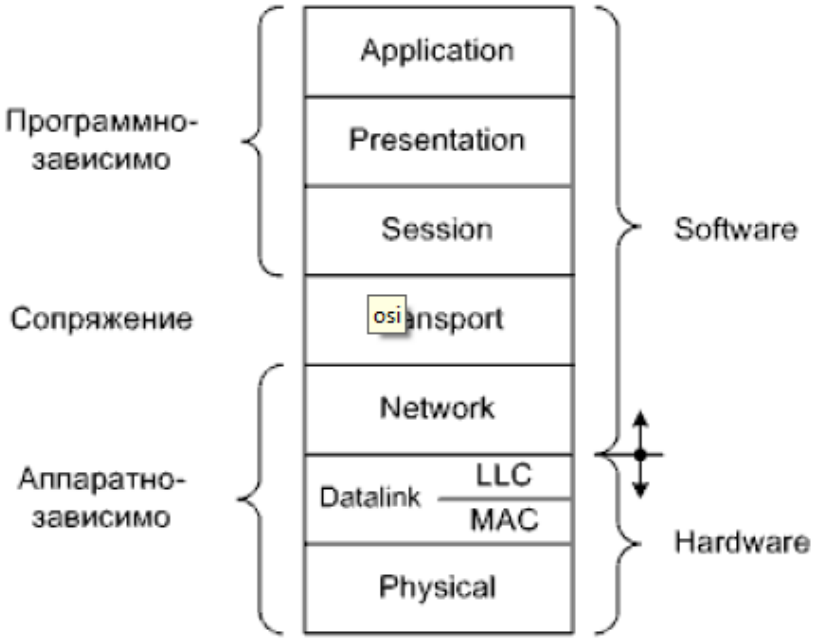
Стандарты Ethernet по пропускной способности делят на три группы:

1) Ethernet - до 10 Mbit/s включительно
2) Fast Ethernet - 100 Mbit/s
3) Gigabit Ethernet – 1, 10, 100, 40, 25, Gbit/s и Multigigabit

4) Наиболее распространенные модели компьютерных сетей

Из всех моделей КС наиболее фундаментальной является открытая модель взаимодействия систем - Open System Interconnection (OSI) разработанная ISO.

Модель включает 7 уровней.



Тут можешь не называть эту схему а сказать что она есть и просто подождать

На вершине иерархии находится человек, но абонентами КС являются взаимодействующие программы.

Взаимодействие в рамках модели OSI может быть «вертикальным» и «горизонтальным»:

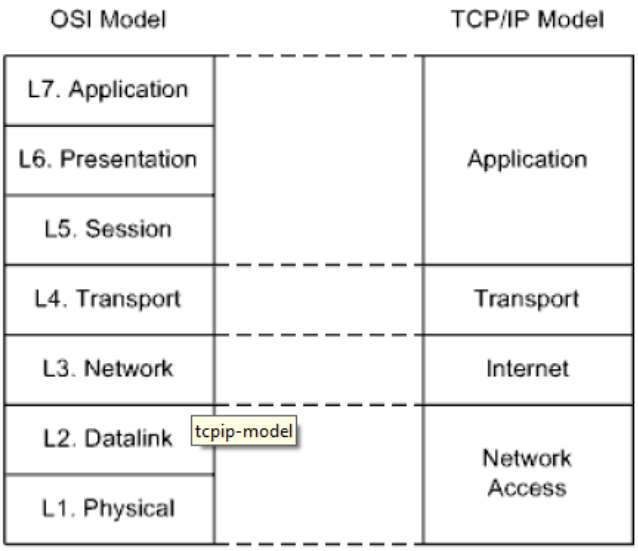
1) Интерфейс - это правила взаимодействия между пространственно совмещенными соседними уровнями модели OSI
2) Протокол - правила взаимодействия между пространственно разнесенными одинаковыми уровнями модели OSI

И в том, и в другом случае предполагают определенную абстракцию.

Названия структурных единиц передаваемой информации в привязке к уровням модели OSI:

L1 -- сигналы (signals).
L2 -- *кадры* (frames).
L3 -- собственно *пакеты* (packets).
L4 + L5 -- *сегменты* (segments).
L6 + L7 -- *сообщения* (messages).

Ещё одна модель связана с семейством протоколов TCP/IP

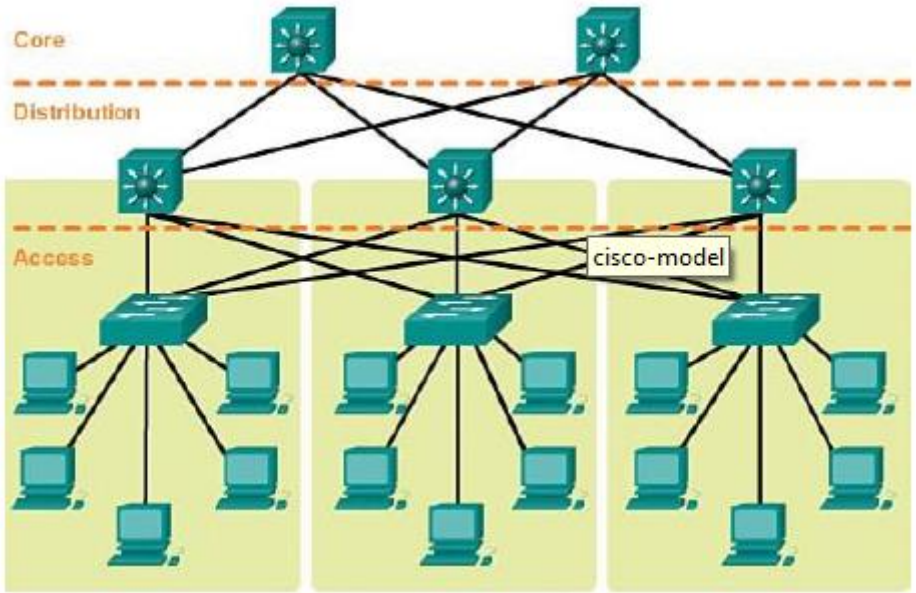


Тут нужно назвать с правого столбика самый нижний сетевой доступ интернет транспортный прикладной

Компания Cisco на основе многолетнего опыта проектирования сетей разработала собственную иерархическую сетевую модель (Cisco hierarhical network model), которую рекомендует использовать в корпоративных сетях разного масштаба.

Модель включает 3 уровня:

1) Access – доступ
2) Distribution (иногда aggregation) – распределение
3) Core – ядро



Этот рисунок не нужен

Уровень доступа предназначен для обеспечения подключений к КС оконечных пользователей. Особое внимание здесь уделяют предоставлению пользователям требующихся им ресурсов. Уровень распределения предназначен для маршрутизации.

Уровень ядра предназначен для обеспечения высокоскоростной связи между относительно удаленными группами пользователей. Особое внимание здесь уделяют характеристикам трафика.В настоящий момент самая популярная из пропагандируемых Cisco архитектур – Cisco Borderless Network.

5) Физический уровень модели OSI

На физическом (physical) уровне формализуют подключение того либо иного сетевого устройства к СрПД. Соответственно в пространстве физический уровень охватывает «точку» подключения.

Основные функции средств, относящихся к данному уровню, является побитовое преобразование цифровых данных в сигналы среды передачи, а также собственно передача сигналов по физической среде.

Специфическими понятиями физического уровня являются:

*-- **среда**; - физическая субстанция по которой происходит передача той или иной информации*

*-- **разъем (физический порт);** - на компьютере или ноутбуке разъём куда можно подключить различные устр-ва, физический порт обменивается электричеством с чем-то, что в этот порт вставлено. В электричестве закодированы какие-то данные.*

*-- **несущая (частота)**; - частота электрич. колебаний, подвергаемых модуляции сигналами с целью передачи информации*

*-- **модуляция**; - изменение несущей частоты таким образом, чтобы она повторяла закономерности сигнала, который мы хотим передать. Это видоизменение называется модуляцией.*

Способы модуляции:

1. *Амплитудная*
2. *Фазная*
3. *Частотная*
4. *Импульсная*

*-- **сигнал**. – код созданный и переданный в пространство.*

Фундаментальная задача физического уровня заключается в передаче сигнала.

6) Канальный уровень модели OSI

На канальном (datalink) уровне формализуют взаимодействие станций в пределах сегмента.

Специфическими понятия канального уровня являются:

1) Сегмент сети
2) Физическая и логическая топология сегмента
3) Пакет(кадр)
4) Бит- и байт-стаффинг
5) Адресация в пределах сегмента
6) Канальный код
7) Код проверки целостности сегмента (кадра)
8) Алгоритм доступа к моноканалу

Каждый из уровней модели OSI может быть реализован достаточно сложно, но канальный уровень особенно сложен. Поэтому его разделяют на два подуровня:

1) MAC (Media Access Control) – контроль доступа к СрПД
2) LLC (Logical Link Control) – контроль логического соединения

На подуровне MAC, более низком, выполняется взаимодействие с физическим уровнем, то есть средозависимые операции, такие как формирование и распознавание пакетов, адресация, канальное кодирование и другие.

На подуровне LLC, более высоком, выполняется взаимодействие с сетевым уровнем, то есть средонезависимые операции, такие как разбиение данных на пакеты, сборка данных из пакетов, определение соответствующей подсистемы сетевого уровня и другие.

7) Сетевой уровень модели OSI (предназначается для определения пути передачи данных.)

Сетевой уровень позволяет «выйти» за пределы сегмента. На сетевом (network) уровне формализуют построение полноценной КС произвольного масштаба, охватывающей произвольное количество сегментов.

3-й уровень сетевой модели OSI, предназначается для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию, отслеживание неполадок и заторов в сети. На этом уровне работает такое сетевое устройство, как маршрутизатор.

Специфическими понятиями сетевого уровня является:

1) Пакет (собственно пакет)
2) Адресация в пределах всей КС
3) Маршрутизация

Можешь мне сказать чтобы я перечислил протоколы и оборудование которое там работает и для чего оно нужно.

8) Транспортный и сеансовый уровни модели OSI

Транспортный уровень позволяет перейти от оборудования к программам.

На транспортном (transport) уровне формализуют использование программным обеспечением сетевого оборудования, то есть как отдельно взятым программам предоставляется «транспорт».

Специфическими понятиями транспортного уровня является:

1) Пакет (сегмент сообщения)
2) Программный порт
3) Логическое соединение
4) Надёжность доставки
5) Алгоритм борьбы с заторами СПД

Сеансовый или сессионный (session) уровень позволяет предоставить доступ к транспорту всем программам в многозадачном окружении.

Кроме собственно сессии, имеются ещё два основных специфических понятий сеансового уровня:

1) Программный порт
2) Алгоритм мультиплексирования программ

В практических реализациях сеансовый уровень выражен слабо и обычно совмещается с транспортным.

Тоже можешь мне сказать перечислить протоколы какие есть
Интерфейс какой есть и суть протоколов

9) Прикладной уровень и уровень представления модели OSI(обеспечивает взаимодеййтсиве пользовательских приложений с сетью, обеспечивает преобразование протоколов и кодирование/декодирование данных)

Прикладной (application) уровень призван решать конкретные пользовательские задачи с помощью КС. Уровень разрешает приложениям пользователя иметь доступ к сетевым службам, таким, как обработчик запросов к базам данных, доступ к файлам, пересылке электронной почты. Также отвечает за передачу служебной информации, предоставляет приложениям информацию об ошибках и формирует запросы к уровню представления

Уровень представления (presentation) позволяет адаптировать прикладную информацию в форму, приемлемую для передачи по КС, то есть

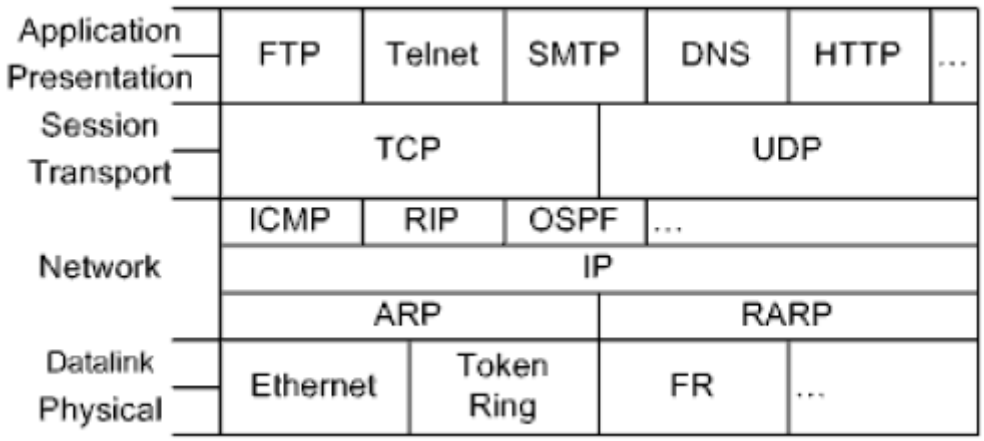
является прослойкой между программами и транспортом. Запросы приложений, полученные с уровня приложений, он преобразует в формат для передачи по сети, а полученные из сети данные преобразует в формат, понятный приложениям. На этом важном уровне может осуществляться сжатие/распаковка или кодирование/декодирование данных, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально.

Основными задачами уровня представления являются:

1) Кодирование информации (включая возможное сжатие) с целью обеспечения её защиты при пересылке по отрытым для прослушивания сетям.
2) Шифрование информации с целью обеспечения её защиты при пересылке по открытым для прослушивания сетям.

Поскольку обычно уровень представления «привязан» к прикладному уровню, в реализациях эти уровни часто совмещаются.

10) Семейство протоколов TCP/IP



Здесь нужно перечислить мне 4 уровня как в 4 билете tcp/ip и назвать то что в ячейках я и сам это помню но так чисто и подождать пока я это распишу

Семейство протоколов TCP/IP описано в стандартах RFC (Request For Comments)

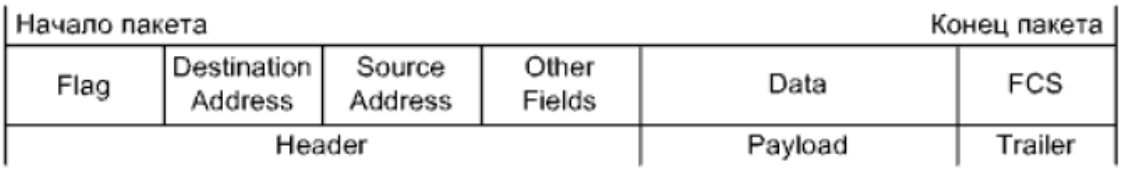
С семейством протоколов TCP/IP связана одноименная модель.

11) Структура типового пакета компьютерной сети

Компьютерные сети имеют последовательную природу т.к. реализовать передачу данных на большие расстояния в параллельном виде гораздо

сложнее чем в последовательном. Внутри данные обрабатываются параллельно.

Для именования порции информации, передаваемой по каналам компьютерных (и не только компьютерных) сетей, используют обобщенный термин пакет (packet). Пакет содержит последовательно сформированные станцией передатчиком поля (fields) предназначенные для их интерпретации в станции приемнике. В общем случае, пакеты могут быть самыми разнообразными (как по структуре, так и по длине), но подавляющее большинство пакетов подпадают под типовую структуру.



Назначение полей:
Flag - флаг начала пакета - позволяет определить начало пакета.
Destination Address - адрес назначения - позволяет указать станцию, для которой предназначен пакет.
Source Address - адрес источника - позволяет указать станцию, сгенерировавшую пакет.
Other Fields - прочие поля - специфические поля (в том числе и специфические флаги) определенной реализации.
Data – данные - «полезное» наполнение пакета
FCS (Frame Check Sequence) - контрольная сумма – позволяет проверить целостность пакета.

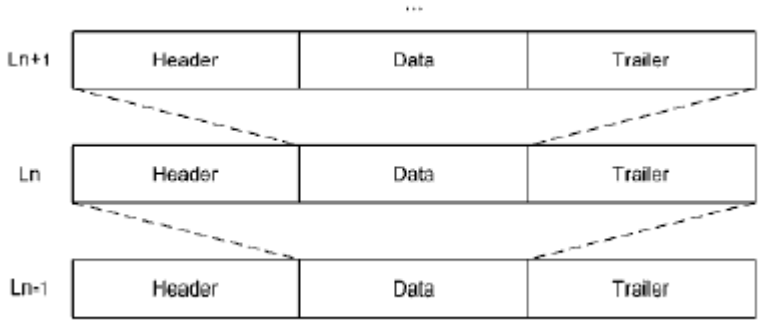
Часть пакета, включающую поля, расположенные до начала данных, принято называть заголовком (header) пакета, после данных – хвостовиком (trailer).

Обычно в байт ориентированных реализациях длина пакета кратна восьми битам, то есть пакет состоит из так называемых октетов.

Все поля в составе любого пакета можно условно разделить на полезные и служебные. Полезная нагрузка заключается в собственно данных. Но следует понимать, что вкладываемая в качестве данных информация может носить служебный характер. В некоторых пакетах поле данных не предусмотрено вообще. Сколько дополнительного трафика порождается в связи с наличием служебных полей оценивают, как overhead.

12) Инкапсуляция и ее проявления в компьютерных сетях

В соответствии с концепцией модели OSI, соседние уровни абстрагированы друг от друга. Поэтому вполне закономерно, что на каждом уровне работают со своими структурами данных. При продвижении информации между уровнями возникает необходимость в преобразованиях структур данных. Преобразования выражаются в инкапсуляции и декапсуляции. Под инкапсуляцией в КС понимают вкладывание пакета определенного вышестоящего уровня в поле данных пакета смежного нижестоящего уровня в процессе подготовки к передаче, то есть при продвижении сверху вниз. Под декапсуляцией понимают обратное действие после приема, то есть при продвижении снизу вверх.



Функционал любого из вышестоящих уровней «знает», какие нижестоящие ресурсы ему необходимы и чем он «располагает». Поэтому процесс инкапсуляции не доставляет трудностей. А вот функционал нижестоящего уровня при разборе полученных пакетов заранее не знает, какой из вышестоящих подсистем передавать эти пакеты. Проблему решают введением в структуру пакета служебного поля, в котором записывается код протокола вышестоящего уровня.

Важной особенностью инкапсуляции является то, что в большинство реализаций заложена возможность передавать пакеты, относящиеся к некоторому протоколу некоторого уровня, вкладывая их в пакеты другого протокола того же уровня, то есть организовывать туннелирование. По-сути это инкапсуляция на том же уровне.

Туннелирование нужно для:

1) Для создания защищённых каналов
2) Для передачи стороннего трафика

Инкапсуляция имеет еще ряд проявлений. Если при выполнении инкапсуляции данные некоторого уровня не помещаются в поле отведенной длины, то можно прибегнуть к фрагментации - разбить данные на фрагменты и передать цепочку пакетов. Принимающая сторона будет вынуждена выполнить дефрагментацию. Поле, отвечающее за длину поля данных, может быть не предусмотрено. Если длина поля данных фиксирована, а данных не хватает, то возникает необходимость в автодополнении (например, нулями).

Перемежение позволяет «распараллелить» пересылку пакетов или их фрагментов и заключается в одновременном задействовании нескольких каналов. Особенно это применимо в низкоскоростных СрПД.

Фрагментация (при наличии альтернативных путей в СПД) и перемежение могут привести к «перемешиванию» пакетов и, как следствие, разрушению сообщения. Контроль за порядком фрагментов может быть возложен как на протокол подверженного фрагментации уровня, так и на протокол вышестоящего уровня.

13) Бит-стаффинг

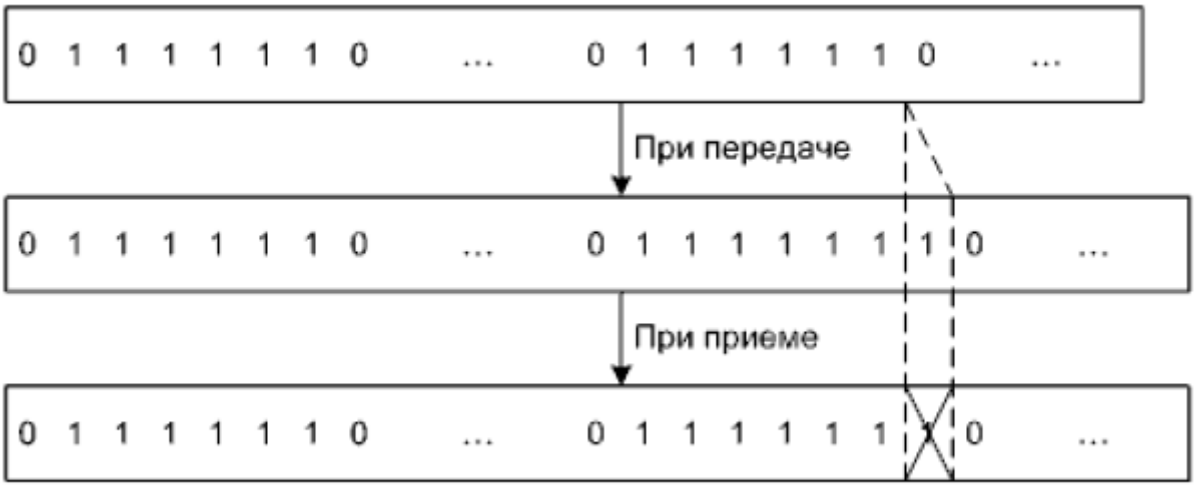
Понятно, что для правильной интерпретации пакета нужно его считать из канала полностью, причем с соблюдением последовательности. Если бы взаимодействующие станции работали бесконечно и находились в соответствующей степени готовности, то это не составляло бы особого труда.

Но, поскольку станция приемник может подключиться к каналу (да и вообще начать работать) в произвольный момент времени, возникает проблема, связанная с распознаванием флага начала пакета. Флаг начала пакета представляет собой зарезервированную цифровую последовательность, которая собственно позволяет станции приемнику определить начало пакета.

Проблема заключается в том, что такая же последовательность вполне может встретиться в пакете и после флага начала. Следовательно, возникает задача обеспечения уникальности флага начала пакета, то есть исключения этой последовательности из оставшейся части пакета. Это достигается за счет действия, заключающегося в модификации следующей за флагом цифровой последовательности, которое в бит ориентированных системах называют бит-стаффингом, а в байт ориентированных байт-стаффингом.

При бит-стаффинге совпадающая с флагом последовательность разбивается с помощью вставки дополнительно бита с соответствующим значением. Применение бит-стаффинга приводит к увеличению длины пакета. Теоретически, с целью уменьшения связанных с бит-стаффингом «издержек», следует стремиться к минимизации количества вставок: разбивающий бит нужно вставлять после наиболее длинной уникальной подпоследовательности в флаговой последовательности.

Классическим флагом начала пакета является байт со значением 01111110 b (7Eh).



На передающей стороне после нуля и шести единиц всегда вставляется седьмая единица, а на принимающей стороне единица после нуля и шести единиц всегда удаляется.

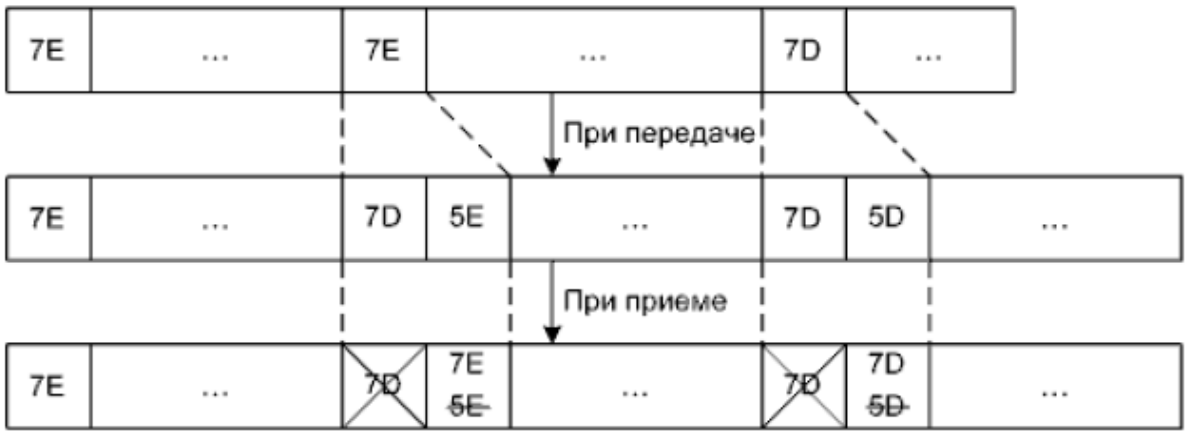
Следует отметить, что на практике бит-стаффинг выполняется вставкой нуля после пяти единиц.

Бит-стаффинг используется при задействовании синхронных СрПД.

14) Байт-стаффинг

Всё начало из бит-стаффинга.

В сравнении с алгоритмами бит стаффинга, алгоритмы байт стаффинга манипулируют байтами, являются более сложными и более «затратными», но при программировании они позволяют избежать битовых операций (бит-стаффинг, в отличие от байт-стаффинга, обычно реализуют аппаратно)



Единственным способом обеспечения уникальности флагового байта является замена совпадающего с ним байта на некий выбранный другой. Но возникает вопрос, как принимающая сторона отличит замененный байт от такого же не заменённого. Решением является применение так называемого ESC символа. Наличие ESC символа говорит станции приемнику о факте замены, а следующий за ESC символом символ код замены позволяет

определить какая замена была осуществлена. Байт-стаффингу можно подвергать целые группы символов.

Байт-стаффинг применяют при задействовании асинхронных СрПД.

15) Особенности линейного кодирования и классификация линейных кодов, применяемых в компьютерных сетях

Одной из основных предпосылок для разработки линейных кодов, является проблема, проявляющаяся во многих системах передачи цифровой (не только) информации, известная как девиацией несущей (carrier deviation)

Очевидно, передатчик и приемник должны работать на одной частоте. В большинстве случаев, передатчик и приемник имеют разные источники синхронизации. При этом тактовые генераторы далеко не идентичны.

Если состояние линии очень долго не изменяется, что происходит при передаче очень длинных нулевых либо единичных последовательностей с использованием классической амплитудной модуляции цифровых цепей (логический ноль соответствует земле, а логическая единица некоторому положительному потенциалу относительно земли), то приемнику «цепляться не за что». В результате накапливаются фазовые сдвиги, что в конце концов приводит к возникновению ошибок. Современная схемотехническая база для борьбы с девиацией несущей, имеет в распоряжении блок ФАПЧ (фазовой автоподстройки частоты), позволяющий автоматически подстраивать тактовый генератор приемника к тактовому генератору передатчика.

Все линейные коды, в той или иной степени, направлены на преобразование битовых последовательностей, чтобы в линии постоянно происходили изменения. В том числе, за счет равномерного распределения нулей и единиц.

Шесть факторов, влияющих на классификацию линейных кодов:

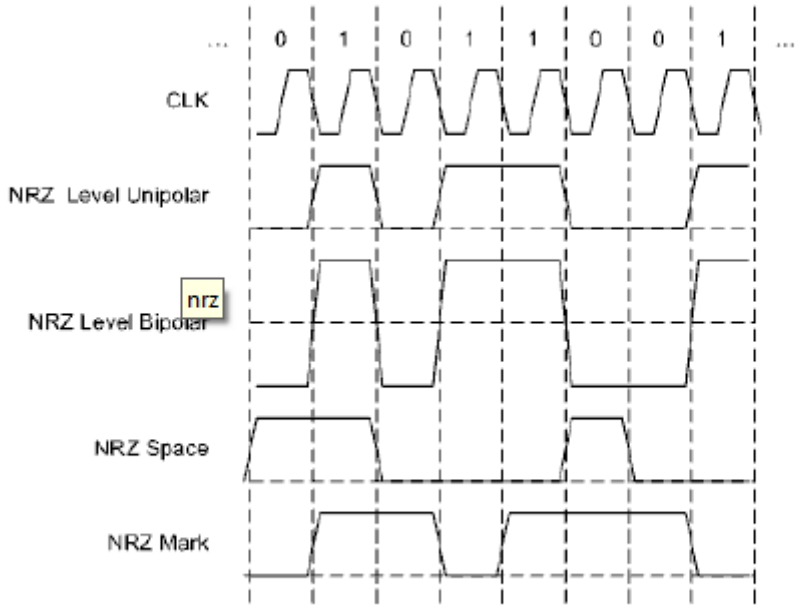
1) Кодирование уровнями либо переходами
2) Наличие инвертирования
3) Однополярность либо многополярность
4) Наличие так называемого “возврата к нулю”
5) Наличие самосинхронизации
6) Наличие перестановки или подмены битов

16) Линейные коды без возврата к нулю и с возвратом к нулю

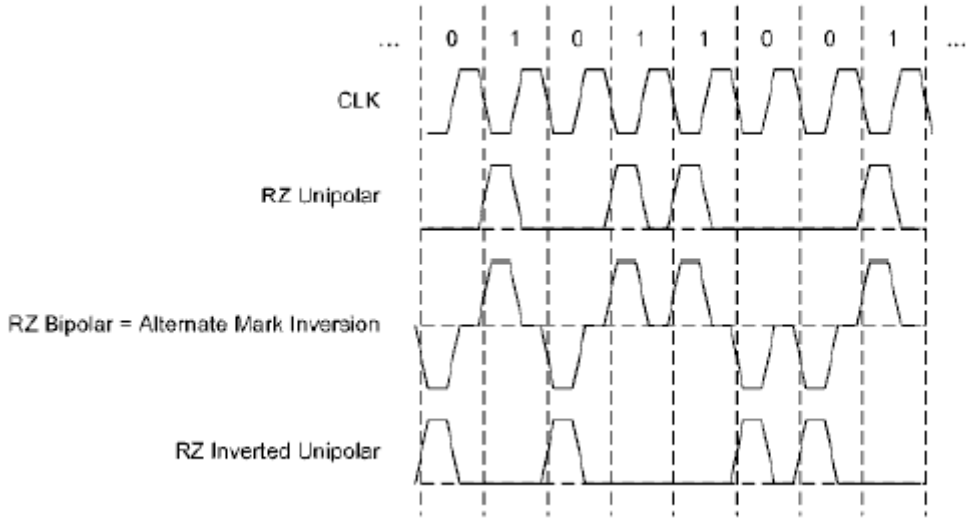
NRZ коды выражаются в изменении уровней между тактами. В простейших случаях, логические уровни в исходной последовательности не преобразуются совсем либо инвертируются. Более сложными случаями являются space и mark. При space-варианте ноль во входной последовательности кодируется сменой текущего уровня в выходной, а

единица сохранением текущего уровня. При mark варианте, наоборот, единицы в исходной последовательности приводят к переключению уровней. Начальное состояние значения не имеет.

Space и mark инверсны друг относительно друга. NRZ коды могут быть однополярными и двухполярными. Требуется наличие дополнительной цепи для тактирования.



RZ коды так же выражаются в изменении уровней между тактами, но на половине каждого такта всегда происходит возврат к нулю (земле). Двухполярные RZ коды обладают свойством самосинхронизации.



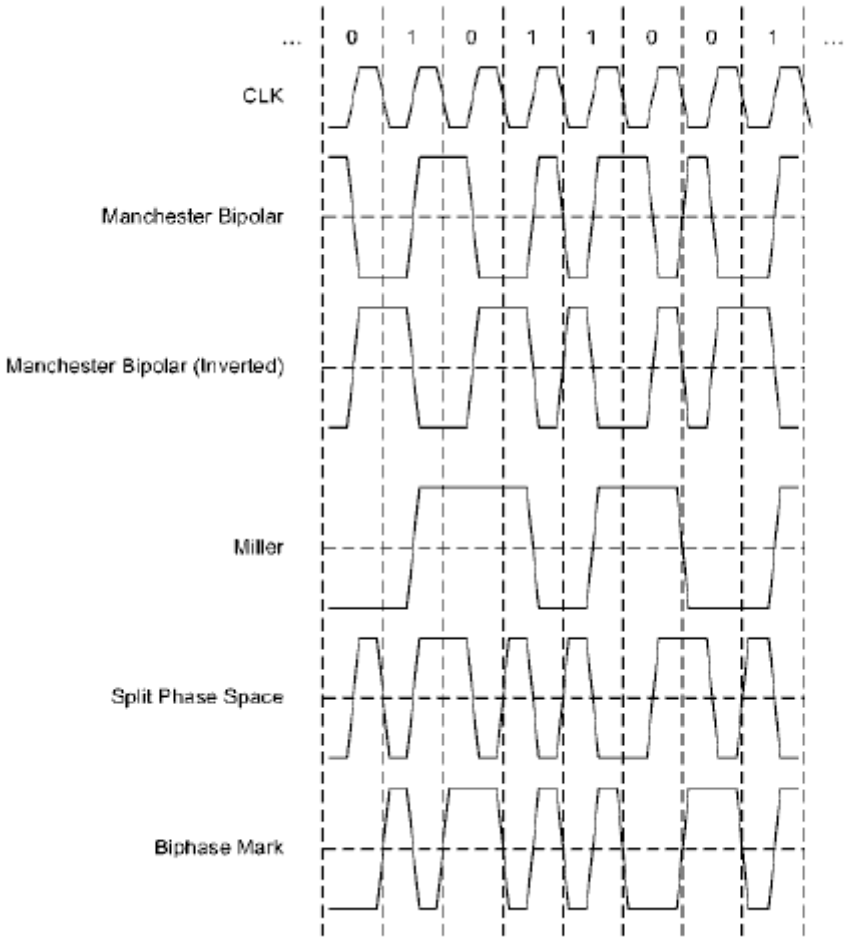
17) Манчестерские и многоуровневые линейные коды

Манчестерские коды выражаются в переходах между уровнями во время тактов, поэтому их иногда называют фазовыми кодами. Есть два

«равноправных» варианта собственно манчестерского кода. 1) Ноль во входной последовательности заменяется на переход от единицы к нулю, а единица заменяется на переход от нуля к единице. 2) Либо наоборот. Манчестерские коды обладают свойством самосинхронизации.

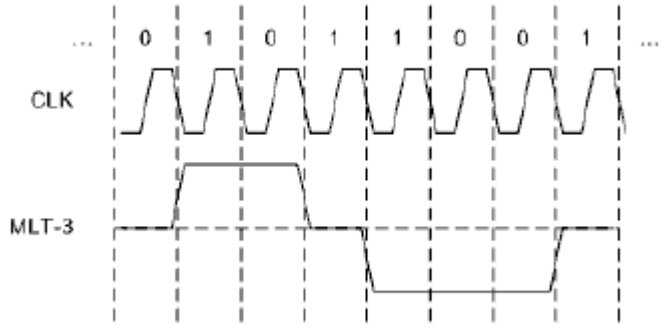
Существуют несколько кодов близких к Манчестерским:

1) Код Миллера
2) Split Phase код
3) Biphase код



Здесь скажешь мне просто нарисовать эту диаграмму где манчест код

MLT коды выражаются в переключении между несколькими уровнями между тактами. Например, код MLT 3 имеет три уровня: -1, 0, +1. Кодирование может начинаться с нуля, ноль в исходной последовательности кодируется сохранением текущего уровня, а единица - переходом к соседнему уровню (с сохранением направления, если это возможно).



А здесь зарисовать эту я их поиню млт

18) Блочные линейные коды

Блочные коды выражаются в замене блоков битов из входной последовательности на бОльшие (как правило) по размеру блоки битов в выходной последовательности. Блочные коды могут комбинироваться с вышеперечисленными кодами. В связи с избыточностью блочных кодов, во многих из них предусмотрены контрольные последовательности, которые, по сути, являются управляющими символами. Первым примером может служить код 4b/5b, применяемый в Fast Ethernet и CDDI.

Более сложным примером может служить код 8b/10b, применяемый в оптических вариантах Gigabit Ethernet. Биты входного блока обозначают как ABCDEFGH от младшего к старшему, выходного abcdefghij так же от младшего к старшему. Входной блок разбивается на два подблока: *x* из пяти битов и *y* из трех битов. Поэтому выходной код представляет собой конкатенацию двух кодов 5b/6b и 3b/4b. Кроме собственно блоков данных D, имеются контрольные блоки K, которые кодируют альтернативно. Таким образом, входной блок обозначают как Dx.y либо Kx.y. Наконец, в код 8b/10b заложена гибкая система уравнивания количества нулей и количества единиц, заключающаяся в динамическом выборе блока для замены (одного из двух) исходя из текущего значения так называемого RD (Running Disparity). Предусмотрено два значение RD -1 и +1. При выборе текущего значения RD учитывается предыдущее значение RD и соотношение нулей и единиц во входном блоке (плюс есть исключения).

19) Поля Галуа и их применение в компьютерных сетях

Поле GF(p) из целых чисел 0, 1, p-1, порожденное в результате отображения f. Z/p->GF(p), где Z/p - факторкольцо множества целых чисел, в котором роль идеала играет простое число p. и f([a]) = a, называют полем Галуа (Galois field) порядка p. При вычислениях с элементами поля Галуа используют целочисленную арифметику с приведением по соответствующему модулю.

Для практического применения полей Галуа в компьютерных системах необходимо перейти от скалярного представления к векторному. Расширенное поле Галуа $GF(p^n)$ можно рассматривать как векторное пространство, где простое число p является характеристикой поля и соответствует количеству состояний разряда вектора, а n является степенью поля над его простым подполем и соответствует количеству разрядов вектора. Поскольку в обычных компьютерных системах разряды регистров бинарные, то наибольший интерес представляют поля $GF(2^n)$

Сложение бинарных векторов (совпадает с вычитанием) проблему не представляет и соответствует поразрядной операции xor. А вот с умножением и делением дела обстоят значительно сложнее. Скалярное произведение не подходит, так как его результат может «выйти» за пределы поля. Векторное произведение определено только для трехразрядных векторов. Полиномиальное представление так же с ходу не решает проблему, так как произведение полиномов опять же «выводит» за пределы поля. Для обеспечения конечности поля Галуа, полученный в результате произведения полином нужно привести. Это достигают путем деления на некий выбранный полином степени n. Ясно, что выбирать можно разные полиномы Выбор другого полинома приведет к другим результатам умножения и, соответственно, к другому полю $GF(p^n)$. Выбранный для построения поля Галуа полином называют порождающим (образующим).

Деление векторов в математике не известно. После перехода на язык полиномов, опять же для обеспечения конечности поля Галуа, деление всегда должно быть безостаточным. Деление можно представить, как умножение полинома, делимого на полином, обратный делителю. При этом для достижения цели на основании математических выкладок, необходимо ввести еще одно ограничение порождающий полином должен быть неприводимым по модулю p (например, если p = 2 и n = 4, то полином x^4+1 (число 17) не подходит, так как $x^4 + 1$ [эквивалентно] $(x^2+1)^2$ (mod 2)). Возведение в степень обладает цикличностью.

20) Модель помехоустойчивого канала связи и теорема Шеннона

Считается, что начало помехоустойчивому кодированию положила теорема Шеннона, утверждающая что любой дискретный канал связи, имеет конечную пропускную способность и этот канал может быть задействован для передачи информации со сколь угодно большой степенью достоверности, не смотря на наличие помех.

Вот эту схему лучше продиктовать я ее забуду

Передаваемое сообщение разбивается на блоки фиксированного размера a из k битов $a_1, a_2, \dots a_k$. Кодер выполняет функцию f называемую схемой кодирования и тем самым преобразует вектор, a в вектор c из n k битов $c_1, c_2, \dots c_n$ называемый кодовым словом. В процессе пересылки кодового слова по каналу связи на него накладывается вектор ошибок e в котором единичные биты соответствуют искажениям. После применения декодером схемы декодирования g получается вектор a' в идеале совпадающий с исходным вектором a.

Подобная схема кодирования является избыточной. На практике всегда ищут компромисс между степенью обеспечения достоверности при передаче и вычислительной сложностью кодов (что в первую очередь отражается на скорости декодирования). В КС множество кодовых слов получается из множества исходных слов как отображение из конечного поля $GF(2^k)$ в конечное поле $GF(2^n)$. При более простых схемах кодирования, в кодовом слове сначала располагаются биты входного сообщения, называемые информационными, а за ними дополнительные биты, называемые проверочными $a_1, a_2, \dots a_k$, битов $c_1, c_2, \dots c_n$. В более сложных случаях проверочные биты чередуются с информационными.
Схему кодирования удобно представлять в матричном виде:

Схема кодирования:

$f: GF(2^3) \to GF(2^6) = a_1,a_2,a_3 \to a_1,a_2,a_3,c_4,c_5,c_6$

Проверочные уравнения:

$$c_4 = a_1 \wedge a_2$$
$$c_5 = a_2 \wedge a_3$$
$$c_6 = a_1 \wedge a_3$$

Переход к матричному представлению:

$$\begin{bmatrix} c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Проверочные уравнения по-другому:

$$a_1 \wedge a_2 \wedge c_4 = 0$$
$$a_2 \wedge a_3 \wedge c_5 = 0$$
$$a_1 \wedge a_3 \wedge c_6 = 0$$

В матричном виде (^{T} означает транспонирование):

$$H c^T = 0$$

Видно, что проверочные уравнения образуют систему линейных уравнений. Следовательно, отображение f (схема кодирования) является линейным.

21) Линейные помехоустойчивые коды, включая коды Хэмминга и циклические коды

Перед выбором того либо иного помехоустойчивого кода всегда нужно определиться, что требуется от кода Если перефразировать, то нужно ответить на два вопроса:

1) Сколько бинарных ошибок код должен обнаруживать.
2) Сколько бинарных ошибок код должен исправлять.

Исправлять ошибки значительно сложнее, чем обнаруживать. Применительно ко многим кодам, исправление ошибки подразумевает нахождение ее позиции.

В общем случае ошибки носят случайный характер. Множественные ошибки могут быть взаимозависимыми, то есть образовывать модули ошибок. Если ошибки расположены рядом, то они образуют пакет ошибок (частный случай модуля).

Число координат (позиций), которыми два вектора x и y различаются называют расстоянием Хэмминга – *d(x,y)*. Число ненулевых позиций вектора x называют весом Хэмминга *w(x)*. Видно, что расстояние Хэмминга показывает количество возникших ошибок.

Для увеличения корректирующей способности кода следует стремиться увеличивать расстояния между кодовыми словами. При этом минимальное

расстояние d_{min} называют кодовым и оно является очень важной характеристикой помехоустойчивого кода. Согласно теореме, для того чтобы линейный код исправлял t ошибок должно выполняться условие: $d_{min}>=2t + 1$.

Для того, чтобы линейный код обнаруживал t ошибок должно выполняться условие $d_{min}>=t + 1$.

Способность того или иного кода сохранять свои характеристики зависит и от количественного соотношения информационных и проверочных символов. В теории помехоустойчивого кодирования определяют так называемые верхние и нижние границы кодов.

Бинарным кодом Хэмминга называют код длины $n = 2m – 1$, $m >=2$ с проверочной матрицей H размером m × 2 m 1 в которой столбцы соответствуют записи 1, 2 … 2m-1 в двоичной системе счисления. Код Хэмминга позволяет исправлять одиночную ошибку и обнаруживать множественные ошибки.

Циклические коды являются особо выделяемой подгруппой линейных кодов. Циклическим кодом называют линейный код, удовлетворяющий дополнительному условию: если вектор a_0, a_1 … a_{n-1} является кодовым словом, то и его циклический сдвиг a_{n-1}, a_0 … a_{n-2} так же является кодовым словом. Циклический код позволяет исправлять одну и более ошибок и обнаруживать множественные ошибки (зависит от параметров).

Базовая идея циклического кодирования состоит в том, чтобы в качестве проверочных битов передавать остаток от деления информационных битов на некоторое выбранное число. После приема снова выполняется деление уже возможно искаженных информационных битов на то же самое число и сравниваются остатки. Если остатки совпадают, то данные с определенной вероятностью приняты без ошибок.

На практике же деление выполняется по правилам арифметики полей Галуа, то есть без учета переносов. Информационные биты, то есть делимое, соответствуют информационному полиному. Делитель соответствует порождающему (образующему) полиному. Частное в процессе кодирования не используется и поэтому «отбрасывается». Для того чтобы максимально разнообразить остатки в качестве порождающего полинома должен выбираться неприводимый полином.

Существуют два подхода к реализации циклического кода на стороне приемника:

1) Согласно базовой идее, описанной выше
2) На порождающий полином делится всё принятое кодовое слово. Если ошибка не прошло, то остаток будет нулевым. Оба подхода равноценны.

22) Классификация помехоустойчивых кодов

Основные группы помехоустойчивых кодов:

1) Линейные коды, в том числе: коды Хэмминга, циклические коды, БЧХ-коды (коды Боуза-Чоудхури-Хоквингема), РМ коды (коды Рида-Маллера), итеративные коды, коды на основе матриц Адамара, симплексные коды и некоторые другие.

2) Коды для контроля модульных и пакетных ошибок, в том числе РС-коды (коды Рида-Соломона), низкоплотные модульные коды, векторные модульные коды, итеративные модульные коды и некоторые другие.

3) Свёрточные коды

4) Арифметические коды

5) Низкоскоростные коды, в том числе: коды максимальной длины, нелинейные коды, D-коды и некоторые другие.

Если что, взять что-нибудь с 21)

23) Классификация каналов в сети передачи данных

С точки зрения направленности, последовательный канал может функционировать в одном из трех режимов:

1) Симплексном – передача данных по каналу возможна только в одном направлении.
2) Полудуплексном – данные могут передаваться в обоих направлениях, но в один момент времени возможна передача только в одном направлении.
3) Полнодуплексном – данные могут передаваться в обоих направлениях одновременно.

Сейчас в КС доминируют полнодуплексные каналы.

Последовательный канал может быть:

1) Выделенным – зарезервирован за определённой парой станций-абонентов.
2) Разделяемым – может использоваться несколькими станциями-абонентами.

Причем канал, который не может разделяться несколькими станциями передатчиками одновременно, в отечественной литературе принято называть моноканалом. Во многих реализациях ситуация именно такая.

24) Логические и физические топологии LAN

Топология «возникает» на канальном уровне, когда речь идет об организации сегмента.

Прежде всего, топологии делят на два типа:

1) Point-to-point - топология «точка к точке» связывает только две станции

2) Multi-access (multipoint to multipoint) - топология с множественным доступом - связывает более двух станций

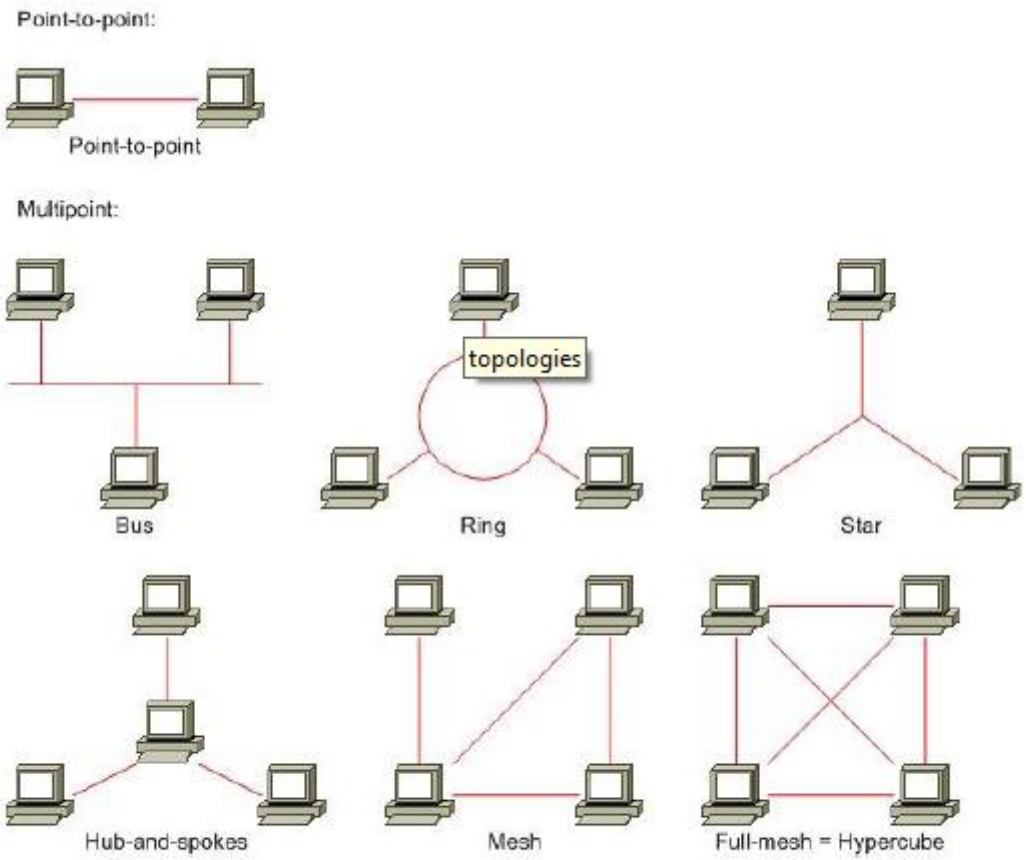
Эти два типа позволяют организовывать двунаправленные каналы между любым требующимся количеством абонентов поэтому их реализуют наиболее часто.

Применительно к однонаправленным каналам можно добавить еще два пункта:

3) Point-to-multipoint – иногда

4) Multipoint-to-point - очень редко

Менее двух станций в сегменте быть не может.



Топологии КС с детализацией до станций и СрПД

В общем случае, направленность каналов может «накладываться» на топологии по разному. Например, кольцо может быть однонаправленным и

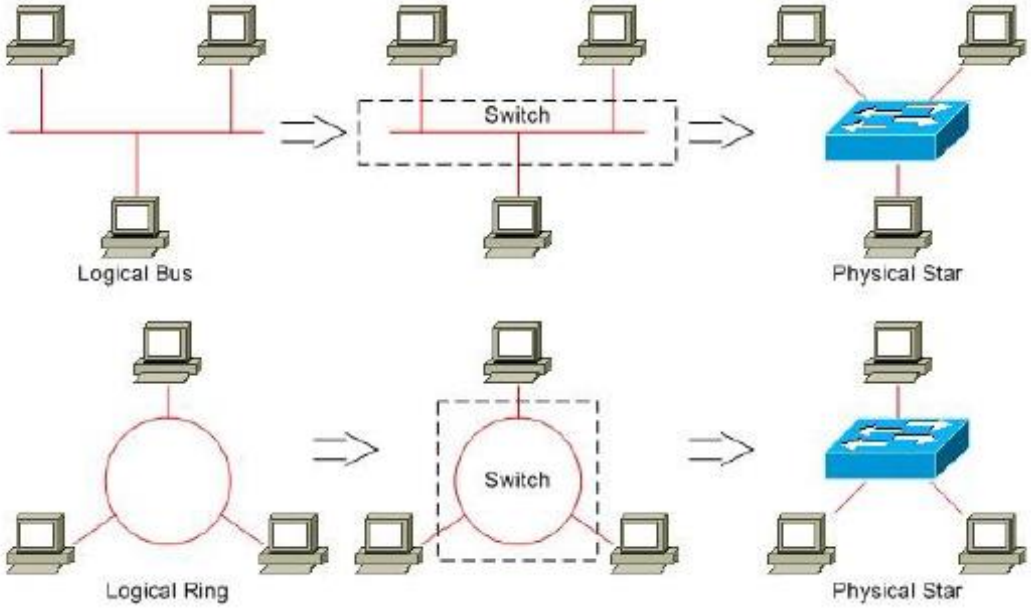
двунаправленным. Сегмент может иметь и гибридную топологию (hybrid topology)

Если топологически классифицировать аппаратные технологии (охватывающие два нижних уровня модели OSI), то есть еще два ракурса:

1) Физическая топология – отражает физические связи между устройствами.
2) Логическая топология – отраает лигические связи между устройствами.

Часто логическая топология не совпадает с физической.

Примеры соответствийй между физ. и логич. топологиями:



Характерными топологиями ЛКС являются:

1) Шина
2) Кольцо
3) Звезда

Скажи мне их зарисовать

Сегменты соединяют произвольным образом, поэтому на сетевом уровне уместно говорить о топологии с произвольными связями, хотя топологию в отношении третьего уровня упоминают весьма редко. Протоколы сетевого уровня обычно разрабатывают топологически независимыми.

25) Логические и физические топологии WAN и RAS

Та же вода про топологии что и в 24

Характерными топологиями ГКС являются:

1) Сеть (произвольно связанная) (mesh)
2) Ступица со спицами (hub-and-spoke)

3) Полносвязная сеть (full-mesh)

Скажи тоже зарисовать

Характерной RAS топологией является point-to-point. Можно сказать, что для ГКС-технологий существует только одна типичная топология (произвольно связанная сеть), остальные можно рассматривать как ее частные случаи.

Для RAS технологий существует только одна типичная топология. На начальных этапах изучения, Cisco не отделяет RAS от ГКС.

26) Особенности случайных методов доступа к моноканалу

В первую очередь затрагиваются передатчики, то есть активные компоненты системы. Проблема заключается в «столкновениях» конкурирующих передатчиков. Пассивные по своей природе приемники априори конфликтовать не могут. Хотя количество приемников всегда ограничивается, так как передатчики имеют конечную нагрузочную способность. Если находящиеся в равных условиях два либо более передатчиков одновременно выдают сигналы в СрПД (устанавливают соответствующие уровни напряжения), то возникает противоречие. Таковое единовременно неразрешимое противоречие принято называть коллизией.

Коллизия может быть как логической (информационный конфликт) так и физической (несовместимые физические процессы). Обычно коллизия возникает при попытках установить противоположные логические уровни.

Классическим способом защиты оборудования от коллизий является так называемая гальваническая развязка (трансформаторная либо оптронная). При попытках установить разные уровни, как правило, наблюдаются эффекты «зануления» и «заединичивания» в зависимости от особенностей элементной базы.

Ситуация с коллизией может затрагивать только станции, подключенные к одной СрПД, то есть сегмент компьютерной сети. Сегмент, в котором возможно возникновение коллизий называется доменом коллизий. Понятие коллизии относится не только к сигналу, а и к пакету.

Чтобы передатчик мог бороться с коллизиями он безусловно должен иметь возможность определять факт их наличия. Борьба с коллизиями, по определению, актуальна применительно к многоточечным топологиям.

Физические свойства СрПД не позволяют мгновенно передавать сигналы. Следовательно и возникшая коллизия распространяется по сегменту конечной скоростью. Под окном коллизий (collision window) понимается временной интервал, в течение которого любая из станций гарантирован н о обнаруживает коллизию, равный удвоенному времени прохождения сигнала между двумя максимально удаленными станциями. Без учета окна коллизий,

влияющего на время постудержания сигнала, невозможно спроектировать работоспособный сегмент.

Существуют два основных подхода к проблеме коллизий:

1) Не допускать коллизии вообще, то есть пользоваться детерминированными методами доступа к моноканалу.
2) Допускать коллизии и каким-то образом выходить из них, что достижимо только использованием случайных методов доступа к моноканалу

Во втором случае так же можно выделить два подхода:

1) Не обращать внимание на причины возникновения коллизий, а упор делать на способ выхода из них
2) Пытаться предотвращать коллизии тем самым максимально снижая их количество, ну а если коллизии все таки возникают, то «тяжело» выходить из них

Таким образом, все методы доступа к моноканалу делят на:

1) Случайные
2) Детерминированные

Все случайные методы основаны на использовании генератора случайных чисел (поэтому их так и называют который позволяет делать случайные задержки при доступе к моноканалу, а значит и с определенной степенью вероятности избегать коллизии.

На эффективность случайных методов наиболее существенное влияние оказывают следующие факторы:

1) Кол-во взаимодействующих станций
2) Инертность среды передачи данных
3) Длина кадра
4) Частота синхронизации

27) CSMA/CD (Ethernet)

С точки зрения изучения случайных методов доступа к моноканалу наиболее наглядным примером является классический алгоритм CSMA/CD (Carrier Sense Multiple Access with Collision Detection) множественный доступ с прослушиванием несущей и обнаружением коллизий, описанный в стандарте Ethernet (IEEE 802.3).

Задержка перед началом очередной попытки передачи после коллизии измеряется в так называемых слот таймах, количество которых является случайным целым числом r

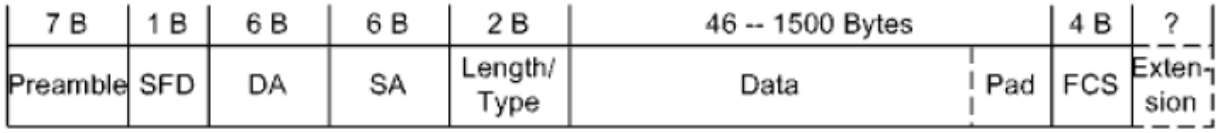
$0 <= r <= 2^k$, где k = min (n, 10), где n номер попытки.

После превышения счетчиком попыток некоторого порогового значения дальнейшие попытки считаются бесперспективными.Значение k не может быть больше 10.

Качество диспетчеризации при обработке коллизий по большому счету зависит от одного базового параметра. Слот-тайм (slot time) является минимальной неделимой единицей времени при диспетчеризации и подбирается с учетом многих других параметров. По крайней мере, он должен быть больше суммы удвоенного времени прохождения сигнала по сегменту и времени передачи jam-сигнала.

В стандарт заложен механизм ускорения распределенного обнаружения коллизий, заключающийся в их «усилении». Каждая обнаружившая коллизию станция передает специальный jam-сигнал некоторой длительности (значение стандартом не регламентируется). Jam-сигнал выполняет две важные функции. Во-первых, является признаком возникновения коллизии, что позволяет другим станциям сразу «увидеть» коллизию (столкнувшиеся передатчики, выставившие jam сигнал, и так знают о коллизии). Во-вторых, позволяет синхронизировать время начала отсчетов случайных задержек.

28) Кадр Ethernet



Поля:

1) Preamble – преамбула
2) SFD (Start Frame Delimiter) – разграничитель начала кадра
3) DA (Destination Address) – адрес назначения
4) SA (Source Address) – адрес источника
5) Length/Type – длина либо тип
6) Data – данные
7) Pad – наполнитель
8) FCS (Frame Check Sequence) – контрольная сумма
9) Extension – расширитель

Предусмотрены полудуплексный и полнодуплексный режимы, «поведение» в которых несколько различается. В качестве преамбулы выступают семь байтов со значением 10101010b, а в качестве SFD байт со значением 10101011b. При сборке кадра учитываются ограничения на его длину. Ограничивается не только максимальная длина, а и минимальная. При недостаче в поле данных вслед за ним в кадр вставляются дополнительные октеты наполнители (значения стандартом не регламентируются). Параметр MTU (Maximum Transmission Unit) определяет максимальный размер вкладываемых данных. Применительно к Ethernet, если значение поля Length/Type больше либо равно 1536 (600h), то указывает тип

инкапсулируемых данных. При необходимости, октеты расширители дополняет кадр до тайм-слота (только в полудуплексном режиме).

Ethernet-заголовок имеет фиксированную длину. Но, поскольку многие базирующиеся на Ethernet технологии имеют собственные подзаголовки, заголовок, а, следовательно, и весь кадр, может увеличиться, правда незначительно и не затрагивая MTU. Некоторые технологии предусматривают значительное увеличение кадра уже за счет увеличения MTU. Наконец, многие производители оборудования Ethernet предусмотрели нестандартное (но в большинстве случаев совместимое) административное увеличение MTU вплоть до 9000 байтов в первую очередь, для оптимизации пересылки больших объемов данных. Такие Ethernet кадры называют гигантскими.

В качестве контрольного кода используется код CRC

При функционировании с пропускной способностью выше 100 Mbit/s (только в полудуплексном режиме) реализация может опционально передавать серию кадров ослабив контроль среды. Такой режим работы называется пакетным режимом. Сразу после успешной передачи первого кадра начинается безусловная передача последующих кадров — это возможно, поскольку передатчики других станций по-прежнему будут находиться в состоянии ожидания. Интервалы между кадрами, без которых принимающая станция вообще не сможет различать кадры, укорочены до минимума с помощью октетов расширителей. Количество кадров в пакете ограничивается. Считается, что в правильно сконфигурированном сегменте при передаче второго и последующих кадров пакета коллизии возникать не должны. Однако, если такая коллизия возникает, то она обрабатывается особо (выход из алгоритма с ошибкой) - это так называемая поздняя коллизия.

29) CSMA/CA (Wi-Fi)

Еще одним примером случайных методов доступа к моноканалу является гораздо более сложный алгоритм CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) множественный доступ с прослушиванием несущей и избеганием коллизий, описанный в стандарте Wi Fi (IEEE 802.11.)

В настоящее время реализации Wi Fi на физическом уровне очень разнообразны используются до десяти различных способов модуляции. Более того, для Wi Fi характерно создание большого числа параллельных каналов.

Стандартом предусмотрены целых шесть вариантов отслеживаемых межкадровых интервалов IFSes (InterFrame Spaces). Отслеживание различных IFSes в различных ситуациях влияет на способность станции «видеть щели» между кадрами, а значит и на способность «вклиниваться» в пересылку.

Случайная задержка измеряется в слот таймах, как и в Ethernet, но алгоритм другой Количество слот таймов является случайным целым числом Random:

$0 \le Random \le CW$,

где CW (contention window) -- так называемое окно состязаний:

$CW_{min} \le CW \le CW_{max}$,

и берется из ряда: 7, 15, 31 ... (два в некоторой степени минус один).

Крайние значения зависят от способа модуляции (типичное значение CW_{min} -- 15, типичное значение CW_{max} -- 1023).

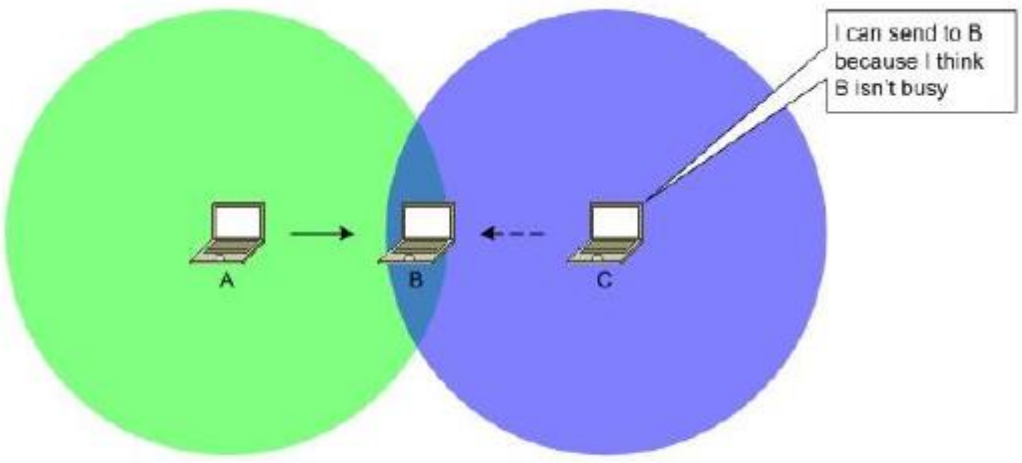
Предусмотрены два счетчика попыток SRC (Short Retry Count) и LRC (Long Retry Count). Количество попыток ограничивается Выбор значения зависит от физического уровня.

Для беспроводных каналов свойственны две проблемы, которые получили следующие названия:

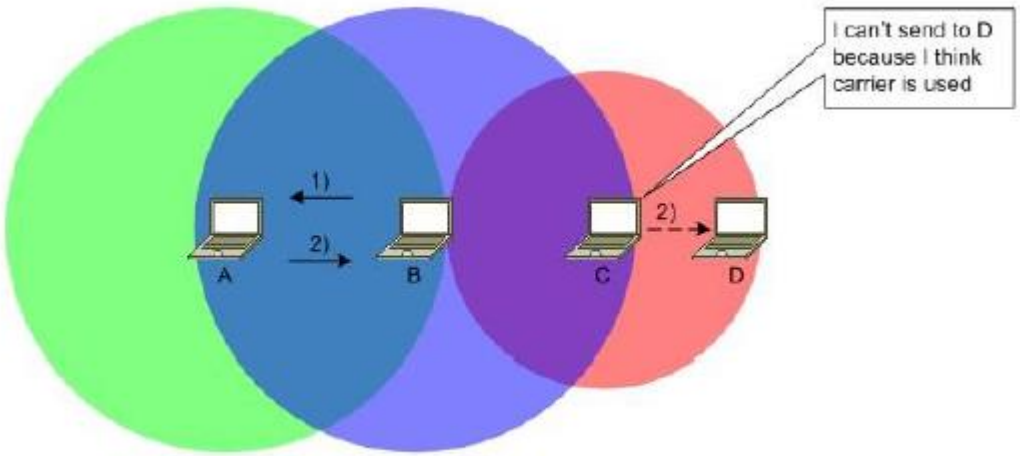
1) Hidden node problem - проблема скрытой станции
2) Exposed node problem - проблема доступной станции

Предполагается, что все станции взаимодействуют в рамках одного канала. (Эти проблемы возникнут и в проводных каналах, если не учесть окно коллизий)

Проблему скрытой станции можно сформулировать так станция С может ошибочно начать передачу станции В, так как не может «услышать» что станция А уже передает станции В (станция А «скрыта» от станции С)



Проблему доступной станции можно сформулировать так станция С, зная о взаимодействии станций А и В, не может передать станции D во время пассивности станции В, а могла бы, поскольку считает канал занятым ошибочно (станция С «доступна» для станции D)



Частично решить проблемы помогает опциональное расширение RTS/CTS.

В связи с особенностями беспроводных каналов, в них передатчикам значительно сложнее самостоятельно обнаруживать коллизии. Поэтому эта функция с них снимается и возлагается на приемники. Вместо обнаружения коллизии, передатчик ждет положительное подтверждение ACK от приемника. Коллизия, как и любая другая проблема с кадром, приведет к отсутствию подтверждения и, далее, к повторной передаче.

В рамках CSMA/CA существуют две группы алгоритмов:

1) Без наличия станции-координатора и с упреждающим jam-сигналом
2) С наличием станции-координатора

30) Кадры Wi-Fi

Формат кадра Wi Fi так же сло'жен в сравнении с форматом кадра Ethernet. При этом наличие и названия последующих полей зависит от значения предыдущих.

2 Bytes	2 B	6 B	6 B	6 B	2 B	6 B	2 B	4 B	0 -- 7951 B	4 B
Frame Control	Duration /ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control	HT Control	Data	FCS

Header

2 bits	2 b	4 b	1 b	1 b	1 b	1 b	1 b	1 b	1 b	1 b
Protocol Version	Type	Subtype	To DS	From DS	More Fragments	Retry	Power Management	More Data	Protected Frame	Order

Поля:

1. Frame Control -- контроль кадра.
2. Duration/ID -- длительность-идентификатор (0 -- 32767 us при резервировании канала, трактовка зависит например от наличия QoS).
3. Address 1 -- адрес 1.
4. Address 2 -- адрес 2.
5. Address 3 -- адрес 3.
6. Sequence Control -- контроль последовательности.
7. Address 4 -- адрес 4.
8. QoS Control -- контроль QoS.
9. HT Control (High Throughput) -- контроль интенсивной пересылки (при QoS).
10. Frame Body -- содержимое кадра (данные).
11. FCS (Frame Control Sequence) -- контрольная сумма.

Поля контроля кадра:

1) Protocol Version – версия протокола (до сих пор равна нулю)
2) Type – тип: 00 – Management управление, 01– Control контроль, 10 – Data данные, 11 – Reserved зарезервировано.
3) Subtype – подтип (в настоящее время определено около сорока подтипов)
4) To DS – флаг направления в распределительную систему (проводную систему, связывающую беспроводные сегменты)
5) From DS – флаг направления из распределительной системы
6) More Fragments – флаг наличия фрагментации
7) Retry флаг – повторной попытки передачи
8) Power Management – флаг режима энергосбережения
9) More Data – флаг наличия дополнительных данных (например, буферизированных данных для находящейся в режим энергосбережения станции)
10) Protected Frame – флаг защищенности кадра (шифрование)
11) Order – флаг упорядоченности (при QoS)

Таким образом, существуют три типа кадров.

В зависимости от подтипа кадра в адресных полях могут комбинироваться до четырех из пяти возможных адресов:

1) BSSID (Basic Service Set Identifier) – идентификатор так называемой базовой зоны обслуживания (то есть беспроводного сегмента),
2) SA (Source Address) – адрес источника,
3) DA (Destination Address) – адрес назначения,
4) TA (Transmitting station A ddress) – адрес станции передатчика (непосредственного)
5) RA (Receiving station Address) – адрес станции приемника (непосредственного)

31) Особенности детерминированных методов доступа к моноканалу

Если случайные методы уместно использовать при шинной топологии, применительно к которой четко выражена возможность возникновения коллизий, то детерминированные методы хорошо «ложатся» на кольцевую топологию. Концептуальная разница между случайными и детерминированными методами заключается в том, возникает ли случайность при «обращении» станции к моноканалу.

Если при некотором такте кольца какая-либо из станций имеет собственный кадр для передачи и при этом получила из кольца еще один кадр, который необходимо «продвигать» дальше, то появляется вопрос о том, какой из этих кадров передавать.

Единственным способом преодоления логических коллизий является введение приоритетов. В то время как все случайные методы «завязаны» на генератор случайных задержек, все детерминированные методы «завязаны» на систему приоритетов в том или ином виде. Возникает задача распределенного либо централизованного назначения приоритетов, причем ни одна из станций кольца заранее ничего «не знает» о других станциях.

При использовании механизма приоритетов не обойтись без так или иначе выраженного арбитра. В качестве арбитра может выступать специальный служебный кадр, который в русскоязычной литературе обычно называют маркером (token).

Таким образом, основные критерии классификации детерминированных методов:

1) Централизованное либо распределённое управление
2) Алгоритм назначения приоритетов
3) Топологические особенности

На эффективность детерминированных методов наиболее существенное влияние оказывают те же факторы, что и в ситуациях со случайными методами:

1) Количество взаимодействующих станций
2) Частота синхронизации
3) Длина кадра

Если сравнивать детерминированные методы со случайными, то сложно сказать какие из них «лучше» При применении случайных методов основные потери производительности возникают из-за вносимых задержек, а при применении детерминированных методов потери обусловлены ретрансляцией кадров. Если оценивать реализации, которые уже имеются на рынке, то все же детерминированные алгоритмы в среднем демонстрируют

бОльшую производительность. Однако оборудование в среднем более дорогостоящее.

32) Алгоритм Token Ring

В Token Ring применяется централизованное управление. Закономерным следствием является необходимость включения в кольцо по крайней мере одной управляющей станции, наделенной особыми полномочиями и призванной инициализировать кольцо и следить за его работоспособностью. В терминологии Token Ring так ую управляющ ую станци ю обобщенно называ ют станцией монитором (monitor station). Кроме единственной основной станции монитора (active monitor) в состав кольца может входить некоторое количество резервных (standby monitors). Функции станции монитора:

1) Инициализировать подключившиеся к кольцу станции
2) Тактировать (на физическом уровне) работу кольца
3) Контролировать наличие и валидность маркера
4) Предотвращать зацикливания

В отличие от сегмента Ethernet, где все станции равноправны и действуют по одному и тому же алгоритму, в сегменте Token Ring предусмотрены станции несколько видов. Наряду с выделяемыми на канальном уровне станциями мониторами, на более высоких уровнях рекомендуется выделять следующие станции:

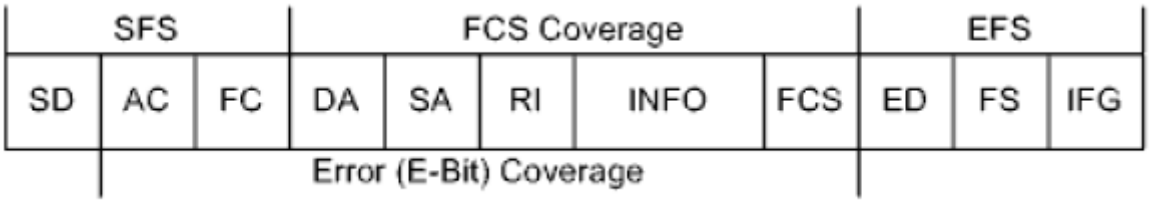
1) System managers - системные менеджеры (на них сосредоточены управляющие системой на основе Token Ring процессы)
2) Servers - различные серверы (configuration report servers, ring error monitors, ring parameter servers)
3) Data stations - информационные станции (обычные пользовательские станции)

Для того чтобы понять заложенный в стандарт алгоритм, сначала необходимо рассмотреть форматы кадров Token Ring и назначение основных полей. В стандарте предусмотрены четыре вида передаваемых

Последовательностей:

1) Token – маркер
2) Frame – кадр
3) Abort Sequence - прерывающая последовательность
4) Fill - заполняющая последовательность

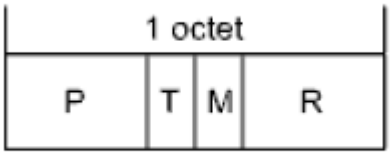
Каждая из станций в любое время должна распознавать (и различать) маркеры, кадры и специальные последовательности.



Поля:

1) SD (Starting Delimiter) - начальный разделитель
2) AC (Access Control) - контроль доступа
3) FC (Frame Control) - контроль кадра
4) DA (Destination Address) - адрес назначения
5) SA (Source Address) - адрес источника
6) RI (Routing Information) - информация о маршрутизации (может отсутствовать)
7) INFO (information) - данные (могут отсутствовать)
8) FCS (Frame Check Sequence) - контрольная сумма
9) ED (Ending Delimiter) - конечный разделитель
10) FS (Frame Status) - состояние кадра
11) IFG (InterFrame Gap) - межкадровый интервал

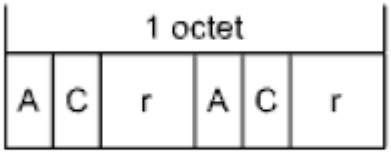
С точки зрения алгоритма контроля доступа наибольший интерес представляет одноименное поле, а также поле состояния кадра



Где:

1) P (Priority bits) - текущий уровень приоритета
2) T (Token bit) - идентификатор маркера 0 маркер, 1 кадр
3) M (Monitor bit) - бит монитора
4) R (Reservation bits) - запрашиваемый уровень приоритета

Формат поля состояния кадра:



Где:

1) A (Address-recognized bit) - флаг распознания адреса (дублируется)

2) C (frame-Copied bit) - флаг копирования кадра (дублируется)
3) R (reserved) – зарезервировано

Механизм приоритетов Token Ring основывается на связке двух полей P и R. Поле P отображает текущий уровень приоритета, а поле R запрашиваемый. Каждое из этих полей может иметь значение от 000b до 111b, то есть доступно восемь уровней приоритета.

Условно можно выделить два режима взаимодействия:

1) Все станции имеют одинаковые приоритеты («отсутствие» приоритетов)
2) Станции могут иметь разные приоритеты («наличие» приоритетов, совместимое расширение первого режима, некоторые станции могут пользоваться кольцом более интенсивно связь с QoS)

При «отсутствии» приоритетов станция монитор создает и «запускает» в кольцо маркер с нулевыми полями P и R (назначение этих полей не проявляется)

С помощью маркера, который передается по цепочке от станции к станции, предоставляется право на передачу. Если у станции нет своего кадра для передачи, то она передает маркер дальше. Если у станции есть кадр для передачи, то она захватывает маркер, заменой значения поля T с нуля на единицу преобразует маркер в кадр, добавляет все соответствующие поля и передает. Приоритет автоматически «достается» станции, до которой маркер дошел раньше. Внесенный таким образом в кольцо кадр ретранслируется всеми промежуточными станциями до тех пор, пока не достигнет адресованной станции-абонента.

За удаление кадра из кольца ответственна станция, создавшая его. Поэтому станция абонент, распознавшая свой адрес в принятом кадре, вместо удаления кадра отмечает факт распознавания присваиванием единичных значений обоим битам A и передает кадр дальше. Если станция абонент «забирает» данные из кадра, то она присваивает единичные значения и обоим битам C. Значения битов A и C проверяются при возвращении кадра к создавшей его станции. На основании результатов проверки делаются соответствующие выводы. Но нужно освободить маркер. В нормальном случае станция освобождает маркер сразу после того, как дождется возвращения кадра.

Существует опциональная возможность освободить маркер более быстро. При раннем освобождении маркера (early token release) сразу вслед за кадром передается новый маркер, а старый маркер не воссоздается. В результате, несколько кадров смогут находиться в кольце одновременно (максимальное количество кадров будет равно максимальному количеству станций), но маркер всегда будет только один. За счет того, что разные такты кольца «накладываются» друг на друга, потенциально можно получить

значительный временной выигрыш. Станции, не использующие и использующие раннее освобождение маркера, могут сосуществовать.

С точки зрения отдельно взятой станции порядок доступа к кольцу можно свести к трем шагам:

1) Захват маркера и передача кадра
2) Освобождение маркера и при необходимости коррекция текущего уровня приоритета
3) Восстановление текущего уровня приоритета если он был скорректирован

В качестве контрольного кода используется код CRC. Скорость Token Ring равна 4 либо 16 Mbit/s (100 Mbit/s самые поздние реализации).

33) Реализации детерминированных методов доступа к моноканалу

Кроме Token Ring следует упомянуть еще ряд существующих технологий реализаций детерминированных методов

1) Технология ARCNET (Attached Resource Computer NETwork) была первой технологией ЛКС, нашедшей массовое применение до экспансии Ethernet, в том числе благодаря своей дешевизне. Стандарт ATA 878.1 был разработан и утвержден ARCNET Trade Association. В настоящее время является сильно устаревшей.
Скорость: 2,5 Mbit/s.
Логическая топология: однонаправленное кольцо.
Физическая топология: шина или звезда.
Во втором случае требовалось дополнительное сетевое оборудование (пассивные или активные концентраторы). Алгоритм являлся аналогом упрощенного варианта алгоритма Token Ring (без системы приоритетов).
2) Технология Token Bus разрабатывалась параллельно с Token Ring. Была стандартизирована как IEEE 802.4. Благодаря плохому масштабированию и сложности восстановления после сбоев, почти не применялась, только в промышленных сетях некоторых индустриальных компаний. Разработка давно остановлена, является сильно устаревшей
Скорость: 1, 5, 10, 20 Mbit/s
Логическая топология: однонаправленное кольцо
Физическая топология: шина
Алгоритм представлял собой адаптацию алгоритма Token Ring к шинной топологии.
3) Технология FDDI (Fiber Distributed Data Interface) разрабатывалась целенаправленно для поддержки оптических СрПД и позволяет значительно увеличить дальность передачи. Кроме собственно FDDI, еще был разработан аналогичный вариант для электрических СрПД под

названием CDDI (Copper Distributed Data Interface). FDDI формализовали в виде комплекса стандартов, которые разрабатывались постепенно в основном ANSI и ISO. Ключевыми являются стандарты ISO 9314-1, ISO 9314-2 и ISO 9314-3. FDDI стал быстро вытесняться с рынка сетевых технологий после появления более дешевого Fast Ethernet, но ограниченно применяется до сих пор CDDI распространения так и не получил.
Скорость: 100 Mbit/s, 200 Mbit/s
Логическая топология: однонаправленное кольцо с резервированием, то есть два отдельных кольца (если оба кольца исправны, то они функционируют параллельно)
Физическая топология: двойное кольцо, к которому с помощью дополнительного сетевого оборудования могут подключаться деревья (узлами дерева являются концентраторы, листьями - станции, концентратор корень включается в двойное кольцо)
Алгоритм представляет собой расширение алгоритма Token Bus.
4) Технология 100VG-AnyLAN была разработана HP и стала альтернативой Fast Ethernet. Идея заключалась в получении по тем временам высокоскоростного гибрида между Ethernet и Token Ring, причем с сохранением совместимости с их кадрами. Позже была стандартизована как IEEE 802.12 На технологию возлагались большие надежды, но она была быстро отвергнута рынком и в скорости практически исчезла.
Скорость: 100 Mbit/s
Логическая топология: дерево
Физическая топология: дерево (с опциональным резервированием), формируемое с помощью дополнительного сетевого оборудования (узлами дерева являются повторители, листьями станции или мосты, с помощью мостов можно подключать сегменты Ethernet или Token Ring)
Метод доступа получил название Demand priority. Основывается на программном автомате под названием MAC state machine.

Таким образом, существуют три основных способа выбора активного передатчика:

1) Перепасовка маркера (token passing)
2) Резервирование (reservation)
3) Опрос (polling)

Выбор может происходить и по расписанию

Если в топологиях с множественным доступом канал не является моноканалом, то совместно использоваться он может следующими методами

1) FDMA (Frequency Division Multiple Access) множественный доступ на разных частотах (частотное разделение)
2) TDMA (Time Division Multiple Access) множественный доступ на одной частоте в разные временны е окна (временно е разделение)

3) CDMA (Code Division Multiple Access) множественный доступ на одной частоте с изменением параметров кодирования

34) Адресация в компьютерных сетях и классификация адресов

Для того, чтобы станции абоненты могли организовать взаимодействие,

им необходимо некоторым образом выделять друг друга среди других станций. С целью идентификации станций им присваивают некоторые адреса. Таким образом, возникает адресация в СПД.

Как было сказано ранее, в форматах большинства пакетов присутствуют

два адреса:

1) Адрес назначения (destination address)
2) Адрес источника (source address)

В процессе пересылки пакета между абонентами адресация играет ключевое значение. Производительность СПД напрямую зависит от расположения адресов в пакете. Поэтому адреса «выносят» в самое начало пакета. Более того, поскольку с точки зрения доставки пакета адрес назначения является более важным (в СПД анализируется именно этот адрес), он как правило располагается раньше.

Многие топологии предполагают возможность приема, переданного одной из станций пакета всеми остальными станциями в пределах сегмента вне зависимости от того, какой из станций пакет был предназначен. Следует различать действия «принят станцией», «проанализирован станцией» и «обработан станцией». Факт приема станцией пакета подразумевает, что пакет будет проанализирован, но не подразумевает «полноценную» обработку. Именно сравнение считанного из принятого пакета адреса назначения со своим адресом, позволяет станции распознать пакет как «свой». Считанный из пакета адрес источника позволяет станции (при необходимости) определить абонента, создавшего пакет.

Следует учитывать, что важное влияние на адресацию оказывает инкапсуляция. Адресация всегда «привязана» к некоторому протоколу, а протокол, в свою очередь, «привязан» к уровню модели OSI. Поэтому закономерно, что на каждом из уровней присутствует своя независимая система адресации. Пакет, воспринятый как «свой» на одном из уровней, после его передачи на более высокий уровень, там вполне может быть «отвергнут». Кроме того, «окончательная» обработка не всегда происходит на прикладном уровне (классический пример ретрансляция пакета между сегментами при маршрутизации).

В каждом пакете должны присутствовать по крайней мере адреса канального уровня. В большинстве же практических реализаций семейств

протоколов, кроме адресации на канальном уровне, предусмотрена адресация на сетевом (в связке с транспортным) и прикладном уровнях.

Адреса канального уровня «зашиваются» в сетевое оборудование при его производстве и поэтому повторяться не должны. Они не предполагают возможность пользовательского вмешательства и их считают абсолютно уникальными. Часто такую адресацию называют физической. Адреса сетевого и прикладного уровней назначают пользователи. Часто такую адресацию называют логической.

В нормальной ситуации, по крайней мере в течение сеанса взаимодействия, адреса разных уровней одной станции должны соответствовать друг другу. Поэтому возникает необходимость в служебных протоколах, отыскивающих эти соответствия.

Для того, чтобы взаимодействующие сетевые процессы могли найти друг друга, во всех реальных системах используется три уровня адресации

1) Необходимо адресовать подсеть используется адрес подсети (subnet address).
2) Необходимо адресовать станцию в подсети используется адрес станции (node address)
3) Необходимо адресовать процесс в станции используется так называемый адрес программного порта (software port)

Под адрес порта, как правило, отведено два байта

При назначении программных портов учитываются диапазоны, к которым они относятся.

Диапазоны программных портов применительно к семейству TCP/IP

Port Number Range	Port Group
0 – 1023	Well Known
1024 – 49151	Registered
49152 – 65535	Private and Dynamic

Так называемые хорошо известные порты предназначены для адресации основных сервисов в Internet. Порты для дополнительных публичных сервисов нужно регистрировать. Порты для приватных (и редких) сервисов регистрировать не нужно.

Специально для компьютерных сетей были разработаны четыре основных способа адресации, которые заключаются в применении адресов четырех базовых типов:

1) Юникаст - пакет с таковым адресом назначения должен быть обработан одной соответствующей станцией.
2) Бродкаст - или, по-другому, широковещательных - пакет с таковым адресом назначения должен быть обработан всеми станциями.
3) Мультикаст - пакет с таковым адресом назначения должен быть обработан несколькими станциями из множества.

4) Эникаст - пакет с таковым адресом назначения должен быть обработан одной станцией из множества.

По сути, мультикаст и эникаст адреса являются групповыми идентификаторами (Group IDs).

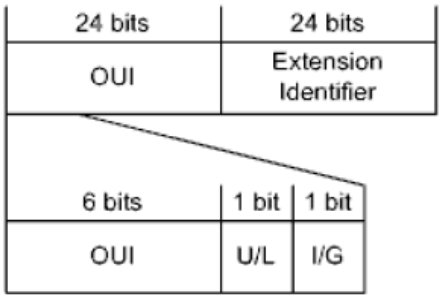
Бродкаст мультикаст и эникаст адреса не могут быть адресами источников, так как отдельно взятый пакет может сгенерировать только одна станция.

Наиболее сложной формой адресации является эникаст адресация.

35) MAC-адреса

Уникальность MAC адресов контролирует IEEE RA (IEEE Registration Authority). В стандартах IEEE определены три базовых формата MAC адресов MAC-48 EUI-48 и EUI-64 где EUI (Extended Unique Identifier) расширенный уникальный идентификатор. MAC 48 можно считать синонимом EUI 48 хотя изначально это было более общее понятие.

Формат EUI-48:



00-16-41-57-7D-48

Поля:

OUI (Organizationally Unique Identifier) -- уникальный идентификатор организации.

U/L (Universal/Local) -- признак универсальности-локальности адреса.

I/G (Individual/Group) -- признак индивидуального-группового адреса.

Extension Identifier -- идентификатор-наполнитель.

OUIs выдают централизовано, уникальность оставшейся части должны обеспечивать сами организации (любым способом по своему усмотрению). Время валидности адресов (время, которое нужно выдержать перед повторным присвоением того же адреса другому устройству) определено как 100 лет. Иногда, при администрировании, возникает необходимость подменить адрес, «зашитый» в оборудование, на некий другой. Этот новый адрес называют локальным административным адресом. Его признаком является единичное значение бита U/L. Согласовывать значение остальных битов не требуется, но в пределах сегмента адрес не должен повторяться.

Граница между OUI и Extension Identifier может проходить не только посредине адреса. В общем случае предусмотрены три варианта разрядности поля OUI:

1) MA-L (MAC Address - Large) - 24 бита (данная схема использовалась IEEE RA до 1 января 2014 г
2) MA-M (MAC Address Medium) - 28 битов (схема доступна после 1 января 2014 г)
3) MA-S (MAC Address Small) 36 битов (схема доступна после 1 января 2014 г)

Иногда поле OUI рассматривают как CID (Company ID), что, по большому счету, то же самое зависит от комбинации значений битов U/L и I/G (рассматривают уже как биты X и M соответственно). При так называемом каноническом представлении MAC-адрес сдвигается в канал начиная со старших разрядов.

По правилам IEEE MAC-адреса записывают в следующей нотации: XX-XX-XX-XX-XX-XX

Где X-шестнадцатеричная цифра (верхний регистр)

Но очень часто используют альтернативные нотации. Примеры:

```
00-16-41-57-7D-48 -- IEEE
00-16-41-57-7d-48
00:16:41:57:7D:48
00:16:41:57:7d:48
0016.4157.7d48 -- Cisco
```

Все юникаст-MAC-адреса должны иметь нулевое значение бита I/G. Групповые MAC-адреса формируются по особым правилам. В качестве бродкаст-MAC-адреса принято использовать значение:

FF-FF-FF-FF-FF-FF

Следует отметить, что EUI-64 может использоваться не только для адресации, а и для просто идентификации устройств.

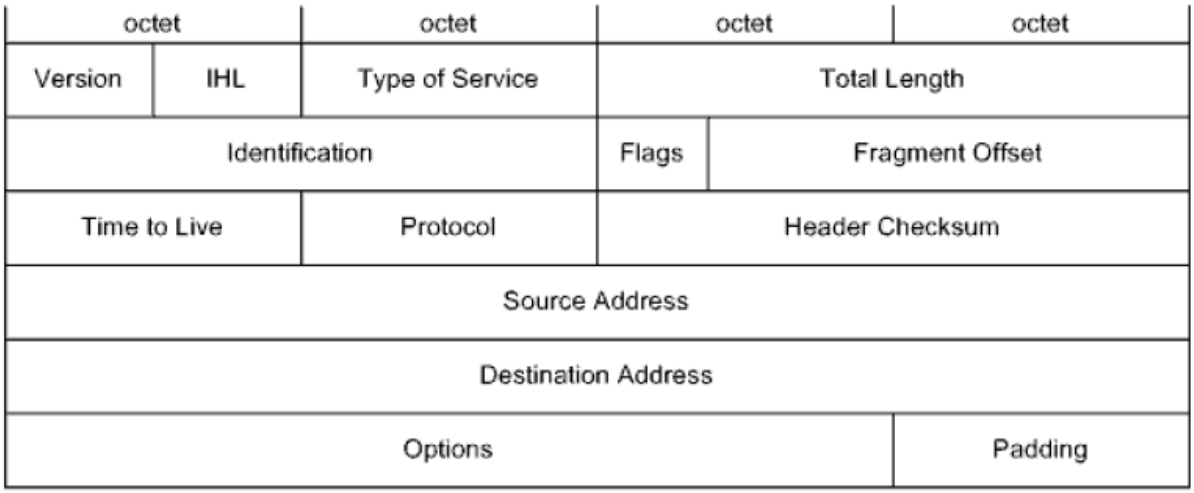
Примеры технологий с применением EUI-48: Ethernet, Wi-Fi, Token Ring

Примеры технологий с применением EUI-64: IPv6, FireWire

36) Заголовок IPv4

В семействе TCP/IP за адресацию на сетевом уровне отвечает протокол IP

Заголовок протокола IPv 4 (версии 4) имеет фиксированную структуру.



Поля:

1) Version - версия (значение равно 4)
2) IHL (Internet Header Length) - длина заголовка (в 32 ухбитных словах, минимальное значение равно 5)
3) Type of Service - тип сервиса (связано с QoS)
4) Total Length - общая длина данных (в байтах, не может превышать 65535 байтов)
5) Flags - флаги
6) Fragment Offset - смещение текущего фрагмента (в 64 ехбитных словах, смещение первого фрагмента равно нулю)
7) Time to Live - «время жизни» (при каждой ретрансляции уменьшается, когда становится равным нулю пакет уничтожается)
8) Protocol - протокол (инкапсулируемый в поле данных)
9) Header Checksum - контрольная сумма заголовка
10) Source Address - адрес источника
11) Destination Address - адрес назначения
12) Options - опции (связанные с безопасностью, размер вариативен)

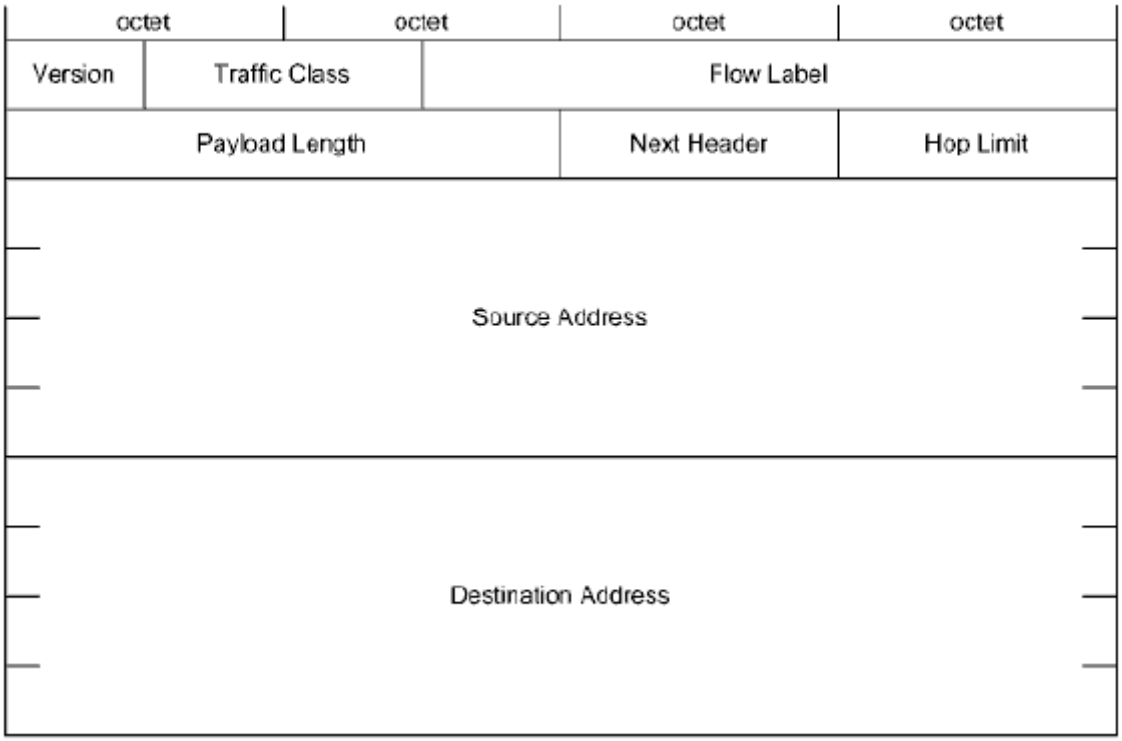
Поле flags:

DF (Don’t Fragment): 0 - пакет фрагментирован, 1 пакет нефрагментирован

MF (More Fragments): 0 - текущий фрагмент является последним, 1 текущий фрагмент не является последним

37) Заголовок IPv6

Заголовок протокола IPv6 имеет «гибкую» структуру. Заголовки «каскадируются» сколько заголовков нужно, столько и вставляется



Новые поля:

1) Traffic Class - класс трафика (связано с QoS)
2) Flow Label - метка потока (связано с QoS)
3) Payload Length - длина полезной нагрузки (в байтах, аналог поля Total Length)
4) Next Header - селектор следующего заголовка (в том числе, аналог поля Protocol)
5) Hop Limit ограничитель числа «прыжков» (аналог поля Time to Live)

Полноценная реализация IPv6 должна поддерживать следующие заголовки:

1) IPv6 header - собственно IPv6 заголовок
2) Hop-by-Hop Options header - заголовок опций ретрансляции
3) Destination Options header - заголовок предназначенных станции назначения опций
4) Routing header - маршрутизационный заголовок
5) Fragment header - заголовок франмента

6) Authentication header - заголовок протокола AH (связано с защитой информации)
7) Encapsulating Security Payload header - заголовок протокола ESP (связано с защитой информации)

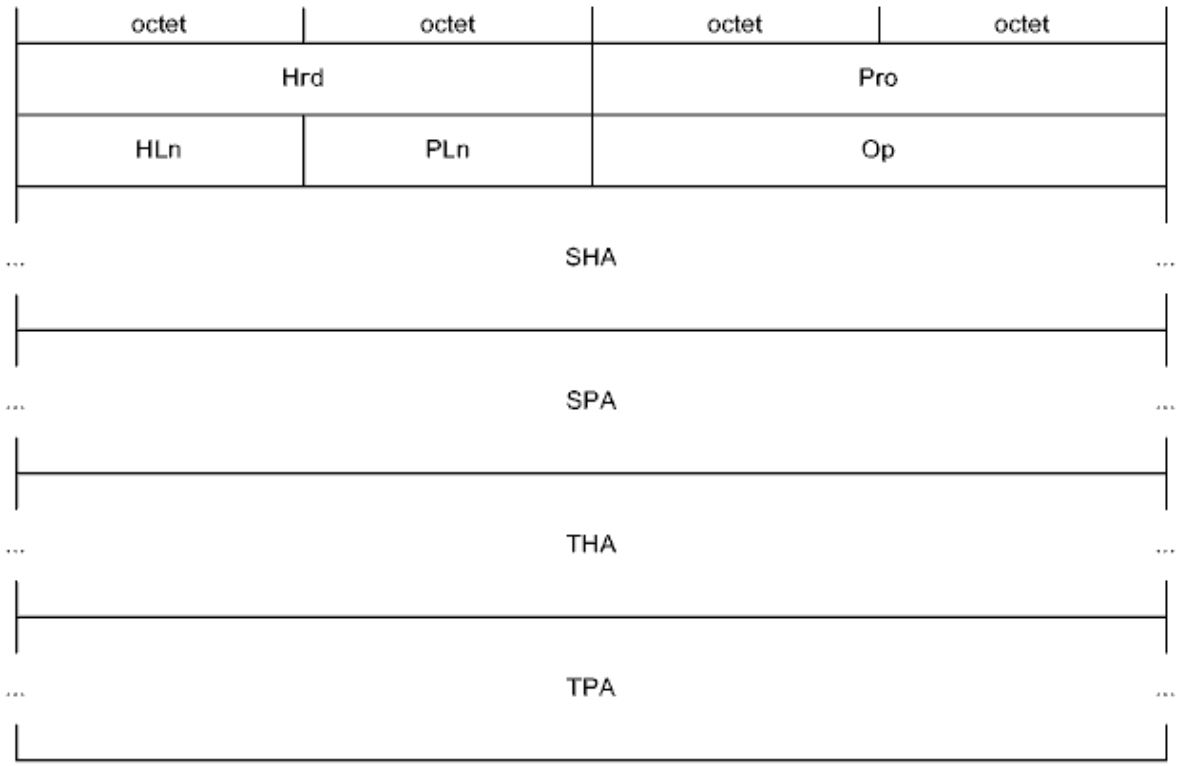
+8) Upper layer header - заголовок протокола вышестоящего уровня

38) Протокол ARP

Группа протоколов под названием ARPs (Address Resolution Protocols) предназначена для восстановления соответствий между MAC адресами и IP-адресами. Под прямым преобразованием, собственно ARP, понимают нахождение MAC адреса по IP адресу. Обратное преобразование выполняется по протоколу RARP (Reverse ARP)

Протокол арп работает в режиме вопрос-ответ, скажи мне здесь привести пример арп запрос арп ответ и арп таблица в которой есть тип.

Формат пакета ARP:



Поля

1) Hrd - тип оборудования (1 - Ethernet)
2) Pro - протокол (800h - IP)
3) HLn (Hardware address Length) - длина аппаратного (физического) адреса (в байтах, 6 - Ethernet)

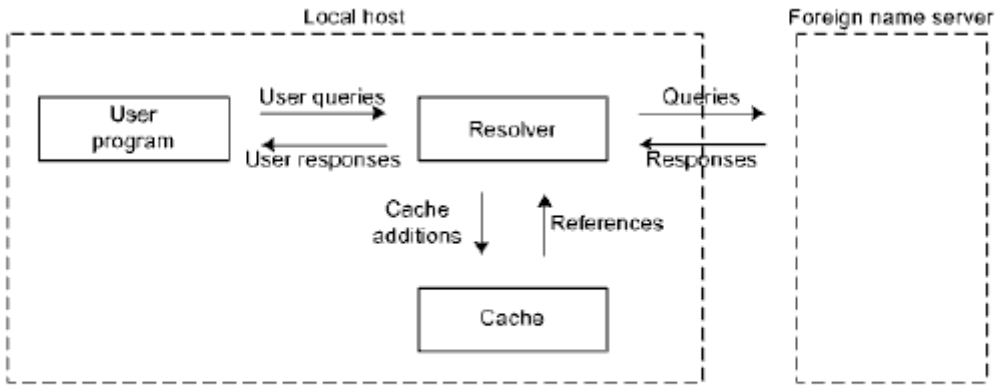
4) PLn (Protocol address Length) - длина протокольного (логического) адреса (в байтах, 4 - IP)
5) Op (Opcode) - код операции: 1 – Request – запрос, 2 – Reply – ответ (и некоторые другие)
6) SHA (Sender Hardware Address) - аппаратный адрес запрашивающей станции
7) SPA (Sender Protocol Address) - протокольный адрес запрашивающей станции
8) THA (Target Hardware Address) - аппаратный адрес запрашиваемой станции
9) TPA (Target Protocol Address) - протокольный адрес запрашиваемой станции

39) Структура системы DNS

Протокол системы DNS (Domain Name System) (основные RFCs 1034 и 1035) предназначен для восстановления соответствий между IP адресами и адресами прикладного уровня.

Следует отметить, что под доменом (иногда cloud) в СПД обобщенно понимают совокупность устройств, работающих в рамках некоторых единых правил.

Структура системы DNS:



Думаю можно и без этой картинки просто текст

Система DNS соответствует клиент серверной модели и включает три основных компонента:

1) Адресное пространство доменных названий (domain name space) и записи о ресурсах - RRs (Resource Records)
2) Серверы названий (name servers)
3) Программы, отвечающие на запросы клиентов (resolvers)

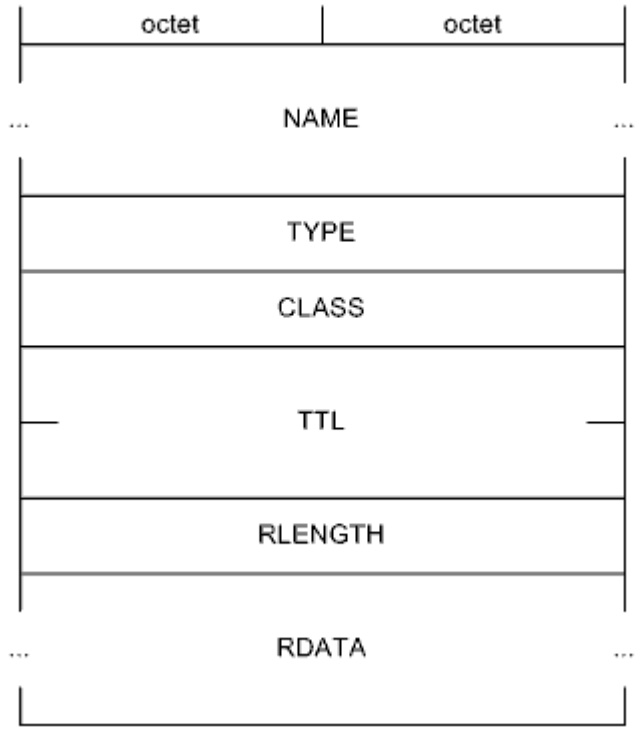
Каждый из этих компонентов «видит» систему DNS по-своему. Адресное пространство доменных названий имеет иерархическую древовидную структуру. Каждый узел дерева на некотором уровне иерархии обозначают DNS меткой (DNS label) длиной от 0 до 63 байтов (должна начинаться с буквы и состоять из комбинации букв любого регистра, цифр и символа -). Метка нулевой длины зарезервирована и является корнем дерева. При присоединении станции к определенному домену ей так же присваивают метку. Доменное название строится из меток в соответствии с путем к корневой метке. Полная длина не может превышать 255 байтов. Доменное название может относиться как к отдельно взятой станции, так и к некоторой ветви дерева, то есть к DNS домену (DNS domain). Доменное название может быть, как абсолютным, то есть содержащим всю цепочку меток от станции до корневой метки, так и относительным, то есть содержащим только часть меток. Внутреннее представление метки один байт, в котором указана длина метки, за которым следуют собственно байты метки. При интерпретации меток регистр букв не учитывается.

Согласно принятой нотации записи доменных названий метки разделяют точками, и корневая метка является крайней справа.

40) Сообщения DNS

Каждой входящей в систему DNS станции (как и каждому домену) соответствует некоторое количество RRs

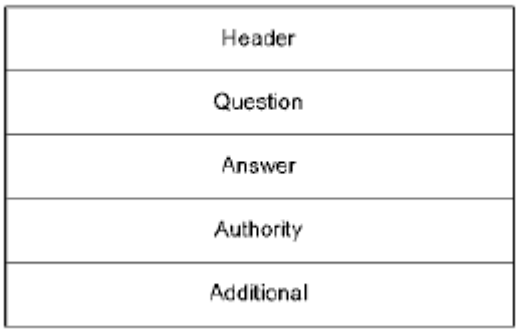
Формат DNS RR:



Поля:
1) NAME - доменное название (к которому относится RR, целевое при поиске)
2) TYPE - тип
3) CLASS - класс (семейство протоколов)
4) TTL (Time To Live) - «время жизни» (то есть время валидности RR, в секундах)
5) RLENGTH (Resource LENGTH) - длина данных ресурса
6) RDATA (Resource DATA) - данные ресурса (зависят от типа и класса)
Основные типы RRs:
1) A (A host address) - IP адрес хоста
2) NS (Name Server) - авторитетный сервер названий домена
5) CNAME (Canonical NAME) - каноническое доменное название (станции либо домена, для псевдонима)
6) SOA (Start of a zone of Authority) - оригинальные параметры зоны (сервер с изначальным описанием зоны, контактное лицо, время валидности и другие)
10) NULL - нулевая запись (произвольная информация)
12) PTR - указатель доменное название станции (при обратных преобразованиях)
13) HINFO (Host INFO) - информация о станции (процессор и ОС)
15) MX (Mail eXchange) доменное название почтового сервера в домене (включая приоритет, этот тип используется и вместо нескольких отмененных типов)
16) TXT (TeXT strings) - текстовые строки (либо строка)
28) AAAA (-) IPv6 адрес хоста
33) SRV (SeRVer selection) - описание сервиса (любого дополнительного сетевого сервиса на станции, например, файлового)
Классы RRs:
1) IN - Internet
2) CS - CSNET (устарел и аннулирован)
3) CH - Chaosnet (устарел)
4) HS - Hesiod (для БД, очень редкий)
Остальные значения классов зарезервированы

Примеры значений RRs класса IN:
A: 192.168.11.1.
CNAME: 5-508-fileserv.bsuir.by.
MX: 10 mail.bsuir.by.
NS: proxy1.bsuir.by.
PTR: 5-508-fileserv.bsuir.by.

Формат сообщения DNS:



Поля:

1) Header - заголовок
2) Question - вопрос
3) Answer - ответ
4) Authority - авторитетный ответ
5) Additional - дополнение

Заголовок присутствует всегда, остальные поля вариативны

41) Виртуальные соединения в сети передачи данных

Одним из ключевых терминов транспортного уровня является термин соединение. По сути дела, понятие соединения связано с понятием готовности. Если абоненты находятся в состоянии «нормальной готовности» передавать или принимать данные, то считают что между ними установлено соединение. С учетом абстрагирования от более низких уровней модели OSI и инкапсуляции, соединение может быть выражено неявно.

Нужно отличать виртуальные соединения (virtual connections) от физических соединений (physical connections). Абоненты программы физически соединены быть не могут. Следовательно, применительно к ним, соединения являются сугубо виртуальными.

Следует также учитывать, что нормальная готовность может рассматриваться в двух ракурсах:

1) Организация взаимодействия абонентов программ
2) Настройка задействованного промежуточного оборудования

В первом случае речь идет о собственно виртуальных соединениях транспортного уровня, во втором о виртуальных цепях (virtual circuits) сетевого или канального уровней

В свою очередь, виртуальные цепи бывают:

1) PVCs (Permanent Virtual Circuits) выделенные виртуальные цепи
2) SVCs (Switched Virtual Circuits) коммутируемые виртуальные цепи (в отечественной литературе иногда называют виртуальными вызовами

Термин виртуальный канал (virtual channel) может в равной степени подходить как к виртуальным соединениям, так и к виртуальным цепям

42) Классификация оконных механизмов, используемых в сети передачи данных

Простейшим подходом к обеспечению контроля доставки информационных пакетов является применение метода, который обобщенно можно назвать методом запросов подтверждений (requests/acknowledges)

В случае, когда СПД загружена незначительно, а взаимодействующие абоненты расположены далеко друг от друга, задействование классического механизма запросов подтверждений приводит к неэффективному использованию ресурсов Время, затрачиваемое на ожидание квитанций, становится недопустимо большим в сравнении с временем, затрачиваемым на передачу полезных данных. Оптимизировать обмен позволяет применение оконного метода, суть которого состоит в том, что до перехода к ожиданию квитанций передается не один, а несколько пакетов.

Выделяют два основных критерия классификации оконных методов

Исходя из количества пакетов, передаваемых в окне, оно может быть:

1) Статическим - неизменяемый размер окна заложен в протокол или устанавливается на весь сеанс обмена
2) Динамическим размер окна может изменяться (увеличиваться или уменьшаться) в процессе передачи сообщения

Исходя из способа обработки очереди пакетов, окно может быть:

1) Фиксированным перед формированием следующего окна текущее должно быть полностью «закрыто», то есть должны быть приняты все необходимые квитанции
2) Скользящим существует возможность сдвигать окно относительно последовательности пакетов

При реализации оконного метода следует учитывать следующие дополнительные обстоятельства

-- нужна нумерация пакетов в том или ином виде

-- подтверждаться может как все окно, так и каждый из пакетов

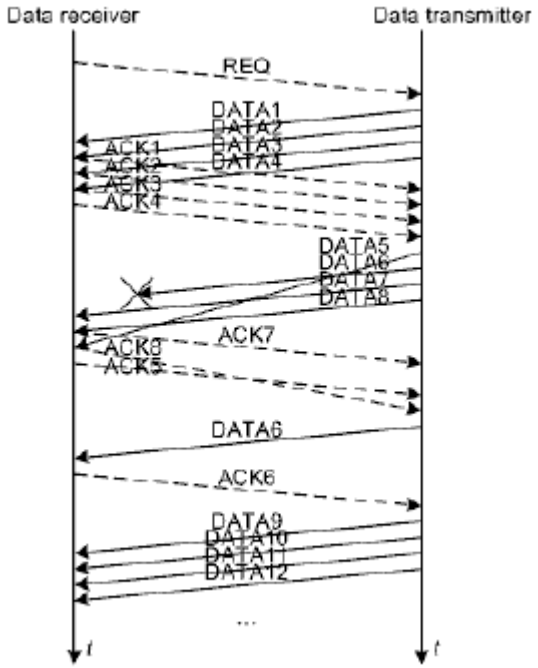
-- размером окна может управлять как передатчик, так и приемник

-- размером окна можно управлять посредством служебных полей, в том числе и в информационных пакетах

-- окно, с которым работает передатчик, может отличаться от окна, с которым работает приемник

-- иногда важен порядок доставки пакетов

С точки зрения реализации, наиболее простым является статическое окно фиксированного размера



Основной его недостаток состоит в отсутствии возможности адаптации к изменениям в СПД

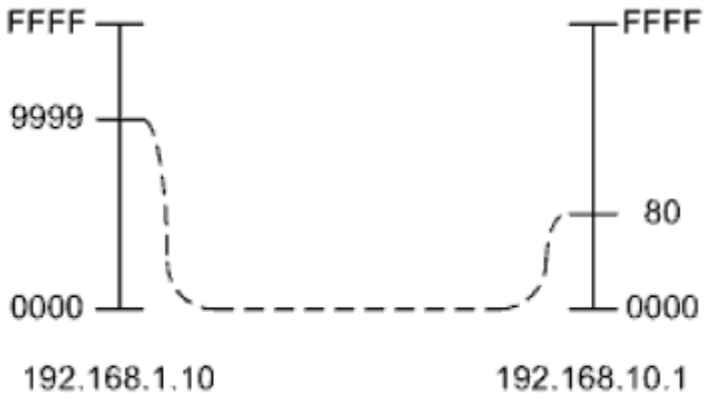
Динамическое окно позволяет успешно адаптироваться к изменениям в СПД. При увеличении загруженности окно целесообразно сужать, а при снижении – расширять.

Скользящее окно, особенно в сочетании с динамическим, позволяет ускорить адаптацию к топологическим и другим изменениям в СПД.

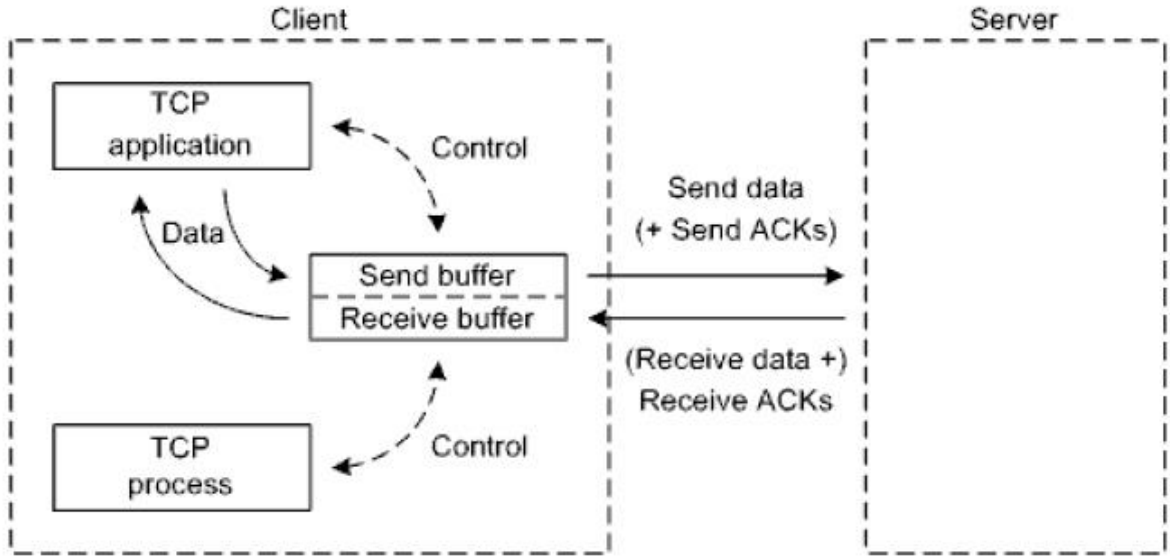
Таким образом, наиболее сложным является динамическое скользящее окно.

43 Структура системы TCP

TCP соответствует клиент серверной модели. Сокет - это «привязка» к виртуальному каналу, соединяющему между собой два взаимодействующих сетевых процесса, с точки зрения одного (из этих процессов, причем с учетом всех трех уровней адресации.



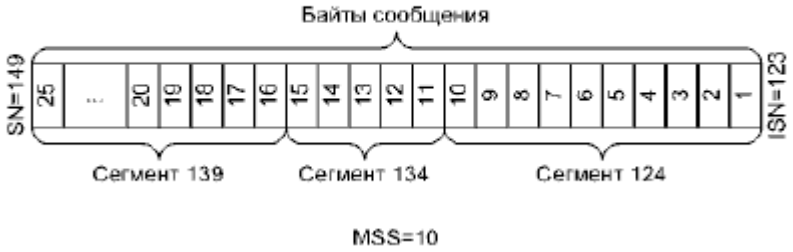
Структура соединения TCP:



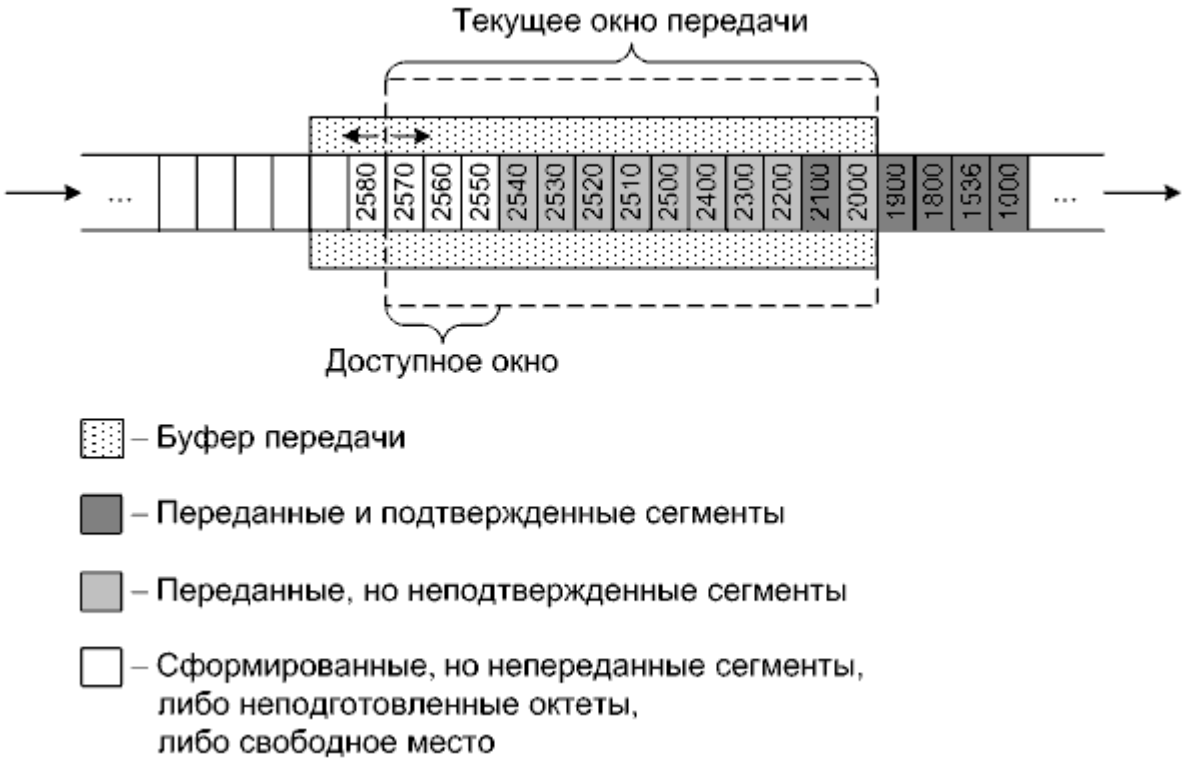
Тоже можно только текст

Применительно к каждому TCP соединению нужно выделять приложение, производящее или потребляющее сетевые данные, и TCP процесс, предоставляющий коммуникационные услуги (например, специальный драйвер ОС). Синхронизировать работу приложения и TCP процесса можно только с помощью буферизации. TCP интерфейс, которым пользуется приложение, состоит из примитивов для работы с буфером, позволяющих контролируя записывать или считывать данные. Доступ к буферу имеет и TCP процесс, который отслеживает наполнение буфера и, используя ресурсы более низких уровней, организует прием или передачу данных.

Предназначенное для передачи сообщение разбивается на сегменты. Минимальной учитываемой в окне единицей данных является октет, то есть байт. Все байты сообщения последовательно нумеруются так называемыми последовательными номерами SNs (Sequence Numbers). Нумерация начинается с некоторого начального последовательного номера ISN (Initial Sequence Number), который как правило не равен нулю, а генерируется реализациями для того чтобы лучше управлять соединениями. Принято, что сам ISN в нумерацию байтов не включается, то есть номер первого байта сообщения больше ISN на единицу. Номером сегмента является SN первого байта данных в нем. По разным понятным причинам длина сегмента может варьировать, но она имеет ограничение. Поэтому важное значение имеет конфигурационный параметр MSS (Maximum Segment Size) максимальная длина сегмента (по умолчанию 536 байтов)



В стандарте выделяют несколько видов окон, которые нужно различать. Благодаря гибкости протокола, передающий и принимающий TCP процессы работают с разными окнами, то есть, в первую очередь, следует отдельно рассматривать окно передачи и окно приема.

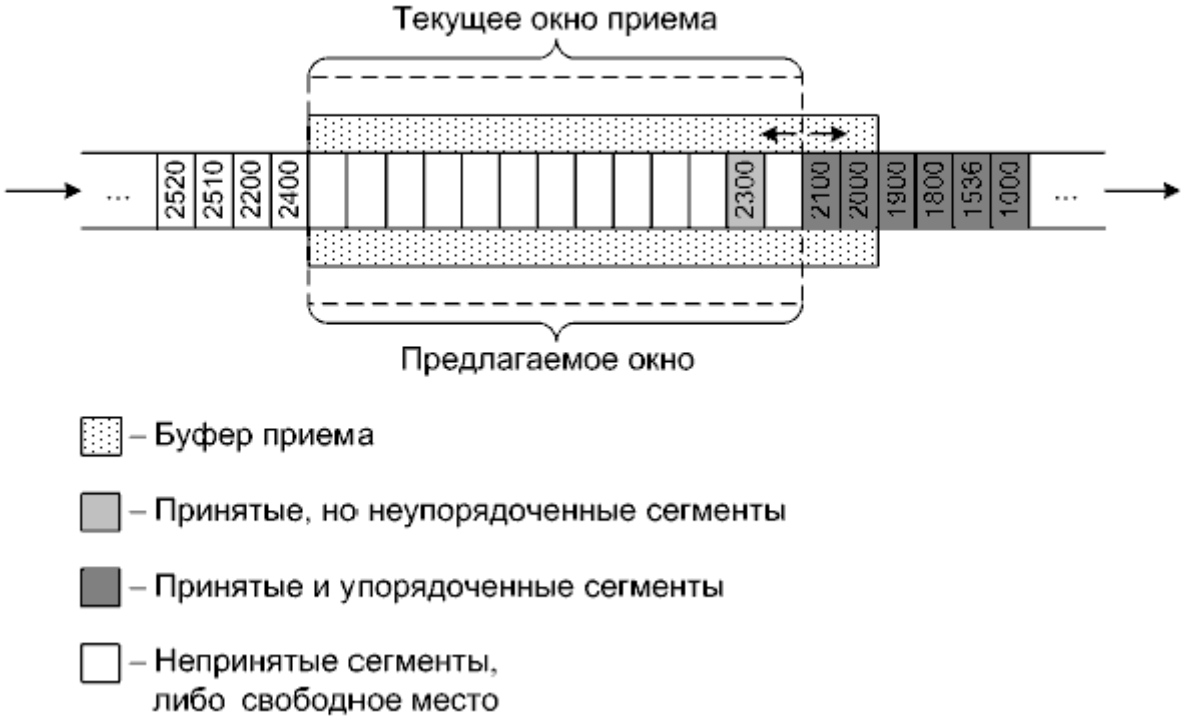


Передающее приложение последовательно, «порциями», записывает блоки байтов сообщения, возможно разной длины, в буфер передачи. Длина сообщения и размер буфера — это вещи независимые, они почти всегда различаются TCP процесс формирует из имеющихся в буфере данных соответствующее количество сегментов и последовательно отправляет их. В любой момент времени текущее окно (current window) передачи имеет некоторый установленный размер и характеризуется тем, что все попадающие в него сегменты с данными можно передавать без ожидания подтверждений. Его правая (на рисунке) граница совпадает с правой границей буфера и скользит налево относительно последовательности сегментов с данными по мере поступления и упорядочивания подтверждений. Переданные, но неподтвержденные сегменты с данными

продолжают оставаться в буфере, так как возможно потребуется их повторная передача.

Левая граница «привязана» к правой в соответствии с размером текущего окна. Но поскольку размер подвержен динамической коррекции, положение левой границы относительно правой постоянно из меняется. Область текущего окна передачи за вычетом переданных, но неподтвержденных сегментов с данными, является доступным окном (useable равно effective window). TCP-процесс должен последовательно отправить все сегменты с данными, попавшие в эту область. Если размер текущего окна передачи равен нулю, то передача приостанавливается полностью

Окно приёма:



На другой стороне соединения, возможно уже разупорядоченные при преодолении СПД сегменты поступают в буфер приема (размер может не совпадать с размером буфера передачи) При этом они размещаются там согласно своим номерам. Текущее окно приема охватывает часть буфера, в которой можно размещать еще неупорядоченные сегменты с данными. Как и текущее окно передачи, в любой момент времени оно так же имеет

некоторый определенный размер. Левая (на рисунке) граница текущего окна приема совпадает с левой границей буфера. Правая граница проходит слева за последним упорядоченным сегментом с данными и поэтому динамически меняет свое положение относительно левой границы. По мере считывания принимающим приложением упорядоченных байтов из буфера окно скользит относительно последовательности сегментов с данными. Если

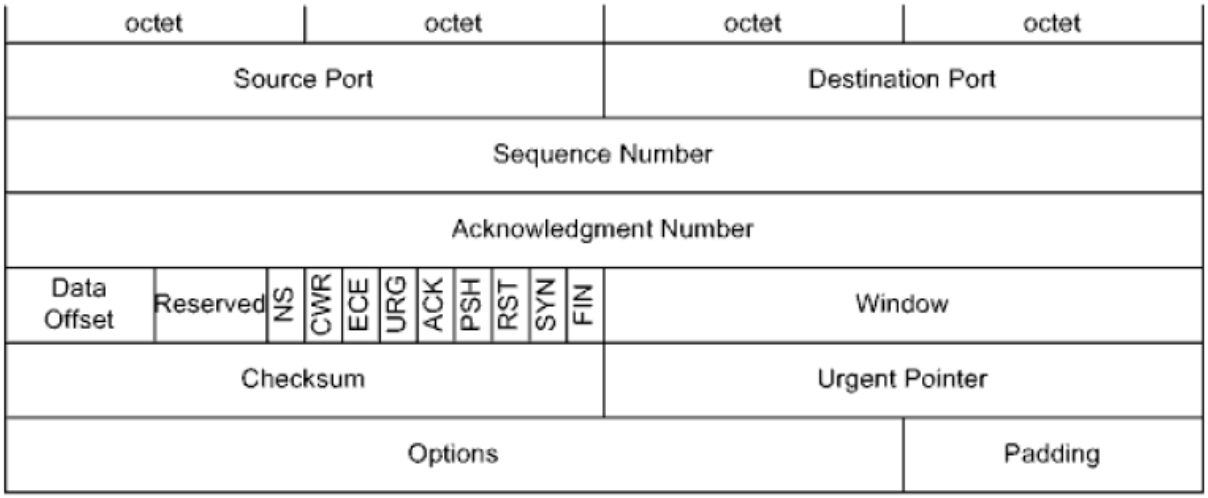
размер текущего окна приема равен нулю, а сегменты с данными продолжают поступать, то возникает переполнение.

Вполне закономерно, что именно на принимающий TCP-процесс, как на более подверженный влиянию недетерминированности СПД, возложен контроль «поведения» оконного механизма. Это делается посредством «обратной связи». Принимающий TCP процесс пытается информировать передающий о состоянии своего буфера, точнее о наличии в нем свободного места. Для этого он при подтверждениях сообщает предлагаемое окно (announced равно advertised равно offered window).

В качестве размера предлагаемого окна указывается размер текущего окна приема. Последствия разупорядочивания сегментов с данными такому подходу не противоречат.

Максимальный размер любого из окон не может превышать размер соответствующего буфера.

44) Заголовок TCP



Поля:

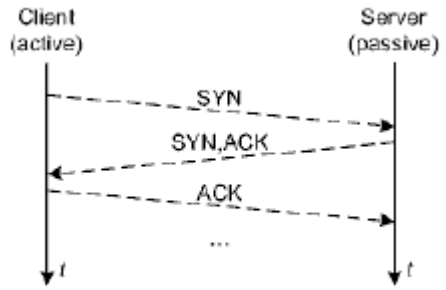
1) Source Port - программный порт источника
2) Destination Port - программный порт назначения
3) Sequence Number (SN) - последовательный номер (сегмента)
4) Acknowledgment Number (AN) - подтверждающий номер
5) Data Offset - смещение данных (в 32 ухбитных словах)
6) Reserved - зарезервировано (должно равняться нулю)
7) URG (URGent Pointer field significant) - флаг значимости указателя на экстренные данные
8) ACK (ACKnowledgment field significant) - флаг значимости подтверждающего номера

9) NS (Nonce Sum) – флаг - контрольная сумма для проверки правильности кодов явных уведомлений о заторах (связан с QoS, связан с IP заголовком)
10) CWR (Congestion Window Reduced) - флаг уменьшения окна затора при явном уведомлении о заторе
11) ECE (Explicit Congestion Notification Echo) - флаг подтверждения явного уведомления о заторе
12) PSH (PuSH Function) - флаг принудительной доставки данных (без буферизации)
13) RST (ReSeT the connection) - флаг разрыва соединения (из-за сбоя на одной из взаимодействующих сторон)
14) SYN (SYNchronize sequence numbers) - флаг синхронизации последовательных номеров
15) FIN (No more data from sender) - флаг последних данных
16) Window (W) - предлагаемое окно
17) Checksum - контрольная сумма
18) Urgent Pointer - указатель на экстренные данные
19) Options - опции (MSS)
20) Padding – наполнитель

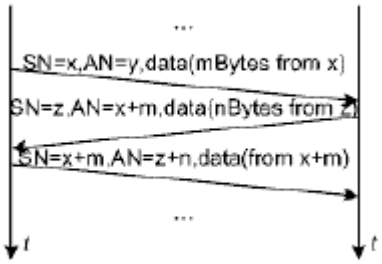
45) Протокол TCP

Функционирование оконного механизма TCP базируется на использовании трех полей в заголовке сегмента SN, AN, W, и трех флагов (из шести стандартизованных изначально) SYN, ACK, FIN

Установление TCP соединения, известное как «тройное рукопожатие» (three way handshake), основывается на использовании флагов SYN и ACK



Не смотря на то, что процесс установления соединения несимметричен, в дальнейшем, в общем случае, оно используется в полнодуплексном режиме

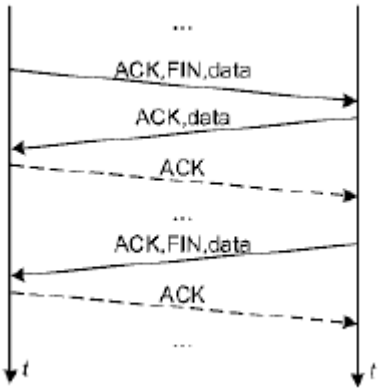


Полнодуплексность самого соединения достигается за счет того, что передаваемый в определенном направлении сегмент служит одновременно для транспортировки как данных и связанных с ними служебных полей от передающей составляющей TCP процесса, так и подтверждений и связанных с ними других служебных полей от принимающей составляющей TCP процесса.

По правилу протокола, поле SN пересылаемого сегмента отражает собственный SN этого сегмента. По другому правилу, в поле AN указывается SN ожидаемого сегмента, коим является следующий по порядку сегмент. При установлении соединения данные не пересылаются. Поэтому, для того чтобы не нарушать указанные правила, в качестве SNs используют невключенные в нумерацию байтов сообщения ISNs, а в качестве ANs просто инкрементированные SNs. Обойтись без передачи SNs при установлении соединения невозможно, так как стороны должны однозначно идентифицировать это соединение. После синхронизации SNs соединение считается установленным.

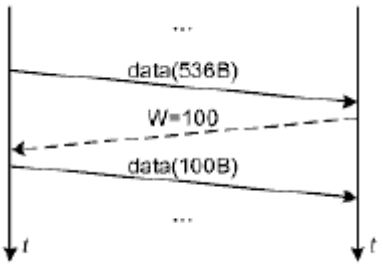
Не смотря на предоставляемые возможности, данные вполне могут пересылаться только в одном направлении, то есть в симплексном режиме. При этом в направлении, попутном направлению пересылки данных, в качестве AN используется SN следующего по порядку несуществующего (вообще, либо уже, либо пока) сегмента, что никоим образом не противоречит уже приведенным правилам. Если сегментов с данными пересылается несколько, то ANs дублируются столько раз, сколько нужно. Это приводит к дублированию SNs в ответных сегментах без данных. Аналогичные дублирования возникают и при приостановке пересылки данных в определенном направлении.

Поскольку при установлении соединения оно всегда открывается в двух направлениях (по инициативе клиента, но может использоваться в одном любом направлении), для нормального завершения оно и закрыто должно быть в обоих направлениях. Для закрытия соединения в своем направлении, сторона, в соответствующем сегменте (обычно с последними данными), устанавливает флаг FIN.



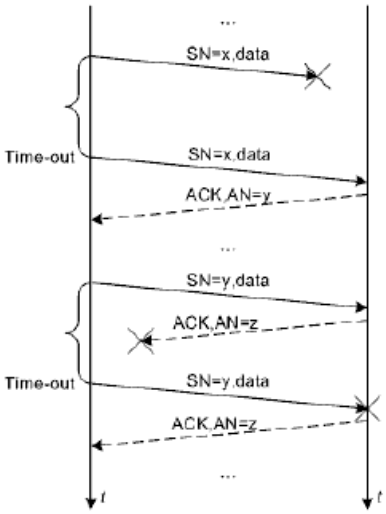
Соединение, нормально закрытое только в одном направлении, или ненормально завершенное на одной из сторон без уведомления другой стороны (в результате сбоя) называют полуоткрытым (half open)

Размер предлагаемого окна в поле W может изменяться каждый раз для соответствующей коррекции текущего окна передачи, в том числе и при установлении соединения для изменения размера текущего окна передачи по умолчанию.



В случае задания нулевого значения поля W передача данных фактически запрещается. После освобождения места в буфере приема подтверждение обязательно повторяется с уже ненулевым полем W, что «разблокирует» передающую сторону.

Проблема возможной потери в СПД некоторых сегментов решается с помощью тайм аутов



Передающий TCP-процесс определяет потерю сегмента с данными либо его подтверждения по отсутствию этого подтверждения в течение установленного интервала времени. После наступления тайм аута сегмент с данными передается повторно. Отрицательные подтверждения не предусмотрены вообще. Принимающий TCP процесс подтверждает все принятые сегменты с данными, причем подтверждает всегда. При этом если принята копия (что говорит о потере подтверждения), то она удаляется. Получение сегмента с SN больше ожидаемого говорит о возможной потере сегментов с данными или о разупорядочивании.

46) Усовершенствования протокола TCP

Хорошо известна проблема, вошедшая в историю под обобщенным названием «синдром глупого окна» («silly window syndrome»), в свое время «стопорившая» значительную часть пространства Internet. Синдром может возникать по разным причинам и проявляется в том, что текущее окно передачи не соответствует состоянию приемника, тем самым не позволяя его как следует «нагрузить» либо, наоборот, «разгрузить». Решение Нэгла (позволяет побороть «синдром глупого окна» когда передающей стороне требуется часто отправлять небольшие сегменты с данными. Решение Кларка (позволяет побороть «синдром глупого окна» когда принимающей стороной часто а но нсируется небольшое предлагаемое окно. Также стандартизированы четыре дополнения Ван Якобсона, призванные бороться с перегрузками в СПД

1) Медленный старт (slow start). Идеязаключается в том, что в начале передачи размер текущего окна передачи нужно увеличивать не «скачком», а плавно, пропорционально скорости получения подтверждений (не превышая размер предлагаемого окна)
   ТОЛЬКО ТАМ ГДЕ cwnd и после что что значит

Рекомендуемые формулы:

IW = 2 * *SMSS*, если *SMSS* > 2190 Bytes ,
IW = 3 * *SMSS*, если 2190 Bytes >= *SMSS* > 1095 Bytes ,
IW = 4 * *SMSS*, если *SMSS* <= 1095 Bytes ,

где *IW* (initial window) -- начальное значение текущего окна передачи:

cwnd += min (*N*, *SMSS*) ,

где *cwnd* (congestion window) -- текущее окно передачи (в данном случае, окно затора), *N* -- количество подтвержденных байтов, *SMSS* (sender MSS) -- MSS передатчика.

2) Избегание затора (congestion avoidance). Состоит в сдерживании экспоненциального роста размера текущего окна передачи после преодоления им некоторого порога. Как правило переход к избеганию затора происходит после медленного старта.

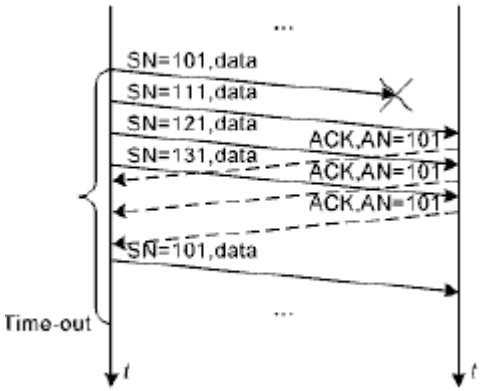
Рекомендуемые формулы:

ssthresh = max (*FlightSize* / 2, 2 * *SMSS*) ,

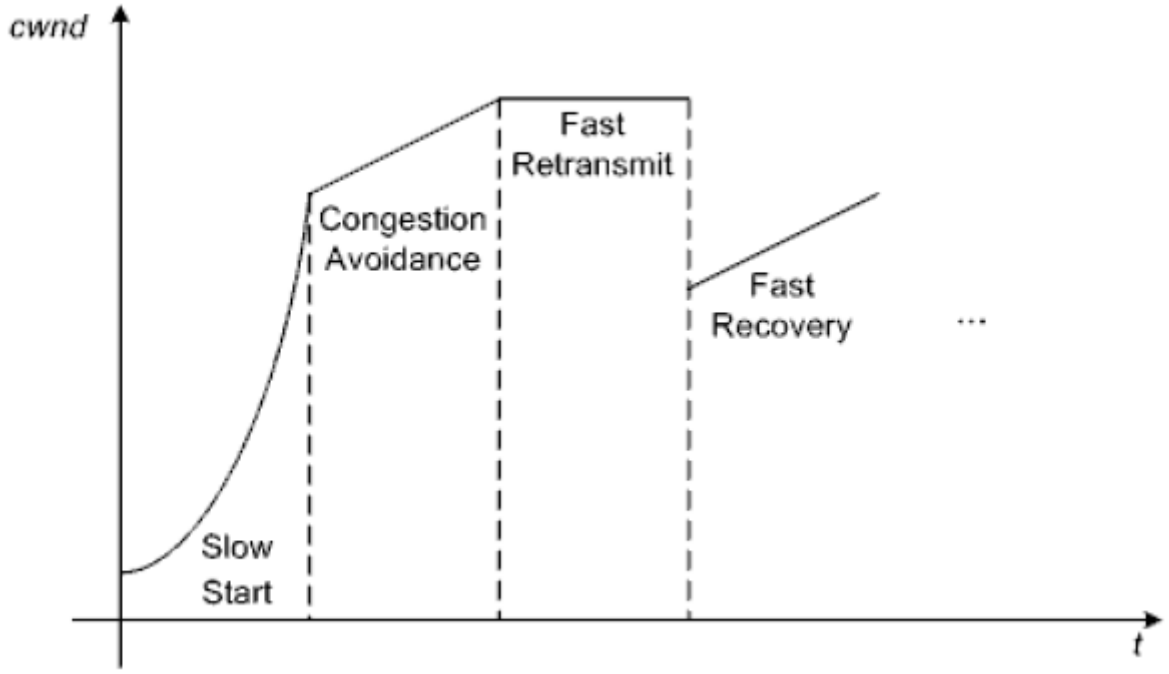
где *ssthresh* (slow start threshold) -- порог перехода от медленного старта к избеганию затора, *FlightSize* -- количество еще неподтвержденных байтов;

cwnd += *SMSS* * *SMSS* / *cwnd* .

3) Быстрая повторная передача (fast retransmit). При получении принимающей стороной разупорядоченного сегмента с данными (возможно из за потери ожидаемого сегмента с данными) незамедлительный повтор подтверждения с AN недостающего сегмента с данными. При получении передающей стороной трех одинаковых подтверждений незамедлительный повтор сегмента с данными согласно AN. Что, в некоторых ситуациях, позволяет успешно переслать потерянный сегмент еще до наступления тайм аута.



4) Быстрое восстановление (fast recovery). После обнаружения затора, переход сразу к избеганию коллизий, минуя стадию медленного старта. Как правило в связке с быстрой повторной передачей.



Последствия потерь и разупорядочивания сегментов заключаются в разрушении «маятника» взаимодействия и приводят к необходимости еще одной важной оптимизации, четко проявляющейся при быстрой повторной передаче. Согласно базовому алгоритму все сегменты должны быть подтверждены, а значит, после быстрой повторной передачи принимающая сторона должна послать все недостающие подтверждения. Но стороны могут «договориться», что текущий AN отражает номер первого ожидаемого получателем сегмента плюс автоматически подтверждает все сегменты с меньшими номерами (cumulative acknowlegement).

47) Протокол UDP и заголовок UDP

Протокол транспортного уровня UDP (User Datagram Protocol) реализует способ пересылки данных без гарантии доставки, часто называемый дейтаграммным (datagram)(хотя user datagram это пакет с контролируемыми пользователем данными, а datagram это любой пакет с данными)

octet	octet	octet	octet
Source Port		Destination Port	
Length		Checksum	

Поля:

1) Source Port - программный порт источника
2) Destination Port - программный порт назначения

3) Length - длина дейтаграммы включая заголовок (в байтах)
4) Checksum - контрольная сумма (подзаголовок, плюс заголовка, плюс данных)
При вкладывании UDP дейтаграммы в IP пакет (IPv 4 IPv 6 между UDP заголовком и IP заголовком вставляется дополнительный так называемый UDP псевдозаголовок, в котором дублируются некоторые значения из основного IP заголовка.

48) Классификация и характеристики сред передачи данных

Все исконно используемые в КС СрПД можно разделить на пять типов:
1. Коаксиальные кабели (coaxials) с различным волновым сопротивлением.
2. Экранированные и неэкранированные кабели на основе витых пар (twisted pairs) различных категорий.
3. Одно- и многорежимные (одно- и многомодовые) оптоволоконные кабели (fiber равно fibre).
4. Эфир (ether).
5. Телефонные пары (phone pairs).
Где: 1, 2, 5 -- «медь» (copper); 3 -- «оптика» (optics); 1, 2, 3, 5 -- проводные (wired) СрПД; 4 -- беспроводные (wireless) СрПД.

Физически проводные СрПД выражаются в виде отдельных *проводов* (wires), *кабелей* (cables) и *шлейфов* (multiconductor cables). В КС в основном применяются различные кабели.

С точки зрения целевой области применения все кабели делятся на:
1. *Кабели для внешней прокладки* (outdoor cables) -- СПД на улице.
2. *Кабели для внутренней прокладки* (indoor cables) -- СПД в помещениях.
3. *Оконечные кабели* (cords) -- для подключения рабочих мест.

Основные отличительные требования outdoor-кабелей: большее число проводников, высокая прочность, улучшенные электро-магнитные характеристики, влагостойкость, широкий диапазон рабочих температур, наличие дополнительных упрочняющих или гальванически развязывающих вставок.

Indoor-кабели отличаются от outdoor-кабелей меньшими габаритами и массой, большей гибкостью, лучшей пожаростойкостью, при сохранении тех же ключевых достоинств.

Кабели cords являются сравнительно простыми и низкокачественными.

В простейшем случае отдельный провод состоит из *физического проводника* (conductor) и *изоляции* (isolation).

Проводники могут быть *одножильными* (solid) и *многожильными* (stranded).

Отдельно выделяются так называемые *витые* (twisted) провода. Обычно вьются два провода, образующие дифференциальную пару.

Любой *разъем* (connector) состоит из *вилки* (male) и *розетки* (female). Контакты разъемов могут быть либо штыревыми, либо гнездовыми. В настоящее время для соединения разъемов с проводами пайка практически не используется. Следовательно, широко применяются специальные инструменты и почти всегда отсутствуют соответствующие пайке специальные покрытия проводников.

В сегментах КС широко использовались три базовых вида коаксиальных кабелей: с волновым сопротивлением 50 Ω -- RG-8, RG-58, и с волновым сопротивлением 75 Ω -- RG-59.

Коаксиальные outdoor- и indoor-кабели отличаются от cord-кабелей в основном внешней изоляцией.

49) Среды передачи данных на основе коаксиальных кабелей

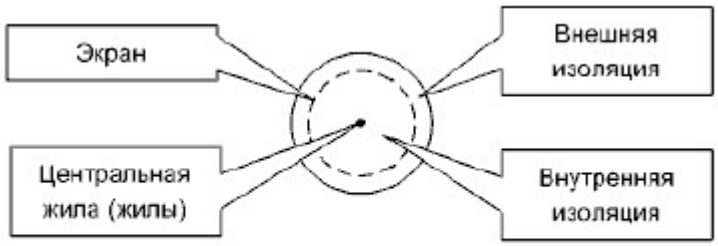


Рисунок -- Структура коаксиального кабеля

Нужна структура я не понмю ее задиктуй

Для формирования сегмента на базе коаксиального кабеля необходимо соответствующее количество BNC-разъемов, Т-разъемов и пара *терминаторов* (terminators), один из которых заземляется.

В сегментах КС широко используются четыре основных вида кабелей на основе витых пар.

Для формирования сегмента на базе коаксиального кабеля необходимо соответствующее количество BNC-разъемов, Т-разъемов и пара *терминаторов* (terminators), один из которых заземляется.

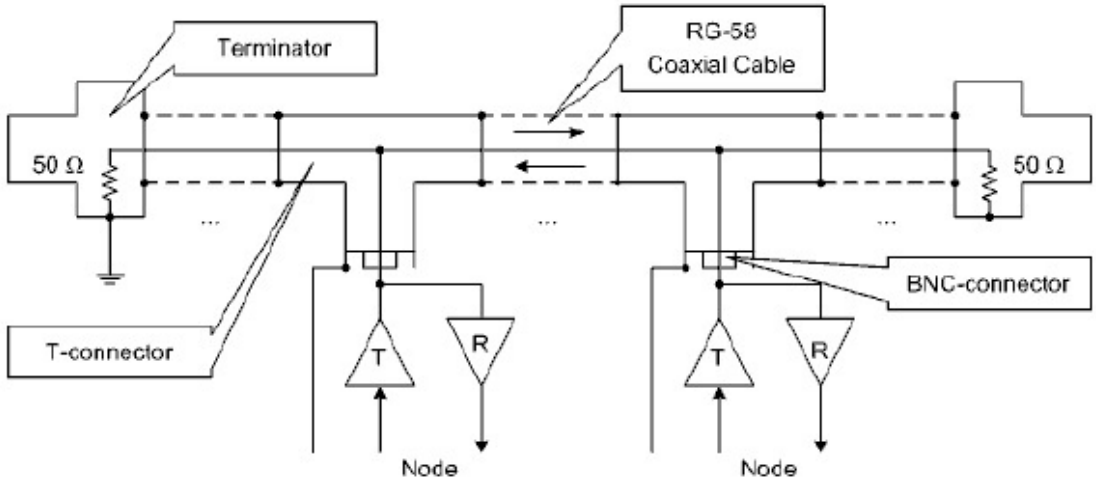


Рисунок -- Пример структуры сегмента с использованием коаксиального кабеля (10BASE2)

50) Среды передачи данных на основе витых пар

В сегментах КС широко используются четыре основных вида кабелей на основе витых пар.

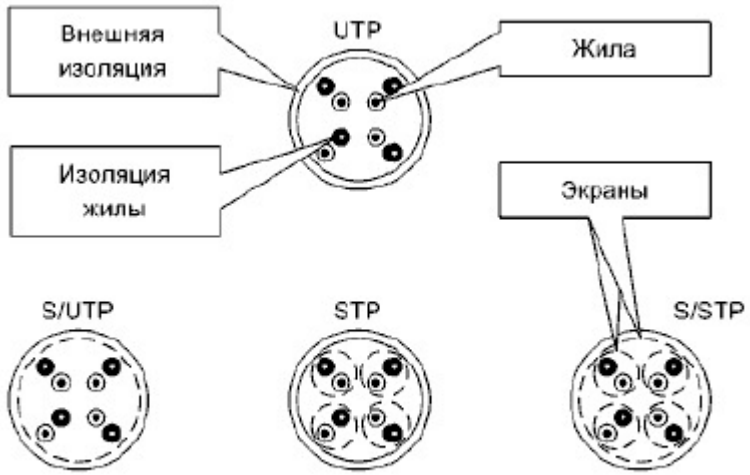


Рисунок -- Структура кабелей на основе витых пар

Где: TP -- Twisted Pair, S -- Shielded, U -- Unshielded, плюс может быть F -- Foiled (если для изготовления экрана применена фольга).

Особо выделяется плоский (flat) кабель для напольной прокладки.

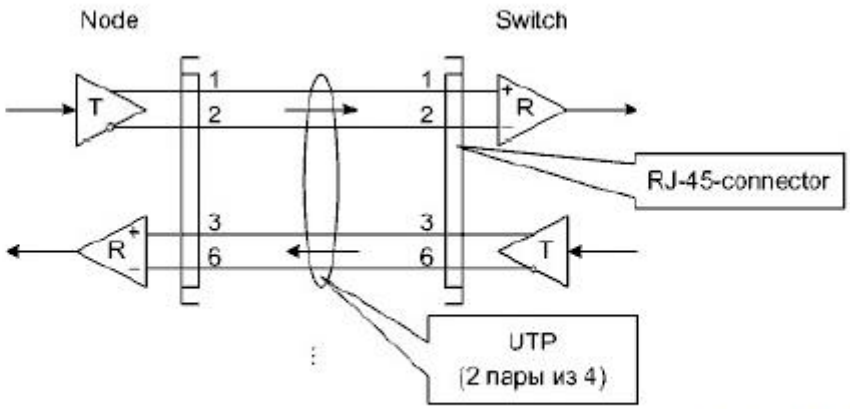


Рисунок -- Пример структуры сегмента с использованием витых пар (100BASE-TX)

В типовых случаях, витыми парами соединяется разноранговое сетевое оборудование. Например, пользовательские станции подключаются к коммуникационному оборудованию, или связывается разноранговое коммуникационное оборудование. При этом используются кабели с «прямой» разводкой. При необходимости, для соединения одноранговой оборудования, например непосредственного связывания двух пользовательских станций, используются кросс-кабели -- пары TD и RD скрещиваются. (Полная аналогия с вариантами соединений ООД и АПД.)

Для подключения кабелей на основе витых пар применяются разъемы RJ-45.

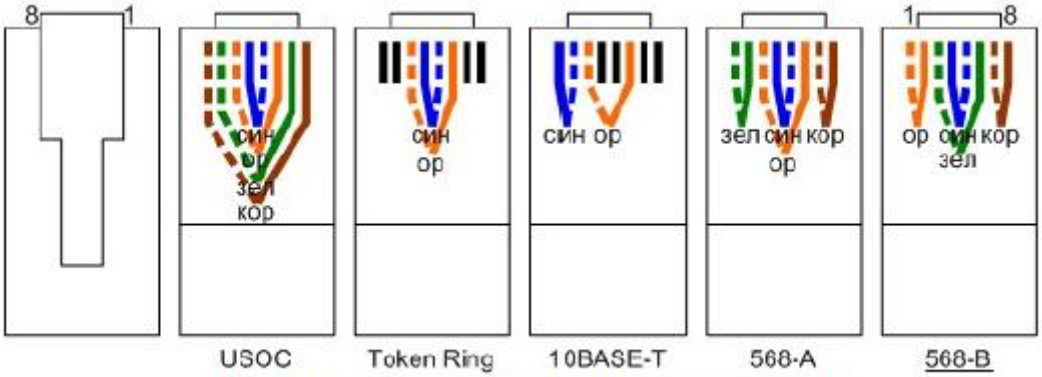


Рисунок -- Стандартная разводка витых пар

(У нас традиционно выбирают вариант 568-B.)

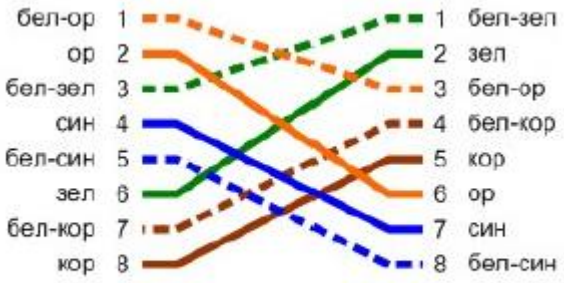


Рисунок -- Кросс-кабель Gigabit Ethernet

51) Среды передачи данных на основе оптоволоконных кабелей

Используемые оптоволоконные кабели отличаются больши'м разнообразием -- следствие относительной дороговизны.

Рабочими компонентами оптоволоконных кабелей являются *световоды* (primary fiber, waveguide, lightpipe), изготовленные из оптоволокна, то есть особого кварцевого стекла. Поскольку оптоволокно очень хрупкое, оно многократно защищается различными способами. Рабочими компонентами самого световода являются *оболочка* (cladding) и *сердцевина* (core).

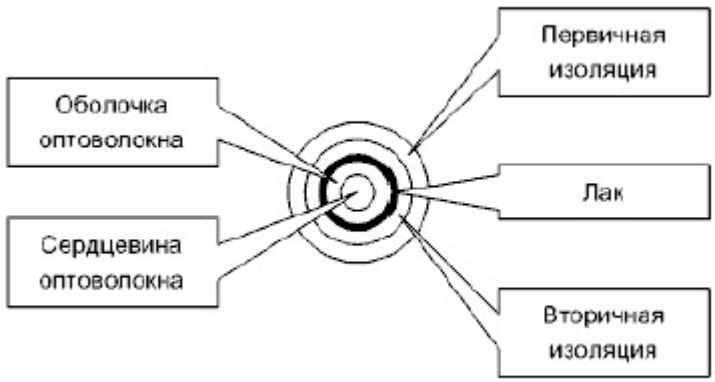


Рисунок -- Структура световода

Стандартами предусмотрены шесть базовых видов световодов: OM1, OM2, OM3, OM4 -- многорежимные; OS1 и OS2 -- однорежимные (по-другому MM1, MM2, MM3, MM4; SM1, SM2 соответственно). Отличаются полосой пропускания и другими техническими характеристиками. Диаметр сердцевины: 62,5 um (американский стандарт) -- OM1; 50 um (европейский стандарт) -- OM2, OM3 и OM4; 9 um -- OS1 и OS2. Диаметр оболочки: 125 um -- для всех видов. Общий же диаметр световода, с учетом буферизации, обычно равен около 250 um (может быть до 1 mm).

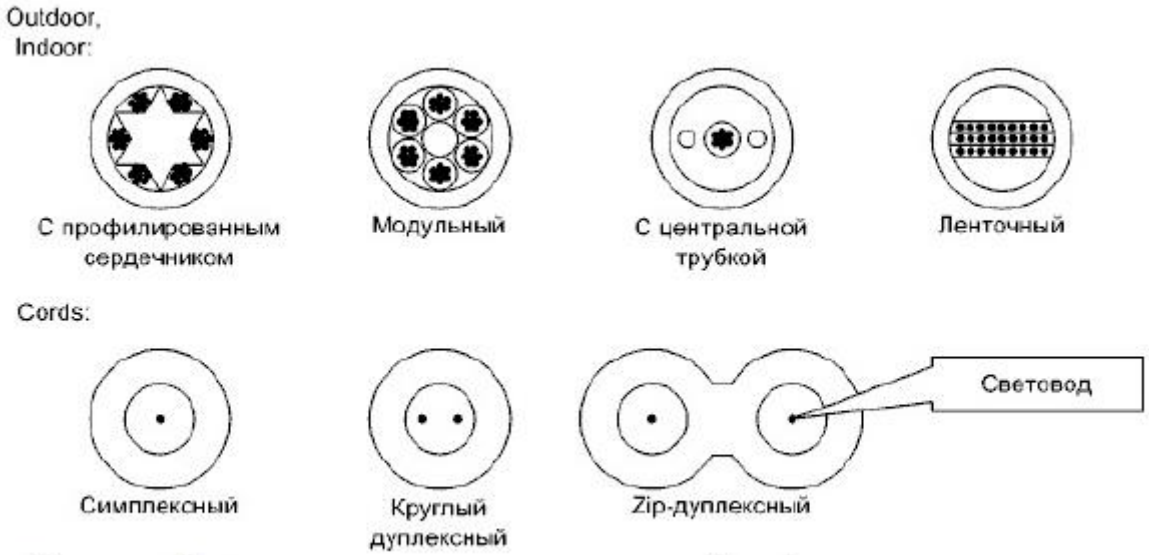


Рисунок -- Примеры структур оптоволоконных кабелей различного назначения

Дополнительно все оптоволоконные кабели делятся на два подтипа:
1. Содержащие металлизированные упрочняющие конструкции или проводники.
2. Полностью диэлектрические.

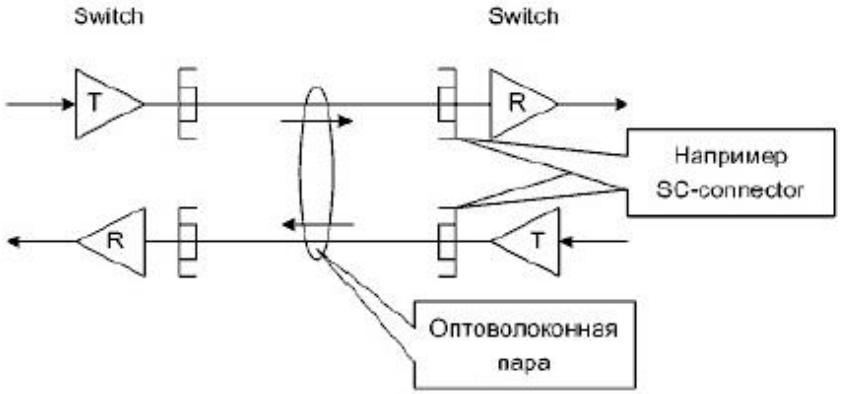


Рисунок -- Пример структуры сегмента с использованием оптоволокна (1000BASE-SX)

Оптоволоконные соединения выполняются двумя способами:
1. Разъемным, причем может быть:
-- контактным;
-- линзовым.
2. Неразъемным, причем может быть:
-- сплавным;
-- механическим.

Оптоволоконные разъемы так же отличаются разнообразием: SC, FC, ST, MIC, E-2000 и другие.

52) Физический уровень Ethernet

Где подчеркнуты ключевые использовавшиеся либо используемые стандарты:
10BASE5 -- «толстый» (thick) коаксиальный кабель 50 Ω, внешние приемопередатчики;
10BASE2 -- «тонкий» (thin) коаксиальный кабель 50 Ω, интегрированные приемопередатчики;
10BASE-T -- две телефонные витые пары;
10BASE-FL -- два многорежимных световода, расстояние до 2 km;
100BASE-TX -- две неэкранированные либо экранированные витые пары категории 5;
100BASE-FX -- два многорежимных световода, с 10BASE-FL совместимости нет;
1000BASE-T -- четыре неэкранированные либо экранированные витые пары категории 5;
1000BASE-SX -- коротковолновые (short wavelength) лазеры, два многорежимных световода, расстояние до 220 m (62,5 um) либо до 550 m (50 um);
1000BASE-LX -- длинноволновые (long wavelength) лазеры, два однорежимных либо многорежимных световода, расстояние до 5 km (однорежимные световоды) либо до 550 m (многорежимные световоды);
10GBASE-T -- четыре неэкранированные либо экранированные витые пары категории 6;
10GBASE-SR -- коротковолновые лазеры, два многорежимных световода, расстояние до 33 -- 400 m в зависимости от вида (то есть качества) световодов;
10GBASE-LR -- длинноволновые лазеры, два однорежимных световода, расстояние до 10 km;
10GBASE-ER -- экстрадлинноволновые (extra long wavelength) лазеры, два однорежимных световода, расстояние до 30 km.

Таблица – Реализации Ethernet

Коаксиальный кабель	Витая пара	Оптоволокно	Другие среды
Ранние реализации Ethernet (около 1 Mbit/s)			
Xerox Ethernet	1BASE5 StarLAN1	--	2BASE-TL (телефонная пара)
Ethernet (10 Mbit/s)			
10BASE5 10BASE2 10BROAD36	10BASE-T StarLAN10	FOIRL 10BASE-FB 10BASE-FL 10BASE-FP	10PASS-TS (телефонная пара)
Fast Ethernet (100 Mbit/s)			
--	100BASE-T4 100BASE-TX 100BASE-T2	100BASE-FX 100BASE-SX 100BASE-BX10 100BASE-LX10	--
Gigabit Ethernet (1 Gbit/s)			
1000BASE-CX (твинаксиальный кабель)	1000BASE-T 1000BASE-TX	1000BASE-SX 1000BASE-LX 1000BASE-LX10 1000BASE-EX 1000BASE-BX10 1000BASE-PX10 1000BASE-PX20 1000BASE-ZX	1000BASE-KX (кластерные шлейфы)
Gigabit Ethernet (10 Gbit/s)			
10GBASE-CX4 (твинаксиальный кабель) 10GBase-CR (SFP+ Direct Attach, твинаксиальный кабель)	10GBASE-T	10GBASE-SR 10GBASE-LR 10GBASE-ER 10GBASE-LX4 10GBASE-LRM 10GBASE-ZR	10GBASE-KX4 (кластерные шлейфы) 10GBASE-KR (кластерные шлейфы) 10GBASE-SW (WAN SONET) 10GBASE-LW (WAN SONET) 10GBASE-EW (WAN SONET)
Gigabit Ethernet (40 Gbit/s)			
40GBASE-CR4 (твинаксиальный кабель)	--	40GBASE-SR4 40GBASE-LR4 40GBASE-FR	40GBASE-KR4 (кластерные шлейфы)
Gigabit Ethernet (100 Gbit/s)			
100GBASE-CR10 (твинаксиальный кабель)	--	100GBASE-SR10 100GBASE-LR4 100GBASE-ER4	--

53) Структурированные кабельные системы и их модели

Структурированная кабельная система (СКС) -- Structured Cabling System (SCS) здания либо сооружения -- это упорядоченная по тем или иным критериям совокупность телекоммуникационных и силовых кабелей, различного сетевого оборудования, а также соответствующих специализированных помещений.

Таблица -- Основные действующие стандарты СКС

	Международные, ISO/IEC	Американские, ANSI/TIA/EIA	Европейские, EN
Проектирование	11801, 24702, 15018, 24764	568-C.0, 568-C.1, 568-C.2, 568-C.3, 568-C.4, 570-C, 607-B, 942-A, 1005-A	50173-1, 50173-2, 50173-3, 50173-4, 50173-5, 50173-6
Монтаж	18010, 14763-2, 14763-3	569-C	50174-1, 50174-2, 50174-3, 50310, 50346
Эксплуатация	14763-1	606-B	--

(568-C является обновлением 568-B и 568-A.)

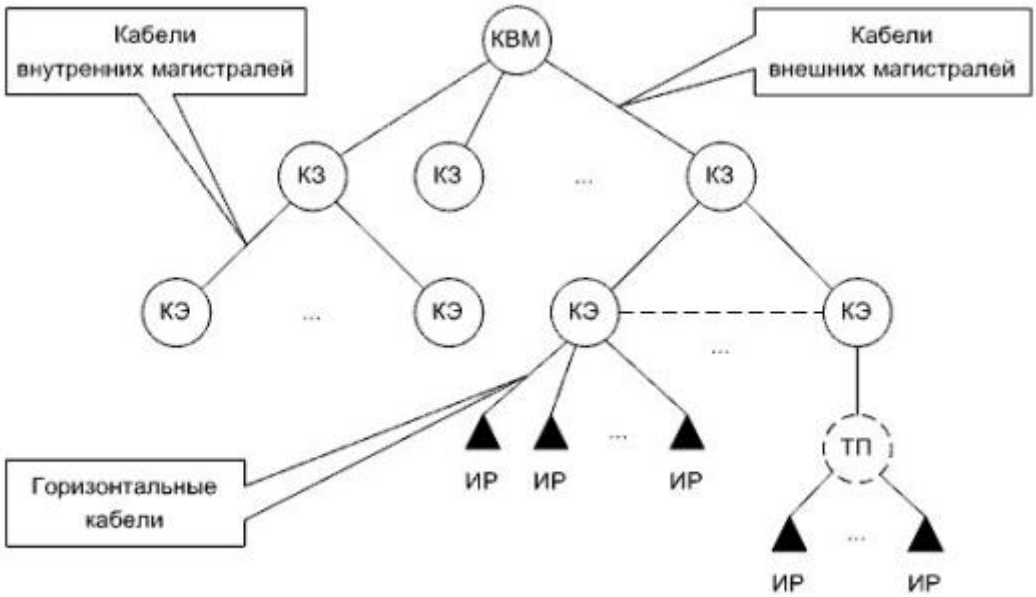
Основой для построения любой СКС является древовидная топология, узлами которой служит сетевое оборудование определенного типа (distributors). В связи с этим, технические помещения СКС (так же distributors) делятся на два типа:

1. *Кроссовые* (telecommunications rooms).
2. *Аппаратные* (equipment rooms).

Аппаратные отличаются от кроссовых тем, что в них, наряду с активным, пассивным, монтажным и вспомогательным сетевым оборудованием, предполагается размещение серверного оборудования.

В общем случае, согласно стандартам ISO/IEC 11801, ANSI/TIA/EIA 568 и EN 50173, СКС включает в себя три подсистемы:

1. *Подсистема внешних магистралей* (main, campus) -- основа для организации связи между компактно расположенными на одной территории зданиями или сооружениями.
2. *Подсистема внутренних магистралей* или, по-другому, *вертикальная* (intermediate, building) -- связывает между собой этажи одного здания или пространственно разнесенные помещения в одном здании.
3. *Горизонтальная подсистема* (horizontal) -- связывает между собой оборудование в пределах этажа или помещения.



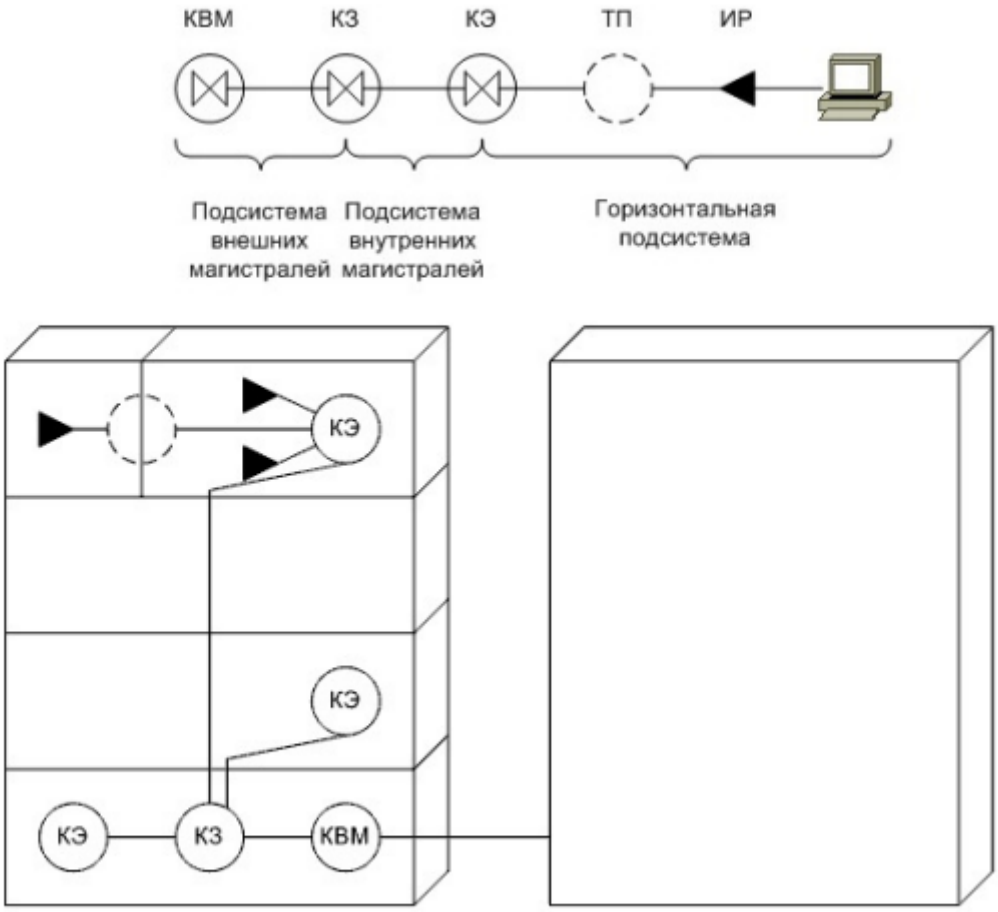


Рисунок -- Модели СКС

Где: КВМ -- кроссовая внешних магистралей, КЗ -- кроссовая здания, КЭ -- кроссовая этажа, ТП -- точка перехода, ИР -- информационная розетка рабочего места. (Вместо кроссовых могут быть аппаратные. Пунктиром обозначены опциональные компоненты. Аббревиатуры -- нестандартные.)

Таким образом, суммарно СКС содержит: кабели и сетевое оборудование всех трех подсистем, плюс точки перехода (consolidation points), плюс информационные розетки.

Иерархическая сетевая модель Cisco хорошо «ложится» на модели СКС.

С точки зрения администрирования СКС выделяются два подхода:
1. *Многоточечный* (*распределенный*).
2. *Одноточечный* (*централизованный*).

Согласно стандарту ANSI/TIA/EIA 606-B выделяются четыре класса администрирования:
Class 1 -- в пределах аппаратной.
Class 2 -- в пределах здания.
Class 3 -- в пределах кампуса.
Class 4 -- за пределами кампуса.

54) Питание и заземление в структурированных кабельных системах

При проектировании СКС важное внимание должно уделяться подключению к силовым сетям, а также организации защиты посредством заземления, зануления или других способов.

Заземление необходимо для:
1. Предотвращения поражения электрическим током людей.
2. Защиты кабельных трактов и сетевого оборудования как от выхода из строя, так и от помех.
3. Обеспечения возможности прохождения сигналов применительно к некоторым видам сетевого оборудования.

Согласно стандарту ANSI/TIA/EIA 607-B, в дополнение к *основному контуру заземления* (grounding electrode) здания либо сооружения, вводится так называемый *телекоммуникационный контур заземления* или, по-другому, *контур рабочего заземления* (telecommunications grounding/bonding).

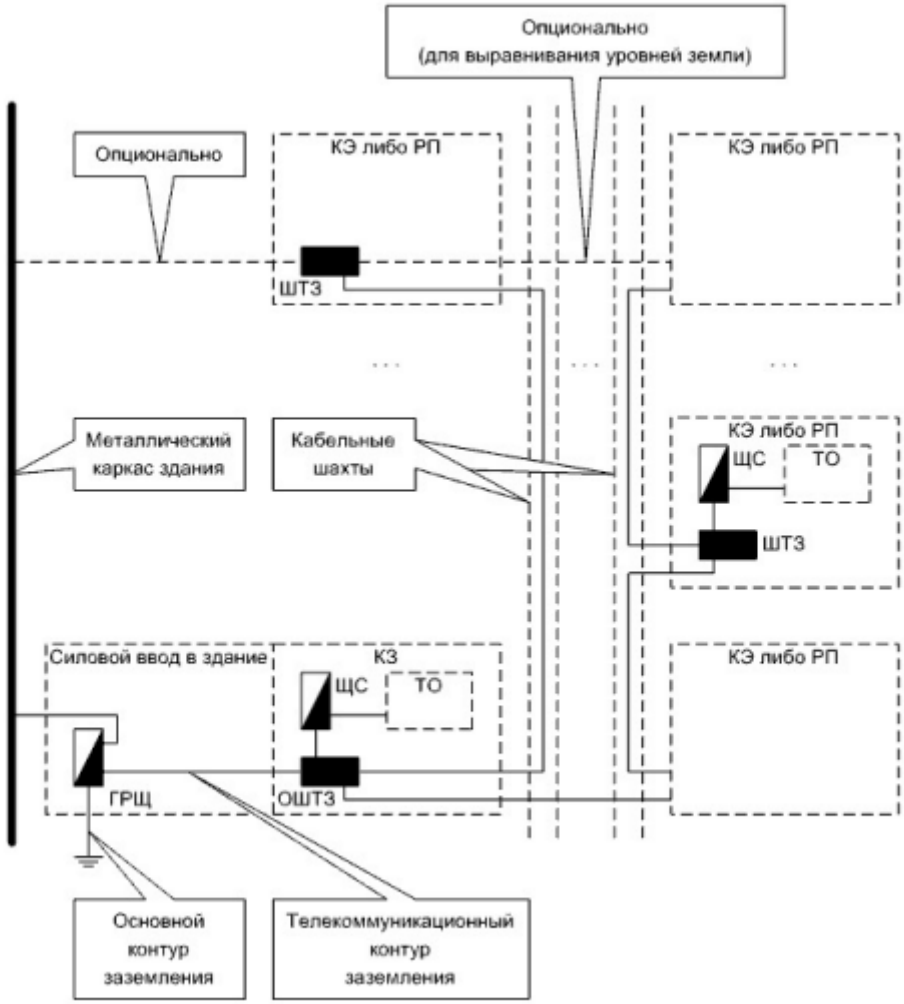


Рисунок -- Модель телекоммуникационного контура заземления

Где: ГРЩ -- главный распределительный щит здания, ЩС -- щит силовой (может быть щит этажный и так далее), ШТЗ -- шина телекоммуникационного заземления, ОШТЗ -- основная ШТЗ, РП -- рабочее помещение, ТО -- телекоммуникационное оборудование.
(Нарисовано с учетом отечественных особенностей. Аббревиатуры кроме ГРЩ и ЩС -- нестандартные.)

Рекомендации стандартов по заземлению экранов кабелей (касается и витых пар):
1. В аппаратных и кроссовых экраны должны заземляться по возможности на телекоммуникационный контур.
2. Экраны вертикальной подсистемы должны заземляться с обоих концов -- в аппаратных или кроссовых.
3. Экраны горизонтальной подсистемы достаточно заземлять с одного конца -- по возможности в аппаратных или кроссовых.

55) Пожарная безопасность структурированных кабельных систем

Согласно американским стандартам NEC (National Electrical Code) предусмотрены четыре уровня сертификации пожарной безопасности кабельных систем (первый уровень -- высший):
1. Plenum -- сюда относят кабели, которые можно без каких-либо ограничений прокладывать в так называемых plenum-полостях (существует приток воздуха, достаточный для постоянного горения).
2. Riser -- сюда относят кабели, которые можно прокладывать в кабельных шахтах (например, вертикальных стояках зданий).
3. General purpose -- сюда относят кабели, которые можно без дополнительной защиты прокладывать везде, кроме plenum-полостей и кабельных шахт.
4. Residential (limited use) -- сюда относят кабели, на прокладку которых наложены специфические ограничения (например, только для жилых помещений).

56) Технология PoE

Относительно недавно производители сетевого оборудования стали разрабатывать технологии, позволяющие запитывать относительно маломощные Ethernet-устройства (например, коммутаторы или точки доступа) через информационные кабели (на основе витых пар), -- технологии под общим названием PoE (Power over Ethernet).

Постепенно были введены два общепромышленных стандарта: 802.3af и 802.3at. Но до сих пор многие производители используют собственные проприетарные технологии. Примерами могут служить Cisco Universal Power over Ethernet (UPOE) (до 802.3af была еще технология Inline Power), Microsemi PowerDsine (ряд производителей), Passive PoE (ряд производителей).

В структуру PoE-системы входит ряд блоков. PSE (Power Sourcing Equipment) вводит питающее напряжение в кабель. PD (Powered Device) питается от этого напряжения. PSE может располагаться либо на конце (одном из двух) кабеля (endspan), то есть быть интегрированным в соответствующее сетевое устройство (как правило, мощный коммутатор, подключенный к силовой сети напрямую), либо «вклиниваться» в кабель (midspan), то есть быть внешним *PoE-инжектором* (PoE injector). Иногда PoE используется и для запитывания «небольших» PD, PoE не поддерживающих, -- со стороны PD в кабель «вклинивается» PoE-DC-адаптер.

Таблица -- Сравнение технологий PoE

	802.3af (PoE) (802.3at тип 1)	802.3at (PoE+) (802.3at тип 2)	Cisco (UPOE)
Максимальный выходной ток PSE	0,35 A	0,6 A	1 A
Выходное напряжение PSE	44 -- 57 V	50 -- 57 V	44 -- 57 V
Максимальный ток, потребляемый PD	0,35 A	0,6 A	1 A
Напряжение питания PD	37 -- 57 V	47 -- 57 V	37 -- 57 V
Максимальная мощность PSE	15,4 W	30 W	60 W
Максимальная мощность PD	12,95 W	25,5 W	51 W
Число задействованных витых пар	2	2	4

Исходя из потребляемой мощности, PDs делят на пять стандартных классов:
Class 0. 0,44 -- 12,95 W -- по умолчанию.
Class 1. 0,44 -- 3,84 W -- очень малой мощности.
Class 2. 3,84 -- 6,49 W -- малой мощности.
Class 3. 6,49 -- 12,95 W -- средней мощности.
Class 4. 12,95 -- 25,5 W -- большой мощности.