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.

**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) Data Center Networks КС - центров обработки данных

7) Home Networks – Домашние КС

8) Industrial Networks - промышленные КС

C 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.

Home, 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)

C 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. Мы имеем дело с хост-терминальной моделью.

В случае слабосвязанной КС подразумевают наличие сервера (server) с одной стороны и клиента (client) - с другой. 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

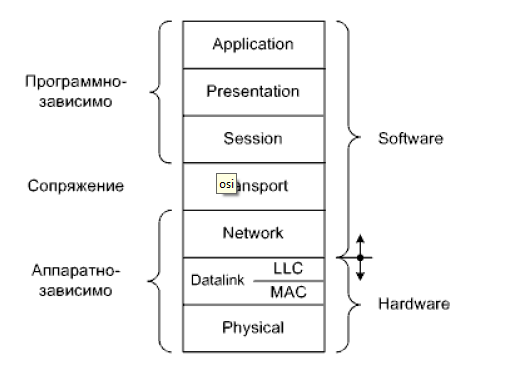
3) Gigabit Ethernet – 1, 10, 100, 40, 25, Gbit/s и Multigigabit

**4) Наиболее распространенные модели компьютерных сетей**

Из всех моделей KC наиболее фундаментальной является открытая

модель взаимодействия систем - Open System Interconnection (OSI) разработанная ISO.

Модель включает 7 уровней.



На вершине иерархии находится человек, но абонентами КС являются

взаимодействующие программы.

Взаимодействие в рамках модели OSI может быть «вертикальным» и

«горизонтальным»:

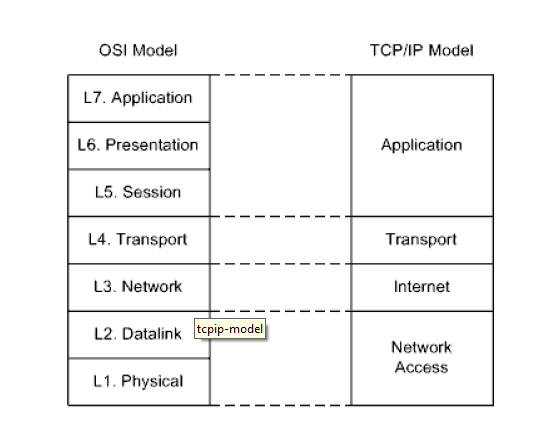
1) Интерфейс - это правила взаимодействия между пространственно-совмещенными соседними уровнями модели OSI.

2) Протокол - это правила взаимодействия между пространственно- разнесенными одинаковыми уровнями модели OSI.

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

Ещё одна модель связана с семейством протоколов TCP/IP

Её сопоставление с моделью OSI на след рисунке.



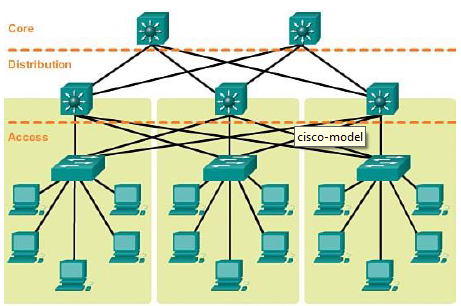
Компания Cisco на основе многолетнего опыта проектирования сетей разработала собственную иерархическую сетевую модель (Cisco hierarhical network model), которую рекомендует использовать в корпоративных сетях разного масштаба.

Модель включает 3 уровня:

1) Access – доступ

2) Distribution (иногда aggregation) – распределение

3) Core – ядро



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

Уровень распределения предназначен для обеспечения взаимодействия в пределах групп пользователей. Особое внимание здесь уделяют резервированию соединений.

Уровень ядра предназначен для обеспечения высокоскоростной связи между относительно удаленными группами пользователей. Особое внимание здесь уделяют характеристикам трафика.

На всех уровнях значительное место отведено разграничению трафика с целями защиты пользователей друг от друга и защиты КС от пользователей.

При этом всем, технологии могут быть различными. Догм нет. 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.

Специфическими понятиями физического уровня являются:

1) Среда

2) Разъём (физический порт)

3) Несущая (частота)

4) Модуляция

5) Сигнал

Фундаментальная задача физического уровня заключается в передаче сигнала.

Для передачи сигнала используется несущая и её модуляция.

Несущая – частота гармонических электрических (электромагнитных) колебаний, служащих переносчиком информации при ее передаче посредством модуляции этих колебаний сигналами, соответствующими передаваемому сообщению.

Модуляция – процесс изменения одного или нескольких параметров несущего высокочастотного колебания в соответствии с изменением параметров передаваемого 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**

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

Третий 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

**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.

**9) Прикладной уровень и уровень представления модели OSI**

Прикладной (application) уровень призван решать конкретные пользовательские задачи с помощью КС. Предназначен для решения таких задач, как:

1) Пересылка файлов между компьютерами

2) Пересылка электронных писем

3) Поддержка удаленных текстовых и графических терминалов, в том числе для администрирования

4) Пересылка мультимедийных документов

5) Обмен мгновенными сообщениями

6) Совместная разработка чего-либо

Специфических понятий прикладного уровня великое множество, и они зависят от решаемых задач.

Уровень представления (presentation) позволяет адаптировать прикладную информацию в форму, приемлемую для передачи по КС, то есть является прослойкой между программами и транспортом.

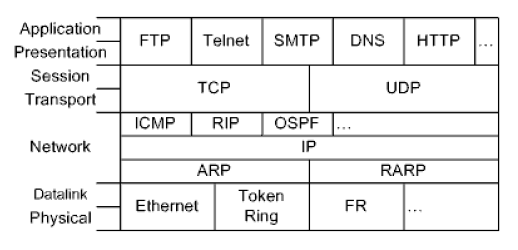
Основными задачами уровня представления являются:

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.

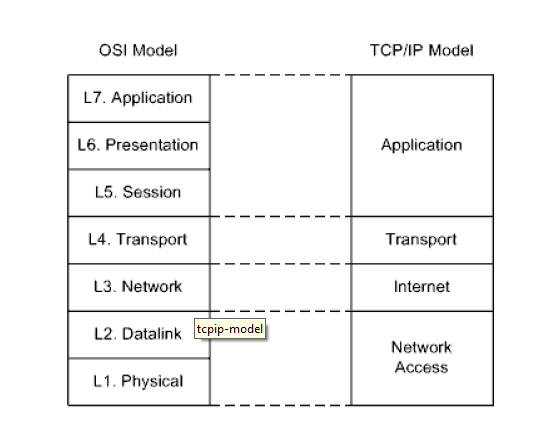
Поскольку обычно уровень представления «привязан» к прикладному уровню, в реализациях эти уровни часто совмещаются.

**10) Семейство протоколов TCP/IP**



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

Также семейство протоколов TCP/IP имеет одноименная модель.



**11) Эволюция COM-портов и их место в современных ПК**

В развитии COM-порта ПК можно выделить следующие основные этапы:

1) В семидесятые годы XX века, компания Intel разработала два контроллера последовательного порта. Один из них, 8250, получил название UART. Второй, 8251, получил название USART. Эти контроллеры были рассчитаны на подключение по шине X-Bus и поэтому без труда были перенесены в первые ПК на базе процессора 8086 и его модификаций с тогда наиболее распространенной системной шиной ISA. Микросхема UART либо USART устанавливалась на плату специального адаптера и подключались к материнской плате ПК посредством разъема системной шины. В это же время возникла традиция устанавливать последовательные порты парами (COM1 и COM2).

2) Времена доминирования процессоров 80286. На первом этапе происходило распространение и развитие самих контроллеров. В СССР был создан аналог 8251 под названием КР580ВВ51А. На Западе же, наоборот, развитие получила микросхема 8250. Апофеозом достаточно быстрого усовершенствования 8250 стали несколько UART, среди которых следует выделить 16550 разработки National Semiconductor. Именно эта микросхема стала де факто стандартной на длительное время (архитектурная совместимость сохраняется вплоть до настоящего времени). 16550 имеет два основных преимущества перед 8250:

­- более высокая пропускная способность последовательного интерфейса (максимальная стандартная пропускная способность увеличена с 9600 baud до 115200 baud);

- возможность буферизации (две очереди FIFO по 16 байт -- на стороне передатчика и на стороне приемника).

3) В дальнейшем интеграционные процессы привели к появлению так называемых мультикарт -- подключаемых посредством разъема системной шины (по-прежнему обычно ISA) плат расширения с интегрированными контроллерами: последовательного порта (2x16550), параллельного порта, игрового порта, НГМД и НЖМД. Причем все эти функции сочетались в одной БИС с типичным названием Multi I/O.

4) Для ПК на базе поздних Intel486 уже была характерна интеграция чипа Multi I/O на материнскую плату.

5) Во времена процессоров Pentium сформировалась действительная до сих пор базовая крупноблочная структура материнской платы ПК, состоящая из четырех основных БИС. Контроллеры последовательного порта (по той же схеме 2x16550) в составе интегрированной периферии были перенесены и в эту структуру. С этого момента реализации последовательных портов не претерпели никаких изменений.

6) В настоящее время (приблизительно с 2005 года) традиционный последовательный интерфейс ПК считают устаревшим (legacy), часто исключают из состава интегрированной периферии.

**12) Структура COM-портов ПК**

На аппаратном уровне приемник и передатчик работают параллельно т.е. по отдельным физическим цепям полностью независимо друг от друга. Для физического подключения по стандарту RS-232 используют девятиконтактные разъемы DE-9. Передатчик и приемник COM-порта представляют из себя сдвиговые регистры: данные, предварительно записанные в регистр передатчика параллельно, последовательно сдвигаются в линию под воздействием тактовых импульсов.

В Структуру com-порта входит сама шина данных, приёмный и передающий буффер, программируемые бод-генератор для тактирования, множество регистров (информационные, служебные, управляющие).

**13) Цепи RS-232 и их использование**

Традиционное назначение цифровых цепей RS-232:

1) SOUT (Serial Output) - выход передатчика;

2) SIN (Serial Input) - вход приемника;

3) RTS (Request to Send) - сигнал-запрос от UART к модему о передаче байта;

4) CTS (Clear to Send) - сигнал-подтверждение от модема к UART о готовности принять байт для передачи;

5) DSR (Data Set Ready) - сигнал от модема к UART о готовности к взаимодействию;

6) DTR (Data Terminal Ready) - сигнал от UART к модему о готовности к взаимодействию;

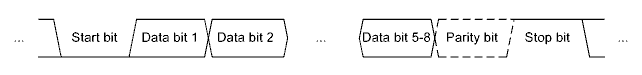
7) DCD (Data Carrier Detect) - сигнал от модема к UART об обнаружении данных;

8) RI (Ring Indicator) - сигнал от модема к UART об обнаружении входящего телефонного звонка.

Служебные цепи RS-232 позволяют организовать контроль информационного потока (flow control). Например, это позволяет избегать переполнения приемника, приостанавливая «быстрый» передатчик. Следует отметить, что практически все служебные цепи напрямую связаны с соответствующими регистрами управления и состояния UART 16550, то есть «открыты» для программирования. Следовательно, алгоритмы контроля реализуют программно и закладывают, например, в драйверы операционных систем.

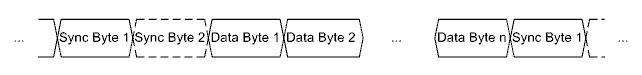
**14) Ассинхронный режим работы com-порта**

Атомарной, то есть минимальной неделимой единицей, с которой работает как UART, так и USART, является байт, причем один байт не обязательно равен восьми битам и может содержать от 5 до 8 битов. По умолчанию линия находится в состоянии логической единицы. При наличии байта для передачи передатчик переводит линию в состояние логического нуля, то есть передает старт-бит, что говорит приемнику о том, что на следующем такте нужно «ловить» первый информационный бит. Стоп-бит необходим для того, чтобы после передачи информационной последовательности гарантированно вернуть линию в исходное, то есть единичное состояние. Старт-бит всегда один, а стоп-битов может быть один, полтора либо два. Для проверки целостности информационной части, если эта проверка включена, за информационной частью вставляется бит паритета. При этом действует правило дополнения. Например, если включена проверка единиц на четность (even), то бит паритета формируется таким образом, чтобы общее число единиц (в информационной части плюс бит паритета) было четным. Либо, если включена проверка нулей на нечетность (odd), то общее количество нулей должно быть нечетным. Ошибки отслеживаются приемником.



**15) Синхронный режим работы com-порта**

При «простое» передатчик «заполняет» линию специальными байтами синхронизации, тем самым настраивая приемник. Все поступающие байты передаются без «обрамления». Как и в асинхронном режиме, ошибки отслеживаются приемником. При обнаружении ошибок, а при длительной непрерывной передаче из-за накапливающихся фазовых сдвигов они неизбежно возникают, приемник должен каким-либо дополнительным способом (так как текущую цепь задействовать невозможно) приостановить передатчик, чтобы тот вновь «заполнил» линию байтами синхронизации.



**16) Тактирование com-порта**

Тактирование сдвиговых регистров UART 16550 осуществляется с помощью встроенного программируемого бод-генератора (baud generator) (тактирование некоторых первых реализаций UART осуществлялось таймером). Бод-генератор представляет собой программируемый делитель частоты. Выходная частота бод-генератора Fout определяется по формуле:

Fout = Fin / (16 DL),

где Fin -- входная частота, DL -- шестнадцатибитная константа, старшая и младшая части которой хранятся в двух регистрах UART (DLL и DLM).

На вход бод-генератора поступает меандр, получаемый от внешнего кварцевого резонатора, который тактирует и сам автомат UART. Частота тактирования автомата UART по крайней мере в 16 раз больше Fout. Следует учитывать, что, для того чтобы правильно рассчитать DL, необходимо точно знать Fin. Вполне естественно, что на разных материнских платах используют разные микросхемы и разные кварцевые резонаторы. Применительно к современным Super I/O, эта частота может достигать 48 MHz, то есть совпадать с частотой синхронизации Super I/O. Но, за счет еще одного деления частоты (при загрузке ПК BIOS конфигурирует UART инициализируя соответствующие регистры конфигурационного пространства Super I/O), как правило, Fin приводится к классическому значению 1,843 MHz. При этом, если DL = 1 (нулевое значение DL использовать крайне не рекомендуется), то Fout = 115200 Hz.

**18) Стандарты, близкие к RS-232**

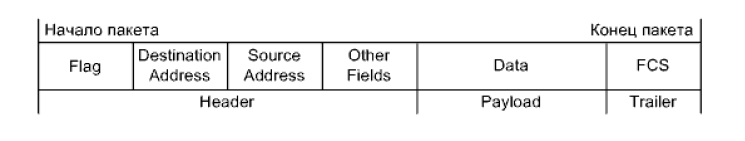
В результате, закономерным продолжением стандарта RS-232 стали два стандарта: RS-422 и RS-485. При этом RS-422 можно рассматривать как промежуточный на пути к RS-485 стандарт. Сравнение основных характеристик упомянутых стандартов приведено в таблице.

Основные сравнительные характеристики RS-232, RS-422 и RS-485

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | RS-232 | RS-422 | RS-485 |
| Способ передачи сигнала | Изменение потенциала относительно земли | Дифференциальная пара | Дифференциальная пара |
| Направление передачи | Одностороннее, двустороннее | Одностороннее, двустороннее | Одностороннее, двустороннее |
| Максимальное количество передатчиков | 1 | 1 | 32 |
| Максимальное количество приемников | 1 | 10 | 32 |
| Ориентировочная максимальная пропускная способность | 1 Mbit/s | 10 Mbit/s | 10 Mbit/s |
| Ориентировочное максимальное расстояние | 15 m | 1200 m | 1200 m |

Для передачи данных посредством интерфейса RS-485 требуются специальные трансиверы с гальванической развязкой, позволяющие реализовать дифференциальный способ передачи сигнала. Гальваническая развязка может быть либо трансформаторной, либо оптронной. О СрПД в стандарте не сказано, но, как правило, используют витую пару (twisted pair) и разъемы типа RJ.

**19) Структура типового пакета компьютерной сети**



Назначение полей:

Flag - флаг начала пакета

Destination Address - адрес назначения

Source Address - адрес источника

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).

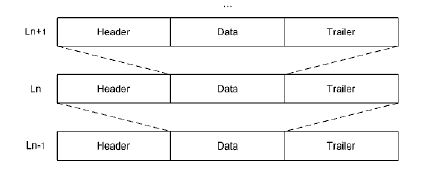
Обычно в байт ориентированных реализациях длина пакета кратна восьми битам, то есть пакет состоит из так называемых октетов.

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.

**20) Инкапсуляция и ее проявления в компьютерных сетях**

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.

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.

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

**21) Бит-стаффинг**

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).

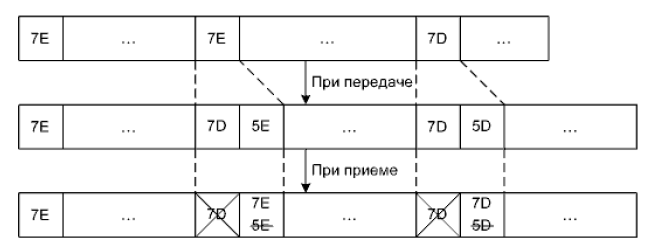


На передающей стороне после нуля и шести единиц всегда вставляется седьмая единица, а на принимающей стороне единица после нуля и шести единиц всегда удаляется.

Бит-стаффинг используется при задействовании синхронных СрПД.

**22) Байт-стаффинг**

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.

**23) 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”

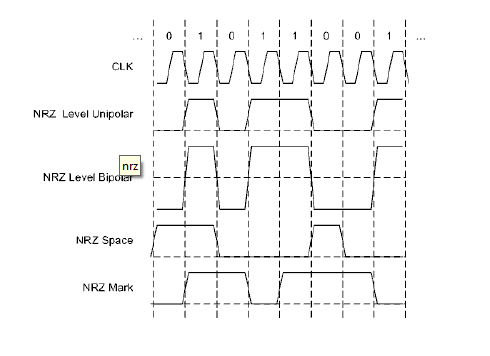
5) Наличие самосинхронизации

6) Наличие перестановки или подмены битов

**24) Линейные коды без возврата к нулю и с возвратом к нулю**

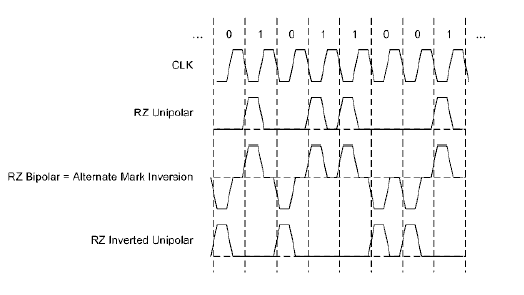
NRZ коды выражаются в изменении уровней между тактами. В простейших случаях, логические уровни в исходной последовательности не преобразуются совсем либо инвертируются. Более сложными случаями являются space и mark. При space-варианте ноль во входной последовательности кодируется сменой текущего уровня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



Примеры технологий с применением NRZ: RS-232, USB, HDLC.

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.

 Примеры технологий с применением RZ: IrDA.

**25) 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

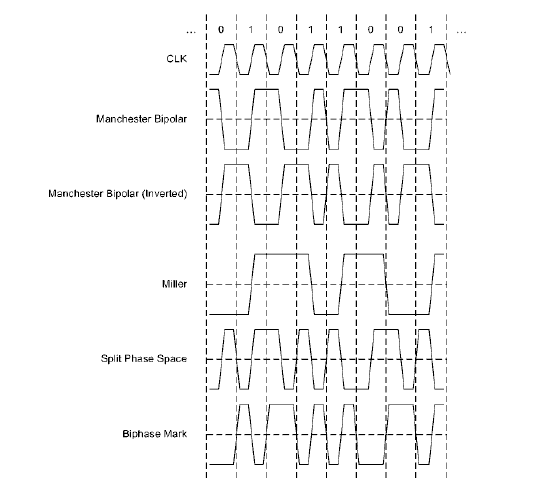
2) Split Phase код

3) Biphase код

Согласно коду Миллера (Miller), ноль соответствует отсутствию перехода во время такта, единица соответствует переходу во время такта, плюс между двумя нулями всегда выполняется смена уровня.

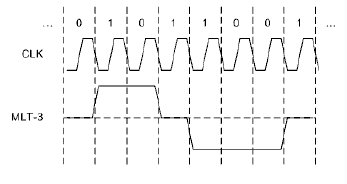
Согласно коду Split Phase учитывается направление предыдущего перехода. При space-варианте ноль соответствует переходу во время такта в направлении, противоположном направлению предыдущего перехода, единица соответствуют переходу во время такта в направлении, совпадающем с направлением предыдущего перехода. При mark-варианте «роли» нулей и единиц из входной последовательности инвертируются.

Согласно коду Biphase, кроме возможных переходов во время тактов, всегда выполняется смена уровня между тактами. При space-варианте ноль соответствуют переходу во время такта, единица соответствуют отсутствию перехода во время такта. При mark-варианте «роли» нулей и единиц из входной последовательности инвертируются.



Примеры технологий с применением манчестерских кодов: Ethernet, Token Ring, некоторые IR-технологии.

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).



Примеры технологий с применением MLT-кодов: Fast Ethernet, FDDI.

**26) 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).

**27) 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.

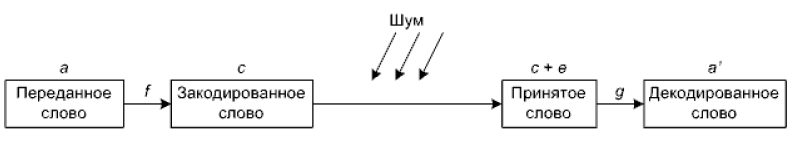
Для практического применения полей Галуа в компьютерных системах необходимо перейти от скалярного представления к векторному. Расширенное поле Галуа GF(pn) можно рассматривать как векторное пространство, где простое число p является характеристикой поля и соответствует количеству состояний разряда вектора, 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(2n)

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(pn). 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 x4+1 (число 17) не подходит, так как x4 + 1 [эквивалентно] (x2+1)2 (mod 2)). Возведение в степень обладает цикличностью.

**28) 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.



Передаваемое сообщение разбивается на блоки фиксированного размера a из k битов a1, a2, … ak. Кодер выполняет функцию f называемую схемой кодирования и тем самым преобразует вектор, а в вектор c из n k битов с1, с2, … сn 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(2k) в конечное поле GF(2n). При более простых схемах кодирования, в кодовом слове сначала располагаются биты входного сообщения, называемые информационными, а за ними дополнительные биты, называемые проверочными a1, a2, … ak, битов с1, с2, … сn. В более сложных случаях проверочные биты чередуются с информационными.

**29) 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:

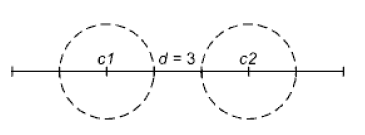
1) Сколько бинарных ошибок код должен обнаруживать.

2) Сколько бинарных ошибок код должен исправлять.

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* = 2*m* – 1, *m* >=2 с проверочной матрицей H размером m × (2m – 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 a0, a1 … an-1 является кодовым словом, то и его циклический сдвиг an-1, a0 … an-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.

**30) 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.

**31) 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.

**32) 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 смножественным 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:

1) Point-to-multipoint – иногда.

2) Multipoint-to-point – очень редко.

Менее двух станций в сегменте быть не может.

Если топологически классифицировать аппаратные технологии, то есть еще два ракурса:

1) Физическая топология – отражает физические связи между устройствами.

2) Логическая топология – отражает лигические связи между устройствами.

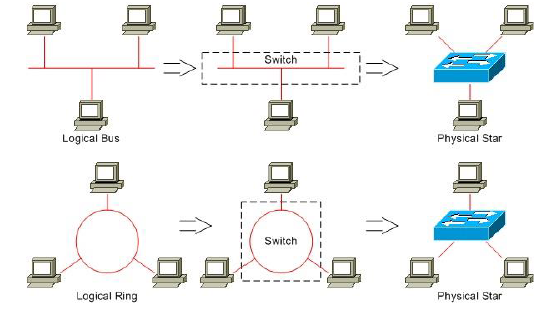
Характерными топологиями ЛКС (LAN) являются:

1) Шина

2) Кольцо

3) Звезда

Часто логическая топология не совпадает с физической. Примеры несоответствий между физ. и логич. топологиями:



**33) Логические и физические топологии WAN и RAS**

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 смножественным 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:

1) Point-to-multipoint – иногда.

2) Multipoint-to-point – очень редко.

Менее двух станций в сегменте быть не может.

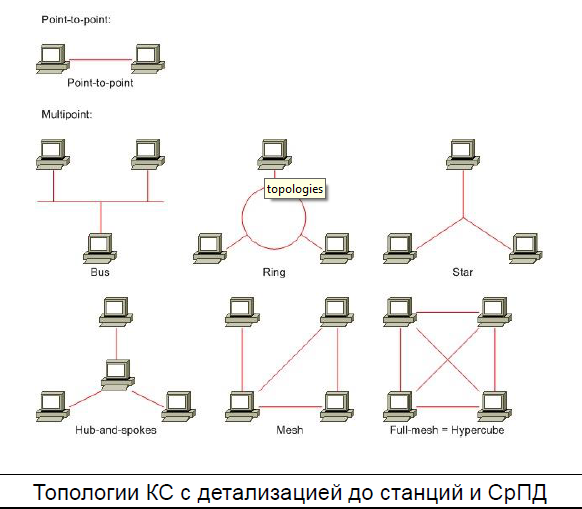
Характерными топологиями ГКС (WAN) являются:

1) Сеть (произвольно связанная) (mesh)

2) Ступица со спицами (hub-and-spokes)

3) Полносвязная сеть (full-mesh)

Характерной RAS(Remote access server) 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.



**34) Особенности случайных методов доступа к моноканалу**

Проблема заключается в «столкновениях» конкурирующих передатчиков. 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).

Классическим способом защиты оборудования от коллизий является 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.

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 гарантированно 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.

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.

**35) 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

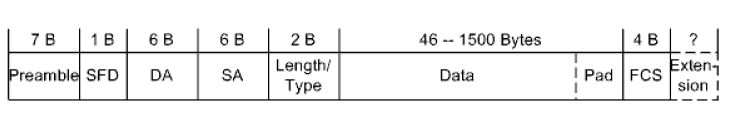
0 <= r<= 2k, где k = min (n, 10), где n номер попытки.

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. Cлот-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.

**36) 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).

В качестве контрольного кода используется код CRC.

**37) 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).

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

0 <= Random <= CW, где CW – так называемое окно состязаний (contention window), CWmin <= CW <= CWmax, и берётся из ряда 7, 15, 31 (2^n – 1). Типичные значения: CWmin =15, CWmax=1023.

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

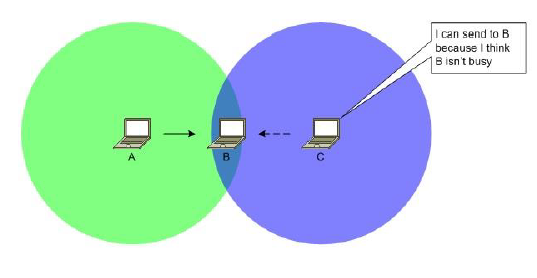
Для беспроводных каналов свойственны две проблемы, которые получили следующие названия:

1) Hidden node problem - проблема скрытой станции

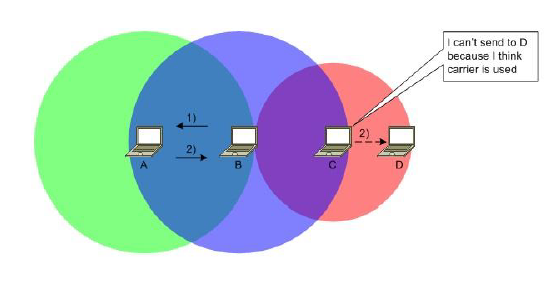
2) Exposed node problem - проблема доступной станции

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 C может ошибочно начать передачу станции B, так как не может «услышать» что станция A уже передает станции B (станция A «скрыта» от станции C)

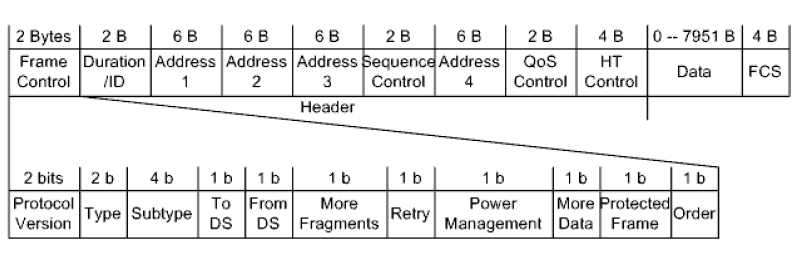


Проблему доступной станции можно сформулировать так станция C, зная о взаимодействии станций A и B, не может передать станции D во время пассивности станции B, а могла бы, поскольку считает канал занятым ошибочно (станция C «доступна» для станции D)

****

Частично решить проблемы помогает опциональное расширение RTS/CTS.

**38) Кадры Wi-Fi**

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: 00b -- Management -- управление, 01b -- Control -- контроль 10b -- Data -- данные, 11b -- 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 режиме энергосбережения станции).

10. Protected Frame -- флаг защищенности кадра (шифрования).

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 Address) – адрес станции передатчика (непосредственного)
5. RA (Receiving station Address) – адрес станции приемника (непосредственного)

**39) 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.

**40) Алгоритм Token Ring**

С точки зрения детерминированных методов доступа к моноканалу наиболее наглядным примером является классический алгоритм, описанный в стандарте Token Ring (IEEE 802.5).

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 такую управляющую станцию обобщенно называют 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 нескольких 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).

В стандарте предусмотрены четыре вида передаваемых последовательностей:

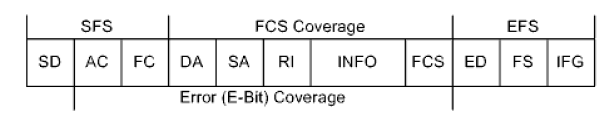
1) Token – маркер.

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. Формат кадров Token Ring и назначение основных полей.



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)

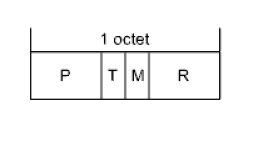
8) FCS (Frame Check Sequence) - контрольная сумма

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:

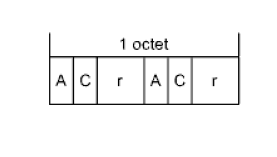
1) P (Priority bits) - текущий уровень приоритета

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 основан 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 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.

Соблюдение перечисленных выше правил гарантирует, что любая станция рано или поздно дождется возможности передать любой кадр.

Кроме всего прочего, в Token Ring заложено несколько механизмов обеспечения надежности, включая авто переконфигурирование и сигнализацию об ошибках.

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

**41) 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)

Скорость: 2,5 Mbit/s.

Логическая топология: однонаправленное кольцо.

Физическая топология: шина или звезда.

Первая широко используемая в ЛКС технология. На данный момент сильно устарела из-за Ethernet.

2) Технология Token Bus (IEEE 802.4).

Скорость: 1, 5, 10, 20 Mbit/s

Логическая топология: однонаправленное кольцо

Физическая топология: шина

Алгоритм представлял собой адаптацию алгоритма Token Ring к шинной топологии. Почти не применяли, сильно устарела.

3) Технология FDDI (Fiber Distributed Data Interface), CDDI (Copper Distributed Data Interface).

FDDI использовался для оптических СрПД, а CDDI – для электрических.

Скорость: 100 Mbit/s, 200 Mbit/s

Логическая топология: однонаправленное кольцо с резервированием.

Физическая топология: двойное кольцо, к которому с помощью дополнительного сетевого оборудования могут подключаться деревья (узлами дерева являются концентраторы, листьями - станции, концентратор-корень включают в двойное кольцо).

Алгоритм представляет собой расширение алгоритма Token Bus. CDDI почти не использовали, FDDI был быстро вытеснен с рынка сетевых технологий после появления более дешевого Fast Ethernet, но ограниченно применяем до сих пор.

4) Технология 100VG-AnyLAN.

Скорость: 100 Mbit/s

Логическая топология: дерево

Физическая топология: дерево (с опциональным резервированием).

Алгоритм представляет собой альтернативу Fast Ethernet (гибрид Ethernet и Token Ring). На технологию возлагали большие надежды, но она была быстро отвергнута рынком и в скорости практически исчезла.

**42) 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.

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.

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.

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

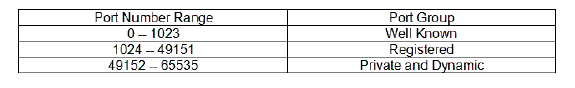
1) Необходимо адресовать подсеть используется адрес подсети (subnet address).

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)

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

Диапазоны программных портов применительно к семейству TCP/IP



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).

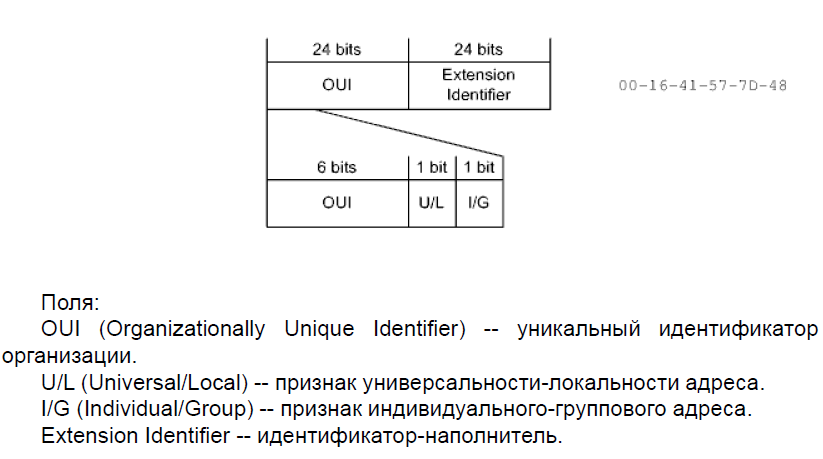
Бродкаст мультикаст и эникаст адреса не могут быть адресами источников, так как отдельно взятый пакет может сгенерировать только одна станция.

Наиболее сложной формой адресации является эникаст адресация. Очевидно, что каждый раз при приеме эникаст пакета должен осуществляться выбор на основе какого-либо критерия. При этом адресуемые станции должны осуществлять выбор в пределах группы сами. Отправившая пакет станция не может принимать участие в алгоритме выбора, она уже сделала свой «выбор» записав в пакет в качестве адреса назначения эникаст адрес. Выбор должен быть сделан заблаговременно, чтобы принимающая станция была готова к поступлению в группу пакета. Примером критерия выбора может служить время задержки. Выбор может осуществляться однократно либо периодически.

**43) 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.

Формат EUI-48:



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)

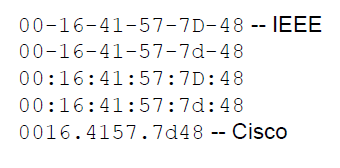
Иногда поле OUI рассматривают как CID (Company ID), что, по большому счету, то же самое зависит от комбинации значений битов U/L и I/G (рассматривают уже как биты X и M соответственно). При так называемом каноническом представлении MAC-адрес сдвигается в канал начиная со старших разрядов.

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:



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.

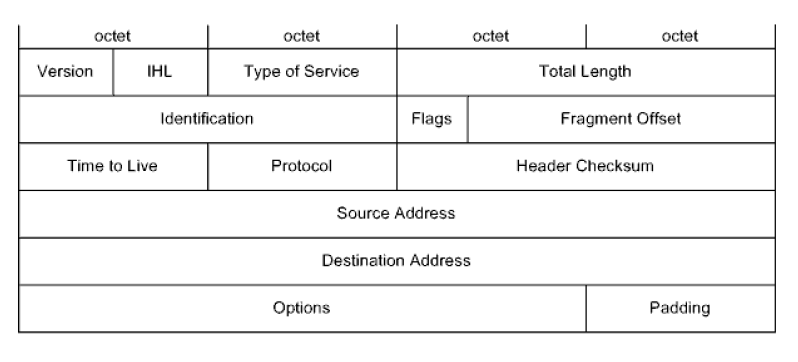
Примеры технологий с применением EUI-48: Ethernet, Wi-Fi, Token Ring

Примеры технологий с применением EUI-64: IPv6, FireWire

**44) 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

**45) Заголовок IPv6**

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



Поля:

1) Version - версия (значение равно 6).

2) Traffic Class - класс трафика (связано с QoS).

3) Flow Label - метка потока (связано с QoS).

4) Payload Length - длина полезной нагрузки (в байтах, аналог поля Total Length).

5) Next Header - селектор следующего заголовка (в том числе, аналог поля Protocol).

6) Hop Limit - ограничитель числа «прыжков» (аналог поля Time to Live).

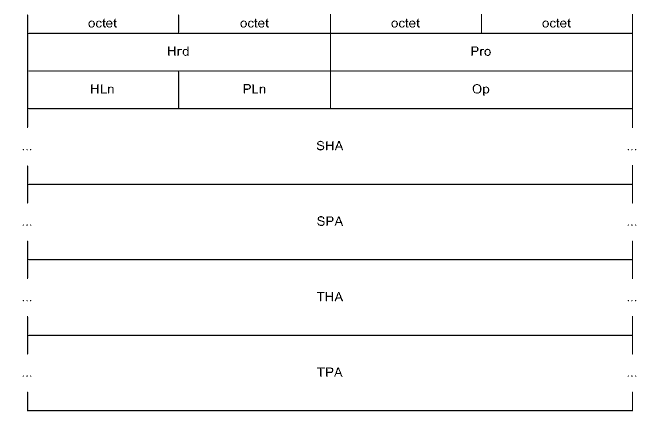
7) Source Address - адрес источника

8) Destination Address - адрес назначения

**46) 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)

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

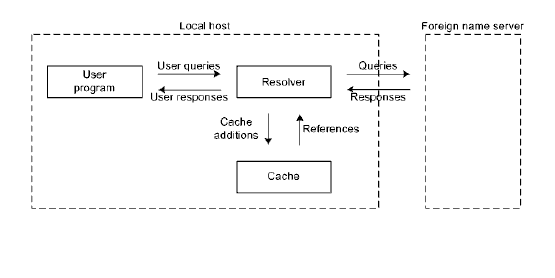
9)TPA (Target Protocol Address) - протокольный адрес запрашиваемой станции

**47) 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:



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).

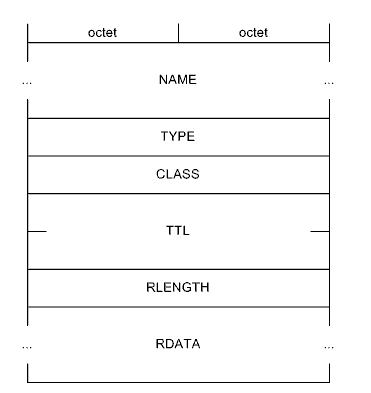
Адресное пространство доменных названий имеет иерархическую древовидную структуру. Каждый узел дерева на некотором уровне иерархии обозначают DNS меткой (DNS label) длиной от 0 до 63 байтов (должна начинаться с буквы и состоять из комбинации букв любого регистра, цифр и символа -). Метка нулевой длины 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.

**48) 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:

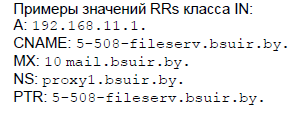
1) IN - Internet

2) CS - CSNET (устарел и аннулирован)

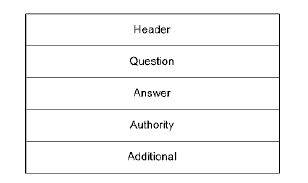
3) CH - Chaosnet (устарел)

4) HS - Hesiod (для БД, очень редкий)

Остальные значения классов зарезервированы



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

**49) Виртуальные соединения в сети передачи данных**

Одним из ключевых терминов транспортного уровня является термин cоединение. 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

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

**50) 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

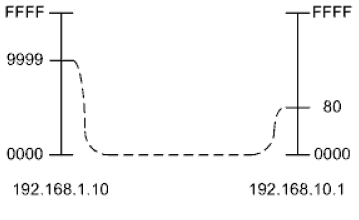
Основной его недостаток состоит в отсутствии возможности адаптации к изменениям в СПД

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

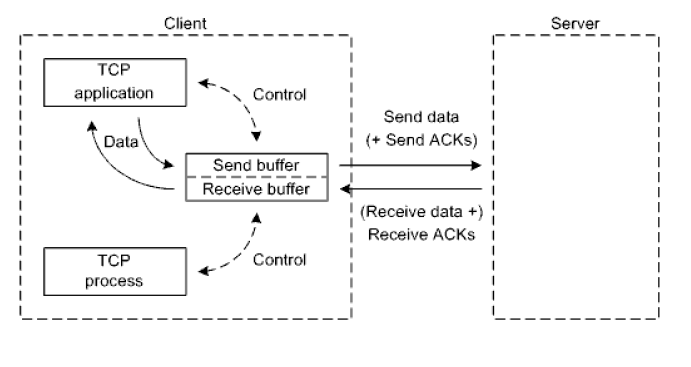
Скользящее окно, особенно в сочетании с динамическим, позволяет ускорить адаптацию к топологическим и другим изменениям в СПД.

**51) 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.



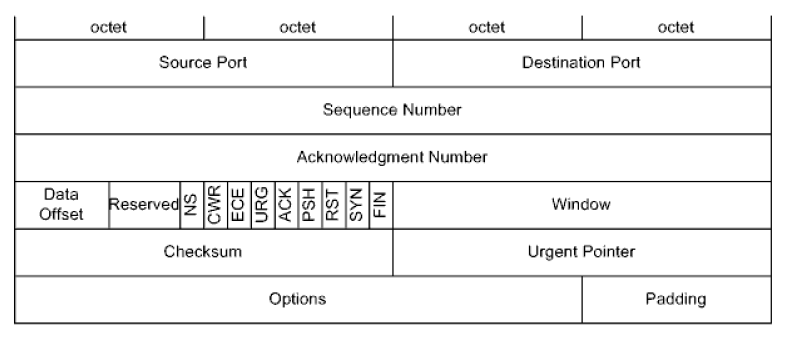
Структура соединения TCP:



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)

**52) 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) NS (Nonce Sum) – флаг - контрольная сумма для проверки правильности кодов явных уведомлений о заторах (связан с QoS, связан с IP заголовком)

8) CWR (Congestion Window Reduced) - флаг уменьшения окна затора при явном уведомлении о заторе

9) ECE (Explicit Congestion Notification Echo) - флаг подтверждения явного уведомления о заторе

10) URG (URGent Pointer field significant) - флаг значимости указателя на экстренные данные

11) ACK (ACKnowledgment field significant) - флаг значимости подтверждающего номера

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

**53) 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.

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.

**54) 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 анонсируется небольшое предлагаемое окно.

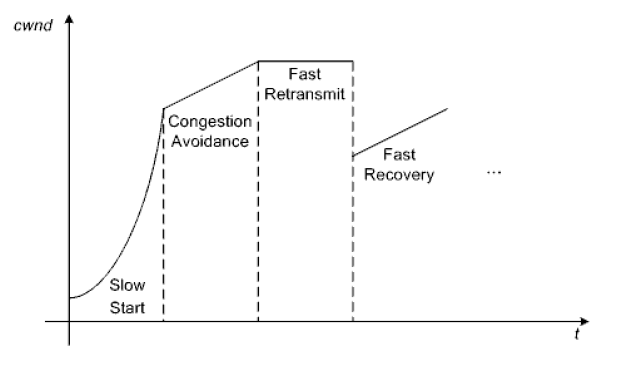
Также стандартизированы четыре дополнения Ван Якобсона, призванные бороться с перегрузками в СПД

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)

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.

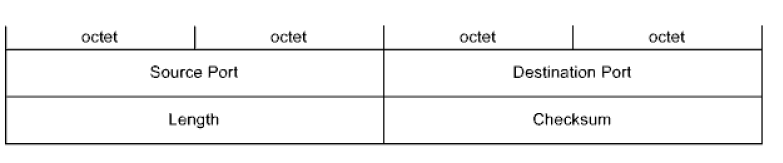
3) Быстрая повторная передача (fast retransmit). При получении принимающей стороной разупорядоченного сегмента с данными (возможно из за потери ожидаемого сегмента с данными) незамедлительный повтор подтверждения с AN недостающего сегмента с данными. При получении передающей стороной трех одинаковых 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.



**55) 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)



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, плюсданных)

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.

**56) 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).

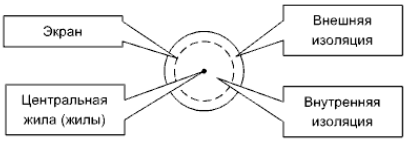
4. Эфир (ether).

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.

**57) Среды передачи данных на основе коаксиальных кабелей**

Широко используется в телевидении. Важное достоинство – передавать в один и тот же момент множество сигналов. Внутри выглядит следующим образом:



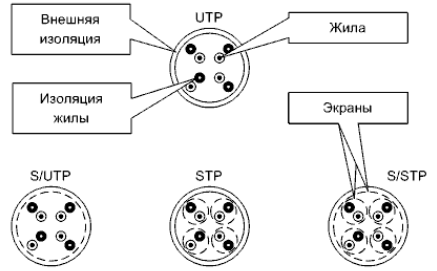
Для формирования системы на таком кабеле нужны как минимум BNC (bayonet-neill-concelman) коннекторы, T-соединители и пара терминаторов, один из которых заземляют.

Коаксиальный кабель, в отличие от витой пары, устойчив к электромагнитным помехам. И способен передавать сигналы на бóльшие расстояния.

Коаксиальные кабели производители обычно выпускают черными, реже серыми.

**58) Среды передачи данных на основе витых пар**

В сегментах КС широко используются четыре вида витых пар:



Где:

* TP – twisted pair
* S – shielded
* U – unshielded

Сама витая пара состоит из 8 кабелей (бело-оранжевый, оранжевый, б-зелёный, синий, б-синий, зелёный, б-коричневый, коричневый), которые разводятся по стандарту 568-B под RJ-45.

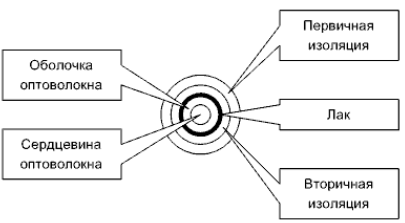
Обычно на витой паре доступны по два асинхронных канала передачи и приёма

В типовых случаях витой парой соединяют разноранговое сетевое оборудование.

Цвета самих кабелей в витой паре не оговорены. Обычно привязаны к палитре RAL и имеют серый цвет. Другие цвета говорят о более высоком качестве.

**59) Среды передачи данных на основе оптоволоконных кабелей**

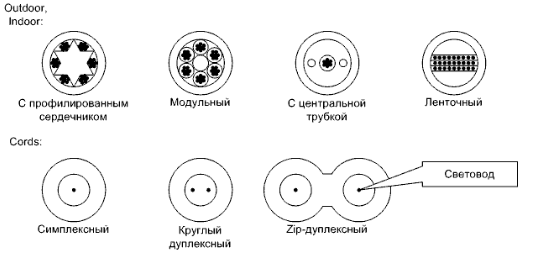
Рабочими компонентами оптоволоконных кабелей являются световоды изготовленные из оптоволокна, т.е. особого кварцевого стекла. Световод – это оптический волновод. Рабочим компонентом является оболочка и сердцевина.



Большое количество изоляции обусловлено хрупкостью кабеля.

В стандартах также предусмотрены 8 видов светодиодов: OM1, OM2, OM3, OM4, OM5 – многорежимные, OS1, OS2, OS1a – однорежимные.

Также применяют множество видов оптоволоконных кабелей:



Также оптоволоконные соединения делят на несъемные (сплавные, механические) и съемные (контактные и линзовые).

**60) Физический уровень Ethernet**

Физический уровень определяет электрические или оптические свойства физического соединения между устройством и сетью или между сетевыми устройствами. Он дополняется уровнем MAC и уровнем логических каналов.

Физический уровень Ethernet включает в себя:

1) несколько интерфейсов физических сред

2) несколько порядков величины

3) скорости от 1 Мбит / с до 400 Гбит / с

Ключевые использовавшиеся либо используемые стандарты:

10BASE5 (1983) -- «толстый» (thick) коаксиальный кабель 50 Ω (до 500 m) плюс внешние приемопередатчики;

10BASE2 (802.3a, 1985) -- «тонкий» (thin) коаксиальный кабель 50 Ω (до 185 m) плюс интегрированные приемопередатчики;

10BASE-T (802.3i, 1990) -- две телефонные витые пары (до 100 m);

10BASE-FL (802.3j, 1993) -- два многорежимных световода (до 500 m) плюс нечетко регламентированные источники излучения (обычно LEDs);

100BASE-TX (802.3u, 1995) -- две неэкранированные либо экранированные витые пары категории 5 (до 100 m);

100BASE-FX (802.3u, 1995) -- два многорежимных световода (до 2 km) (реализации поддерживают и однорежимные световоды длиной десятки километров) плюс нечетко регламентированные источники излучения (реализации поддерживают LEDs и лазеры);

1000BASE-SX (802.3z, 1998) -- два многорежимных световода (до 275 m -- 62,5 um, до 550 m -- 50 um) плюс коротковолновые (short wavelength) лазеры (770 -- 860 nm);

1000BASE-LX (802.3z, 1998) -- два однорежимных (до 5 km) либо многорежимных световода (до 550 m) плюс длинноволновые (long wavelength) лазеры (1270 -- 1355 nm);

1000BASE-T (802.3ab, 1999) -- четыре неэкранированные либо экранированные витые пары категории 5 (до 100 m);

2.5GBASE-T (802.3bz, 2016) -- четыре неэкранированные либо экранированные витые пары категории 5e (расстояние до 100 m);

5GBASE-T (802.3bz, 2016) -- четыре неэкранированные либо экранированные витые пары категории 5e (расстояние до 100 m);

10GBASE-SR (802.3ae, 2002) -- два многорежимных световода (до 33 m -- 62,5 um, до 400 m -- 50 um) плюс коротковолновые лазеры (840 -- 860 nm);

10GBASE-LR (802.3ae, 2002) -- два однорежимных световода (до 10 km) плюс длинноволновые лазеры (1310 nm);

10GBASE-ER (802.3ae, 2002) -- два однорежимных световода (до 30 km) плюс экстрадлинноволновые (extra long wavelength) лазеры (1550 nm);

10GBASE-T (802.3an, 2006) -- четыре неэкранированные (до 55 m) либо экранированные (до 100 m) витые пары категории 6, либо четыре неэкранированные, либо экранированные витые пары категории 6A (до 100 m).

**61) Структурированные кабельные системы и их модели**

Структурированная кабельная система – это упорядоченная по тем или иным критериям совокупность телекоммуникационных и силовых кабелей, различного оборудования, а также соответствующих специализированных помещений. Основой для построения СКС служит древовидная топология, узлами которой служит сетевое оборудование определённого типа. Помещения в СКС бывают:

1) Кроссовые (вспомогательное, активное и пассивное сетевое оборудование)

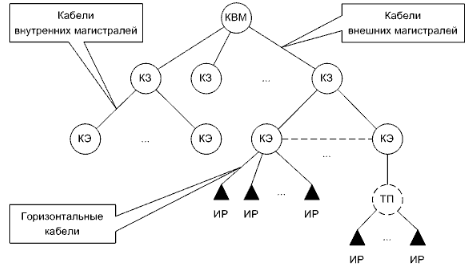
2) Аппаратные (кроме кроссового, может быть расположено серверное оборудование)

СКС включает в себя три подсистемы:

3) Подсистема внешних магистралей (main, campus) – основа связи между компактно расположенными зданиями

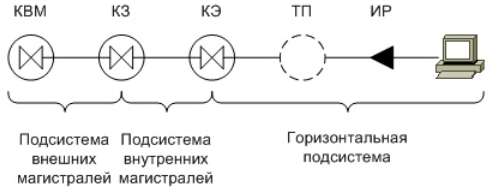
4) Подсистема внутренних магистралей (building) – связывает между собой этажи одного здания

5) Горизонтальная подсистема (horizontal) – связывает оборудование в пределах одного этажа

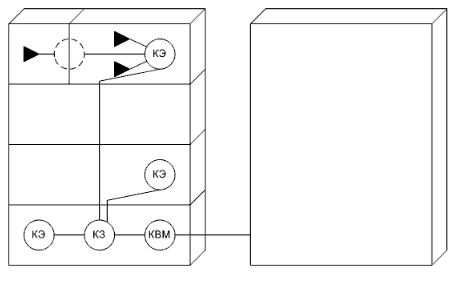


Основная модель СКС

* КВМ – кроссовая внешних магистралей
* КЗ – кроссовая здания
* КЭ – кроссовая этажа
* ИР – информационная розетка (для рабочего места)
* ТП (пунктирной линией) – точка перехода



Горизонтальная модель СКС



Функциональная модель СКС здания

**62) Питание и заземление в структурированных кабельных системах**

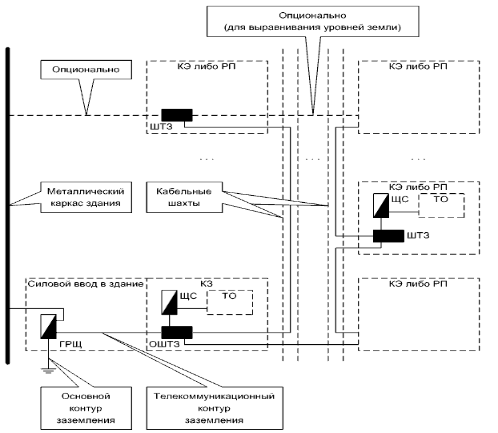
В СКС должно быть уделено внимание заземлению и питанию по следующим причинам:

1) Предотвращение поражения людей электрическим током

2) Защита кабельных трактов и сетевого оборудования от выхода из строя/помех

3) Обеспечения возможности прохождения сигналов применительно некоторым видам сетевого оборудования.

Согласно стандарту TIA-607, в дополнение к основному контуру заземления здания или сооружения создают дополнительный, телекоммуникационный контур заземления (контур рабочего заземления)



Модель заземления

* ГРЩ – главный распределительный щит здания
* ШТЗ – шина телекоммуникационного заземления
* ОШТЗ – основная ШТЗ
* ЩС – щит силовой
* РП – рабочее место
* ТО – телекоммуникационное оборудования

Для защиты от электрических зарядов в атмосфере применяют специальные устройства – газоразрядники.

**63) Пожарная безопасность структурированных кабельных систем**

Т.к. СКС охватывают здание полностью, серьёзное внимание должно быть уделено пожарной безопасности.

Согласно американским стандартам NEC, предусмотрены 4 уровня пожарной безопасности (от высших к низшим):

1) Plenum – сюда относят кабели, которые можно располагать как угодно (при притоке воздуха, достаточного для поддержания горения, так называемая plenum-область)

2) Riser – кабели, которые можно прокладывать в кабельных шахтах

3) General purpose – кабели, которые можно прокладывать везде, кроме plenum-областей.

4) Residential – кабели, на прокладку которых нанесены определённые ограничения (например, только для жилых помещений)

В состав маркировки кабелей обычно вводят дополнительные обозначения материала оболочек:

1) PVC – Поли винил хлорид (ПВХ)

2) PE (PolyEthylene) – полиэтилен

3) FR (Flame retardant) – огнестойкий

4) CST – бронирован гофрированной стальной лентой

5) И т.д.

**64) Технология PoE**

Относительно недавно производители сетевого оборудования стали разрабатывать технологии, позволяющие запитывать относительно маломощные Ethernet-устройства через информационные кабели (например, через витую пару) – технологии под названием PoE (Power over Ethernet).

Постепенно были введены два общепринятых стандарта: 802.3f, 802.3at, но до сих пор производители используют собственные пропреитарные технологии (например, Cisco Universal Power over Ethernet)

В структуру PoE входят ряд блоков:

1) PSE (Power Sourcing Equipment) – вводит питающее напряжение в кабель.

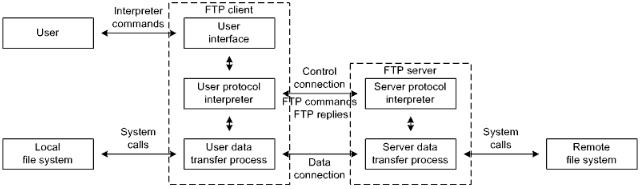
2) PD (Powered Device) – питается от этого напряжения.

PSE может располагаться как на одном конце, так и на обоих. Либо может «вклиниваться» в кабель, т.е. быть PoE-injector-ом.

Обычно используется для небольших PD.

**66) Структура системы FTP**

FTP-клиент обслуживает запросы пользователя и работает на локальной, по отношению к нему, станции. FTP-сервер работает на удалённой станции и обслуживает запросы FTP-клиента.



Структура FTP-системы

Как в составе сервера, так и в составе клиента выделяют протокольные интерпретаторы и процессы пересылки данных.

FTP-сервер представляет собой непрерывно выполняющуюся программу, ожидающую запросы от FTP-клиентов, выраженную в виде демона UNIX или сервиса Windows.

В отличие от многих протоколов FTP использует не одно, а два соединения, значит для него зарезервированы два программных порта: 20 – FTP Data (информационное соединение), 21 – FTP (управляющее соединение).

**67) Протокол FTP и режимы обмена по протоколу FTP**

FTP относят к протоколам прикладного уровня, ориентированным на пользователя. Это значит, что реализация FTP должна предоставить пользователю функционально полный интерфейс.

Классический интерфейс FTP представляет интерпретатор командной строки, активируемый вводом команды ftp. В качестве аргументов принимаются IP-адрес FTP-серверов, а также номер порта, если отличен от стандартного. В FTP два программных порта, 20 – FTP Data, 21 – FTP (управляющее соединение). Представляет собой непрерывно выполняющуюся команду, ожидающую запросы от FTP-клиентов.

FTP поддерживает 3 режима пересылки сообщений:

1) Stream – файл пересылается как непрерывный поток байт. <EOF> - конец пересылки

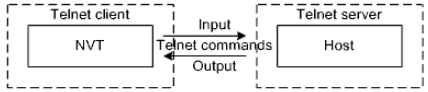
2) Block – файл пересылается в виде последовательности блоков, каждый из которых имеет заголовок, в котором записывается кол-во байт. Редкий способ

3) Compressed – файл пересылается в сжатом простейшими алгоритмами виде. Редкий способ

Также есть два режима работы: активный и пассивный.

**68) Структура и особенности системы Telnet**

Telnet базируется на клиент-серверной модели и использует протокол TCP. Задействуется одно соединение. Стандартный номер программного порта Telnet-сервиса – 23.



Структура telnet-системы

Основная задача telnet – обеспечение корректной транспортировки символов потока и ввода-вывода между NVT (network virtual terminal) и хостом. Используется буферизация, в том числе для того чтобы не нагружать СПД. По умолчанию набранные символы отсылаются моментально. В режиме linemode – при нажатии Enter.

Главный недостаток – полная незащищённость соединения от несанкционированного доступа. Данные и текст пересылается в виде открытого текста (в том числе пароли, номера телефонов и т.д.). На смену telnet пришёл SSH (secure shell). В нём идея абсолютно та же, но соединение защищено.

**69) Электронные письма и почтовые ящики**

Сообщениями протоколов электронной почты являются электронные  
письма (emails).

Электронные письма имеют текстовую природу. По аналогии с бумажным письмом, электронное письмо так же состоит из конверта (envelope) и содержимого (content). Содержимое, в свою очередь, состоит из заголовка (header) и основного текста (body).

Структура (прежде всего, синтаксис) электронных писем неоднократно  
регламентировалась стандартами (начиная с RFC 822 и более старых,  
заканчивая RFC 5322).

Для обеспечения прав и обязанностей, связанных с электронными  
письмами, предусмотрены два механизма: DKIM (DomainKeys Identified Mail)  
Signatures (RFC 6376) и SPF (Sender Policy Framework) (RFC 7208).  
Изначально, в отношении всех компонентов электронного письма  
допускалась только 7-битная кодировка US-ASCII.

Одним из ключевых понятий системы электронной почты является  
понятие почтового ящика (mailbox, иногда maildrop). Электронный почтовый  
ящик по своей сути ничем не отличается от почтового ящика для бумажных  
писем, но не всегда представляет собой файловое хранилище.  
Почтовые ящики могут быть расположены как на выделенных для этого  
почтовых серверах -- лучше переводить как MXes (Mail eXchanges), так и на  
пользовательских станциях. Могут быть как локальными, так и удаленными от пользователя.