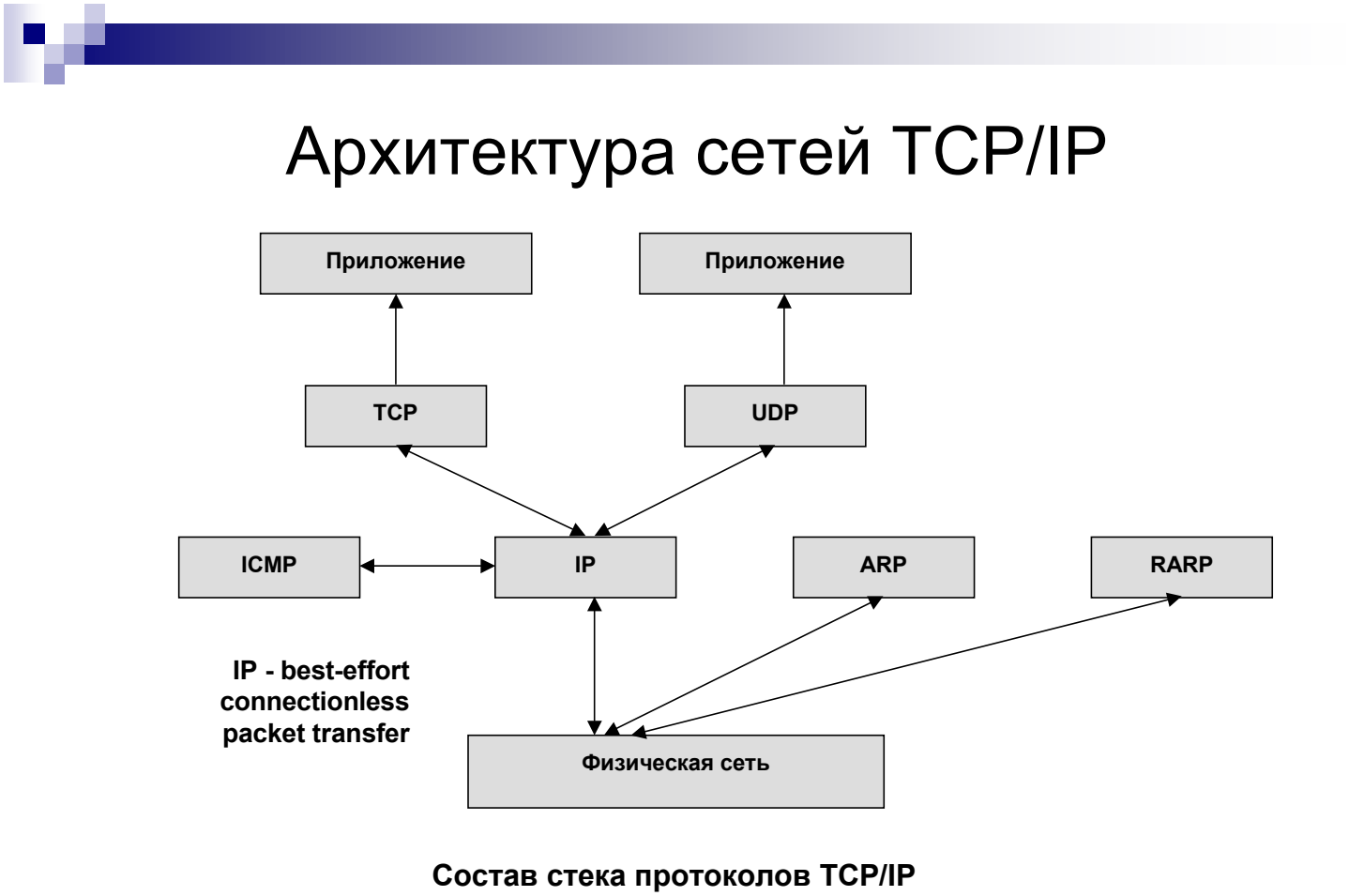
Лекция 3. Сети TCP/IP.



Состав стека протоколов TCP/IP

2

Архитектура сетей TCP/IP.

В состав стека протоколов TCP/IP, кроме давших название этим сетям протоколов Transmission Control Protocol и Internet Protocol, входят: User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), ARP (Address Resolution Protocol), RARP (Reverse Address Resolution Protocol) и ряд протоколов прикладного уровня, в частности TELNET, FTP, SMTP, SNMP и HTTP.

В отличие от 7-ми уровневой модели OSI протокольный стек TCP/IP разбит на 4 уровня. Сетевой уровень (IP) обеспечивает передачу информации через произвольную комбинацию сетей, использующих этот же набор протоколов. **IP-протокол предоставляет лишь один вид сервиса – передачу пакетов без предварительного установления соединения и настолько хорошо, насколько получиться** (best-effort connectionless packet transfer). Пакеты пересылаются между узлами коммутации без предварительного установления соединения; они маршрутизируются независимо, и пакеты одного приложения могут доставляться по разным маршрутам. Узлы коммутации, соединяющие смежные сети, могут испытывать перегрузки и уничтожать пакеты. Ответственность за восстановление утерянных пакетов и надлежащий порядок их передачи приложению лежит на транспортном уровне, который представлен протоколами TCP и UDP.

Разнообразие требований сетевых приложений обусловило необходимость двух протоколов транспортного уровня. Так, например, приложения передачи файлов и web (FTP, HTTP) для пересылки

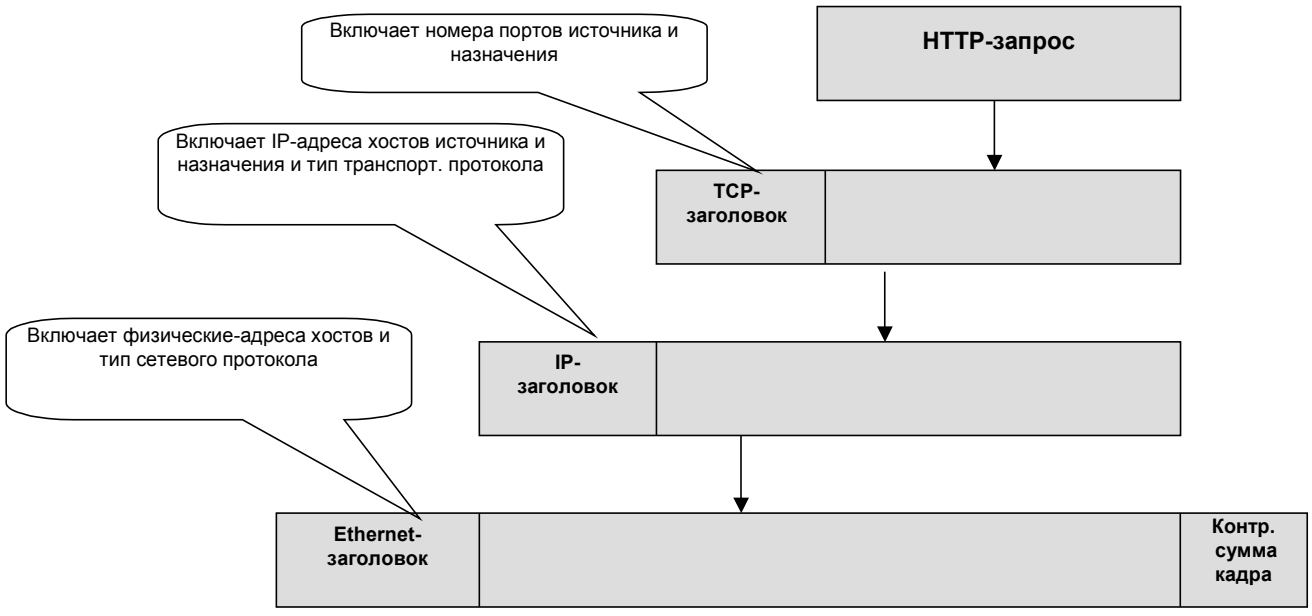
своих сообщений используют TCP, в то время как приложения управления сетевыми устройствами, служба имен (SNMP, DNS), потоковые приложения реального времени используют в качестве транспортного протокола UDP. Далее будем называть протокольные блоки TCP сегментами, а блоки UDP – дейтограммами.

Сетевой уровень (протокол IP) мультиплексирует протокольные блоки транспортного уровня в IP-потоки; при этом сегменты транспортного уровня могут фрагментироваться (если они превышают максимально допустимый размер, определяемый канальным протоколом). Протокольные блоки IP обычно называют пакетами.

После вычисления маршрута передачи пакета, посредством протокола ARP определяется физический адрес следующего на маршруте хоста и пакет направляется на физический уровень сети. Иногда бывает необходимо решить обратную задачу, то есть по заданному физическому адресу определить логический сетевой адрес устройства. В частности, эта проблема актуальна для процедуры загрузки бездисковых станций, когда такая станция рассылает в широковещательном режиме запрос, содержащий ее физический адрес и «просьбу» сообщить ей логический сетевой адрес. Этот запрос обрабатывается сервером RARP, который и передает пославшей его станции требуемую информацию.

Физический уровень TCP/IP сети может использовать любую технологию канального уровня – Ethernet, Token Ring, ATM, PPP и т.д. Но для обеспечения прозрачности физического уровня необходимо, чтобы на сетевом уровне были предусмотрены процедуры дефрагментации пакетов до размера, разрешенного соответствующим протоколом канального уровня. Обратная процедура, т.е. объединение пакетов малых размеров до величины, приемлемой на канальном уровне, не предусматривается.

Архитектура сетей TCP/IP



Инкапсуляция протокольных блоков в TCP/IP стеке

3

Протокольные блоки вышележащих уровней инкапсулируются в протокольные блоки нижележащих уровней так, как это показано на слайде. При этом, блок каждого уровня содержит специфическую информацию, позволяющую точно адресовать его. Так, сегмент TCP (UDP-

дейтограмма) содержит в своем заголовке номер **порта**, однозначно определяющий приложение, которому он принадлежит. IP-пакет адресуется **логическим адресом** хоста, также однозначно определяющим его в распределенной сети. Кадр канального уровня включает в себя **физический адрес** ближайшего на маршруте доставки хоста, которому необходимо передать пакет.

IP-протокол.

Основной протокол стека TCP/IP, - Internet Protocol (IP), - по своим функциям соответствует сетевому уровню модели взаимодействия открытых систем (OSI). Механизмы протокола, описанные в документе RFC 791, обеспечивают ненадежную доставку пакетов данных между сетевыми устройствами (устройствами, имеющими сетевой адрес) в режиме без предварительного установления соединения (дейтограммный сервис). Этот тип сервиса часто называют сервисом «настолько хорошо, как получиться» (best effort service), что отражает отсутствие в протоколе процедур контроля доставки пакетов. Решение задачи надежности доставки возлагается на протоколы верхних уровней, главным образом на TCP. Основными функциями протокола IP являются:

- формирование пакетов из сегментов транспортного уровня, с предварительной фрагментацией (если необходимо) последних;
- обеспечение логической адресации сетевых устройств
- поддержка процесса маршрутизации
- продвижение пакетов от одного узла коммутации до другого.

Структура IP – пакета

<table>
<tr><td colspan="8">0 4 8 16 31</td></tr>
<tr><td colspan="2">Версия (Version)</td><td colspan="2">Длина (IHL)</td><td colspan="2">Тип сервиса (ToS)</td><td colspan="2">Общая длина пакета (Total Lenght)</td></tr>
<tr><td colspan="4">Идентификатор</td><td colspan="2">Флаги</td><td colspan="2">Смещение фрагмента</td></tr>
<tr><td colspan="2">Время жизни (TTL)</td><td colspan="2">Протокол (см. след. слайд)</td><td colspan="4">Контрольная сумма</td></tr>
<tr><td colspan="8">Адрес отправителя (Source IP address)</td></tr>
<tr><td colspan="8">Адрес получателя (Destination IP address)</td></tr>
<tr><td colspan="6">Опции (поле переменной длины)</td><td colspan="2">Выравнивание до 32 бит (padding)</td></tr>
</table>

Формат заголовка IP-пакета

4

Структура IP-пакета.

Функции IP-протокола реализуются посредством специальной структуры заголовка пакета, формат которого приведен на слайде.

Заголовок содержит поля фиксированной длины (первые 20 байт) и поле переменной длины (поле опций), размер которого может достигать 40 байт. Для выравнивания этого поля по 32 битной границе предусмотрено поле «Выравнивание» (Padding) которое заполняется нулями. Рассмотрим назначение полей заголовка.

Версия (Version) – поле определяет номер версии протокола. В настоящее время используется версия 4 и ведется активная подготовка к переходу на версию 6. Версия 5 описывает протокол ST2, разработанный для передачи данных потоковых приложений реального времени. Поле проверяется перед обработкой пакета и пакеты несогласующейся с протокольным стеком приемника версией, отбрасываются. Одновременно, включение версии в каждую дейтограмму позволяет использовать разные, но согласующиеся, версии на разных хостах.

Длина заголовка (Internet Header Length, IHL) - Поле определяет длину заголовка, измеренную в 32-битных словах. Корректный заголовок имеет длину не менее 5 слов. Длина поля опций (в 32-разрядных словах) может быть определена как значение этого поля минус 5.

Тип сервиса (Type of service, ToS) – определяет тип требуемого обслуживания пакета. Первые три бита задают уровень приоритета обслуживания (0-7), 3-5 биты определяют требования к задержке (какая получится, низкая), уровень пропускной способности (обычный, высокий) и надежность доставки (какая получится, высокая). Практически, большая часть маршрутизаторов игнорирует данные этого поля. Однако в настоящее время в связи с разработкой механизмов обеспечения в IP-сетях служб с гаранированным качеством обслуживания делаются попытки использования значений этого поля.

Общая длина (Total length) - поле содержит общую длину пакета, размер которого не может превышать 65535 байт. Практически пакеты такой длины никогда не используются, поскольку технологии канального уровня накладывают свои ограничения. Так, Ethernet не допускает кадров с длиной более 1500 байт, FDDI – 4096 байт и т.д. В этой связи, протокол IP выполняет фрагментацию сегментов данных, поступающих к нему от TCP и UDP протоколов. Следует отметить, что маршрутизатор не выполняет сборку пакетов, даже если следующая сеть имеет параметр MTU (Maximum Transmission Unit), допускающий более крупные пакеты. Сборка пакетов в исходный сегмент производится на месте назначения.

Поля «**Идентификатор**», **«Флаги**» и «**Смещение фрагмента**» управляют процессом сборки сегмента.

Время жизни (Time to live, TTL) - поле, определяющее максимальное время, которое пакет может существовать в сети. Значение этого поля (в секундах) устанавливается при отправке пакета и уменьшается на единицу по мере прохождения им маршрутизаторов. При достижении нулевого значения этого поля пакет уничтожается. Максимальное значение поля – 255 секунд. Этот механизм помогает избежать перегрузок сети при возникновении ошибок в таблицах маршруизации, приводящих к образованию петель.

Структура IP – пакета

Значение поля	Ключевое слово	Протокол
0	**Reserved**	**Зарезервировано**
1	**ICMP**	**Internet Control Message Protocol**
2	**IGMP**	**Internet Group Management Protocol**
4	**IP**	**Internet Protocol**
6	**TCP**	**Transmission Control Protocol**
17	**UDP**	**User Datagram Protocol**
89	**OSPF**	**Open Shortest Path First**

Поле «Протокол» заголовка IP-пакета

5

Протокол (Protocol) – поле указывает модулю какого протокола (TCP, UDP, ICMP) передать полученный IP-пакет. На слайде 5. приведены значения этого поля для некоторых из протоколов. В дальнейшем нас будут интересовать TCP, UDP, ICMP.

Структура IP – пакета

0	4	8	16	31
Версия (Version)	**Длина (IHL)**	**Тип сервиса (ToS)**	**Общая длина пакета (Total Lenght)**	
Идентификатор			**Флаги**	**Смещение фрагмента**
Время жизни (TTL)		**Протокол (см. след. слайд)**	**Контрольная сумма**	
Адрес отправителя (Source IP address)				
Адрес получателя (Destination IP address)				
Опции (поле переменной длины)				**Выравнивание до 32 бит (padding)**

Формат заголовка IP-пакета

Контрольная сумма (Header checksum) – поле содержит значение контрольной суммы**,** рассчитанной только по заголовку. Поскольку значения некоторых полей заголовка изменяются по мере прохождения пакета по маршруту (поле TTL, например), то значения рассматриваемого поля проверяются и пересчитываются на каждом маршрутизаторе. Этот механизм является единственным средством обеспечения достоверности передачи, содержащимся в протоколе IP.

Адрес отправителя (Source IP address) и **Адрес получателя (Destination IP address)** – поля одинаковой длины (32 бита), содержащие соответствующие адреса. Правила адресации в IP-сетях будут рассмотрены далее.

Опции (Options) – необязательное поле, используемое при отладке сетей и для запроса определенных специфических процедур обработки. В настоящее время используется крайне редко. В связи с разработкой новых протоколов, обеспечивающих большую гибкость в обработке IP-трафика, возможность использования этих полей вновь стала предметом обсуждения комитетов по стандартизации.

При получении пакета маршрутизатор вычисляет контрольную сумму заголовка пакета и, если она не совпадает со значением поля «Контрольная сумма», то пакет отбрасывается. При положительном результате проверки, производится изменение некоторых полей и рассчитывается новое значение поля «Контрольная сумма». Затем по таблице маршрутизации определяется адрес следующего маршрутизатора, на который должен быть направлен этот пакет, и он передается на соответствующий интерфейс.

Адресация в сетях IP

Классификация IP адресов

Класс сетей	Значение первого байта адреса	Количество сетей	Количество хостов в сети класса	Удельный вес класса в адресном пространстве, %
A	001 – 126	126	16 777 214	50
B	128 – 191	16 384	65 534	25
C	192 - 223	2 097 152	254	12,5

6

Адресация в сетях IP.

Для идентификации каждого компьютера в IP-сети необходима система их адресации. При этом учитывается, что сетевые устройства (компьютер, маршрутизатор и т.д.) могут иметь несколько сетевых интерфейсов, и каждый из них должен иметь уникальный адрес. Если обратиться к аналогии обычной адресной системы (улица, дом, квартира), то становиться ясной целесообразность построения системы сетевой адресации по иерархии «сеть-интерфейс». Принятая в сетях IP система адресации описана в документах RFC 990 и RFC 997. Каждое сетевое устройство имеет адреса трех типов:

1. Физический адрес узла, определяемый используемой технологией канального уровня. Для Ethernet – это MAC-адрес его сетевой карты, назначаемый фирмой-производителем. Он представляет собой шести-байтовое число, первые три байта которого однозначно определяют фирму-производителя, а последние три байта – уникальны для каждой карточки, произведенной в рамках данной фирмы.
2. IP-адрес, состоящий из 4-х байтов, и также являющийся совершенно уникальным.
3. Символьный идентификатор – имя, назначаемое по определенным правилам и являющееся полным эквивалентом IP-адреса.

IP-адрес строится по двухуровневой иерархии, т.е. он объединяет в себе адрес сети и адрес хоста. Разделение сетевого адреса на 2 части имеет большой практический смысл, ибо позволяет магистральным маршрутизаторам существенно сократить размер своих таблиц коммутации, формируя их на основании только сетевой части адреса назначения. Для удовлетворения потребностей адресации сетей различного масштаба были введены несколько классов сетей, отличающиеся размером полей, отводимых для указания номера сети и номера хоста. При этом, размер поля полного адреса всегда равен 32 битам. Структура адресов сетей разных классов приведена на слайде.

IP-адрес обычно записывается в форме 4-х трехразрядных десятичных чисел, разделенных точкой. Каждое из этих десятичных чисел соответствует одному байту двоичного представления адреса. Так, например, адрес 10000000 10000111 01000100 00000101 в десятичном представлении имеет вид 128.135.68.5. В этом случае, т.к. первые два бита адреса – 10, то это адрес хоста, принадлежащего сети класса **B** и, следовательно, левые 16 бит являются адресом сети, а правые16 бит – адресом хоста.

Некоторые адреса являются зарезервированными и не могут присваиваться хостам. Так, адрес 127.x.x.x (x – означает любое число, обычно 0) зарезервирован для обратной связи, используемой при тестировании взаимодействия процессов на одной сетевой станции. Когда приложение использует этот адрес в качестве адреса назначения, стек TCP/IP данного хоста возвращает данные приложению, ничего не передавая на физический интерфейс. Поэтому адреса, начинающиеся на 127, запрещается присваивать сетевым устройствам. Другим зарезервированным адресом является, так называемый, широковещательный адрес, содержащий 1, или 0, во всех своих битах. Пакет с адресом назначения 255.255.255.255 (1.1.1.1) будет доставлен всем устройствам сети, к которой принадлежит узел-отправитель, но маршрутизаторы такие пакеты не обрабатывают. Существует и направленное широковещание – способ адресации, при котором один пакет, отосланный в определенную сеть, будет доставлен всем ее хостам. Такой пакет должен содержать корректный адрес сети и иметь все биты адреса хоста равными 1. Так, например, пакет с адресом 184.90.255.255 будет доставлен всем станциям сети класса **B**, имеющей адрес 184.90.

Из рисунка вверху хорошо видно, что значения первых четырех битов адреса однозначно определяют обе границы его сетевой части. Нетрудно сосчитать, что максимальное число сетей класса **A** равно $2^6+2^5+ \ldots.+2^0 -1 =126$ (сетевой префикс, состоящий из одних нулей недопустим). Каждая сеть класса **A** может содержать $2^{24}-2 = 16\ 777\ 214$ устройств. В целом, адресный блок сетей класса **A** занимает около 50% общего адресного пространства. В таблице внизу слайда содержатся аналогичные показатели для сетей класса **B** и **C**.

Разбиение IP-сети на подсети.

Рассмотренная выше двухуровневая схема адресации оказалась недостаточно гибкой и в 1985 году была определена трехуровневая схема адресации (RFC 950). Необходимость перехода к такой системе адресации можно проиллюстрировать следующим примером. Организация получила сеть класса **B**, позволяющую адресовать 65000 хостов. Пока в сети работали относительно немного станций – она функционировала благополучно. Но с ростом числа активных хостов неизбежно рос и широковещательный трафик, поскольку этот тип пакетов активно используется в служебных целях многими протоколами. Это обстоятельство неизбежно вело к снижению эффективной производительности сети. Кроме того, управление несколькими десятками тысяч подключений из единого административного центра представляет собой очень трудную задачу. К этим проблем следует добавить и то, что любая организация развивается в направлении роста самостоятельности своих подразделений, что также неизбежно должно отражаться и в структуре сети. Возможность же структурирования сети организации за счет получения дополнительных номеров сетей была быстро исчерпана и резкая нехватка адресного пространства стала реальностью еще во второй половине 80-х годов. Кроме того, рост числа используемых сетей вел к росту таблиц маршрутизации магистральных (межсетевых) маршрутизаторов и, следовательно, к снижению их производительности.

Перечисленные проблемы в значительной степени связаны с проблемой масштабируемости сети, и они, в значительной степени, были разрешены посредством введения в иерархию адресации третьего уровня. Этот уровень, получивший название «уровень подсети», был выделен в пространстве адресов хоста

Разбиение IP-сети на подсети

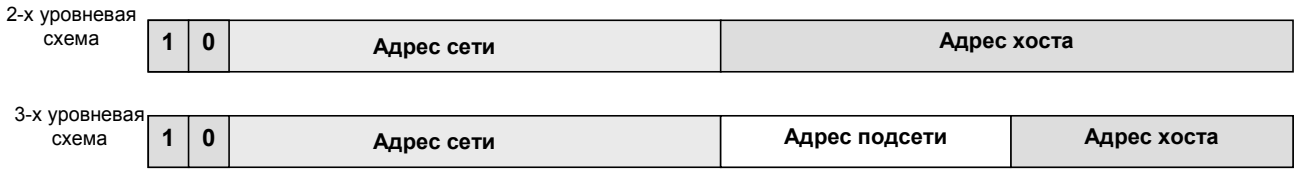


Схема адресации с введением подсетей

	Подсеть	Маска	Эквивалентный адрес подсети	Число хостов в подсети
Исходная сеть	**193.10.1.0**	**255.255.255.0**	**193.10.1.0/24**	**254**
Подсеть 1	**193.10.1.0**	**255.255.255.128**	**193.10.1.0/25**	**126**
Подсеть 2	**193.10.1.128**	**255.255.255.192**	**193.10.1.128/26**	**62**
Подсеть 3	**193.10.1.192**	**255.255.255.224**	**193.10.1.192/27**	**30**
Подсеть 4	**193.10.1.224**	**255.255.255.224**	**193.10.1.224/27**	**30**

Разбиение сети класса «С» на подсети

При этом поля адреса сети и адреса подсети в совокупности называют расширенным сетевым префиксом. В рассмотренном выше примере, выделение в пространстве адреса хоста 6 разрядов для адреса подсети позволяет организовать 62 подсети, содержащих до 1022 хостов каждая. Такое разбиение на подсети не требует согласования со специальным служебным органом Интернет, регулирующим распределение адресного пространства, Network Information Center.

Отрицательным следствием введения подсетей стало усложнение процедуры определения адреса хоста. Действительно, сетевые устройства используют старшие биты сетевого адреса для определения класса сети, после чего, в случае двухуровневой иерархии, легко находится граница между битами сетевого адреса и адреса хоста. В трехуровневой системе адресации этот прием работать не сможет. Для определения адреса подсети и адреса хоста потребовалось ввести **сетевую маску –** 32-битный адрес, в котором все биты, соответствующие битам расширенного сетевого префикса, установлены в 1, а соответствующие битам адреса хоста – в 0. Так, например, пусть необходимо в сети класса **B** 132.10 выделить подсеть, содержащую не более 100 хостов. Для адресации такого числа сетевых устройств достаточно располагать 7 битами (2^7 =128), которые и отводятся для адреса хоста; оставшиеся 9 бит отводятся для номера подсети (их можно организовать 2^9 -2=510) а старшие 16 бит – это адрес сети.

Таким образом, маска подсети, в которой находятся хосты с адресами:

132.10.12.129, 132.10.12.130,….,132.10.12.254

будет иметь вид

11111111 11111111 11111111 10000000 (255.255.255.128).

Если маршрутизатор получит пакет с адресом назначения

10000100 00001010 00001100 10110000 (132.10.12.176)

и выполнит по отношению к нему и сетевой маске операцию логического И, то результат будет содержать номер подсети. В данном случае получаем:

10000100 00001010 00001100 10000000,

что соответствует адресу подсети 132.10.12.128. Далее, по таблице маршрутизации будет найден следующий узел на пути к этой подсети, и пакет отправится на соответствующий интерфейс.

Необходимо заметить, что информация о маске подсети IP-пакетом не переносится (хост-источник о структуре сети, в которой находится получатель, ничего не знает). Передача этой информации является задачей протоколов маршрутизации, или она задается статически при конфигурировании маршрутизатора.

В стандартах, описывающих современные протоколы маршрутизации, часто делается ссылка на длину расширенного префикса сети, а не на маску подсети. В такой записи адрес устройства в рассмотренном выше примере имеет вид 132.10.12.176/25. В качестве примера нумерации подсетей в таблице приведено разбиение сети класса **C** на подсети. Указанное в таблице число хостов в каждой из подсетей на два меньше возможного числа адресов, поскольку адреса, в которых все биты поля адреса хоста установлены в единицу и в ноль, являются зарезервированными и используются как широковещательные для данной подсети. Так при подсетях, приведенных в таблице., адрес 193.10.1.0 является широковещательным лишь для подсети 193.10.1.0/24, а не для всех подсетей. Аналогично, адрес 193.10.1.255 является широковещательным только для подсети 193.10.1.224/27.

Введение подсетей, решив проблемы масштабирования адресного пространства, потребовало определенного усложнения протоколов маршрутизации, которые должны обрабатывать (и переносить) не только адрес сетевого устройства, но и его маску. В настоящее время все широко используемые протоколы маршрутизации (RIP-2, IS-IS, OSPF) переносят эту информацию.

IP маршрутизация.

Каждый хост (станция, маршрутизатор) ведет свои маршрутные таблицы, которые и определяют порядок обработки IP-пакетов. Передача пакетов между конечными станциями, требует взаимодействия

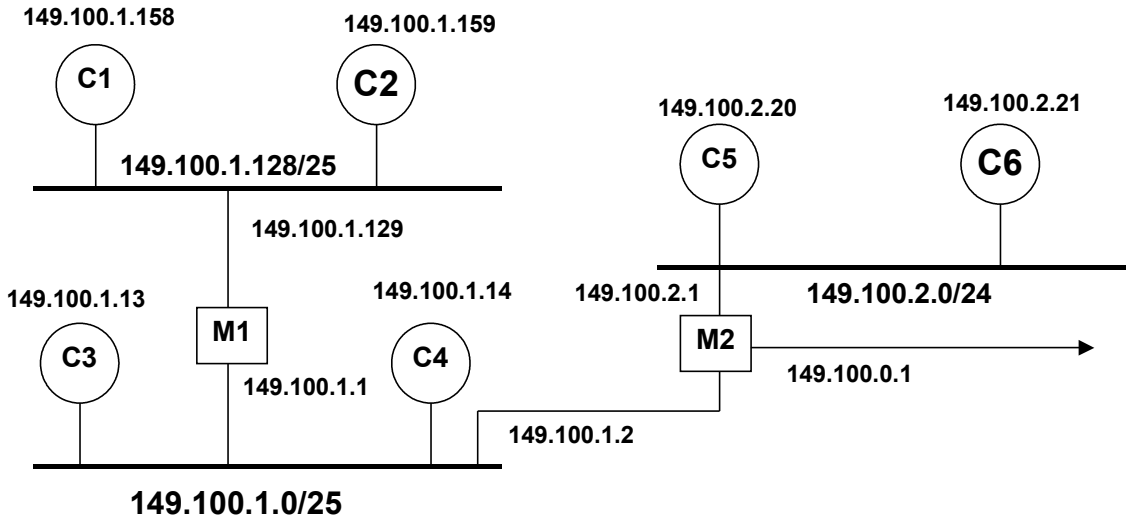
IP-модулей программного обеспечения этих станций и маршрутизаторов, связывающих сети, в которых они (станции) находятся. Рассмотрим, как обрабатывается пакет на этих сетевых устройствах.

Если в таблице маршрутизации станции-отправителя указано, что станция назначения является непосредственно присоединенной к той же ЛВС, то из таблицы физических адресов, которая ведется на каждой сетевой станции, извлекается физический адрес узла назначения, пакет инкансулируется в кадр канального протокола и передается к станции назначения. Если таблица маршрутизации станции-отправителя не содержит искомый сетевой адрес, то пакет отправляется по адресу маршрутизатора, который был указан при конфигурировании станции в качестве шлюза по умолчанию (default router, default gateway). Этот шлюз обязательно имеет физический интерфейс в той же ЛВС, что и станция-отправитель. При получении пакета маршрутизатор проверяет, не совпадает ли адрес назначения этого пакета с его собственным IP-адресом. Если это так, то пакет передается модулю протокола, указанного в поле «Протокол» заголовка пакета. В противном случае, маршрутизатор посредством своей таблицы определяет адрес следующего хоста, которому он должен передать этот пакет, и свой интерфейс, на который следует его направить.

Каждая строка в таблице маршрутизации содержит следующую информацию: IP-адрес сети (узла) назначения, IP-адрес следующего маршрутизатора, способного обеспечить передачу пакета в эту сеть (этому узлу), имя выходного интерфейса и некоторые флаги. Флаги содержат уточняющую информацию о каждой записи в таблице. Так например, флаг **H** определяет, является ли данная строка таблицы маршрутом к хосту (**H**=1), или к сети (**H**=0); флаг **G** уточняет, является ли она маршрутом к другому маршрутизатору (**G**=1), или определяет путь к непосредственно подключенной станции (**G**=0).

Поиск в таблице маршрутизации ведется следующим образом. Прежде всего, сканируется ее первый столбец с целью нахождения записи, точно соответствующей адресу назначения пакета. Если такая обнаруживается, то пакет отправляется по адресу, указанному в столбце «Следующий маршрутизатор». В противном случае, ведется поиск строки, содержащей адрес сети назначения. Если такой записи нет, то ищется строка, определяющая маршрут по умолчанию, т.е. маршрут к узлу с более полной информацией о траектории передачи этого пакета. Если не один из перечисленных вариантов не реализуем, то пакет уничтожается, а узлу-отправителю отправляется сообщение «Хост недостижим». Рассмотрим таблицы маршрутизации в сети, изображенной на след. слайде.

IP – маршрутизация



Пример сети, объединенной маршрутизаторами

Таблица маршрутизации **C6** :

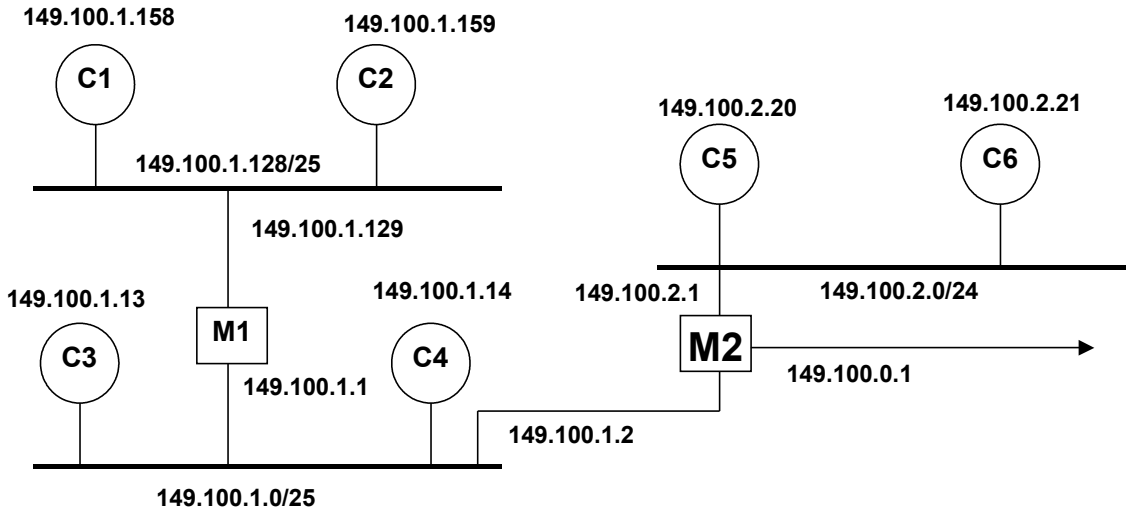
Адрес назначения	Маска сети	Следующий узел	Флаг	Интерфейс
127.0.0.0	255.0.0.0	127.0.0.1	H	lo0
Default	0.0.0.0	149.100.2.1	G	Eth0
149.100.2.0	255.255.255.0	149.100.2.21		Eth0

8

Эта сеть содержит три подсети: 149.100.1.0/25 149.100.1.128/25, 149.100.2.0/24, которые объединены двумя маршрутизаторами M1 и M2. При этом маршрутизатор M2 является шлюзом в Интернет. Адреса интерфейсов маршрутизаторов показаны на рисунке. Предположим, что станция C6 имеет пакет с адресом назначения 149.100.1.159. Таблица маршрутизации C6 выглядит следующим образом (таблица внизу слайда):

Первая строка таблицы определяет закольцовывающий интерфейс. Вторая строка описывает маршрут по умолчанию и флаг G уточняет, что узел 149.100.2.1 является маршрутизатором. Третья запись таблицы не имеет флага H, следовательно, она определяет сеть; нет в ней и флага G и это говорит о том, что определяется маршрут к непосредственно подключенной сети и интерфейсом к ней является внешний интерфейс станции C6, имеющий адрес 149.100.2.21. Прежде всего C6 производит поиск адреса станции (или сети) назначения в своей таблице. Поскольку ни тот, ни другой в ней не присутствуют, то пакет будет отправлен в соответствии с маршрутом по умолчанию на маршрутизатор M2 с адресом 149.100.2.1 через интерфейс Eth0.

IP – маршрутизация



Пример сети, объединенной маршрутизаторами

Таблица маршрутизации **M2** :

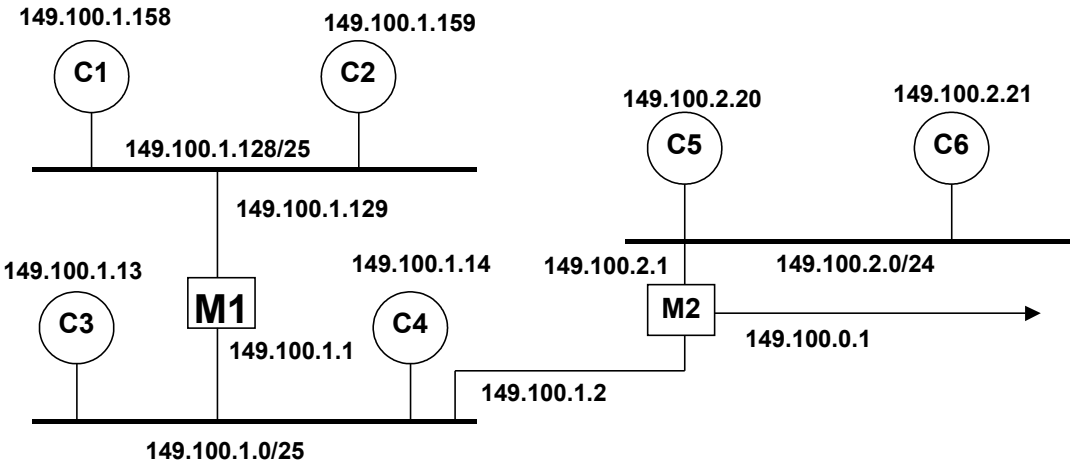
Адрес назначения	Маска	Следующий узел	Флаг	Интерфейс
127.0.0.0	255.0.0.0	127.0.0.1	H	lo0
default		149.100.0.1	G	Serial 01
149.100.2.0	255.255.255.0	149.100.2.1		Eth0
149.100.1.0	255.255.255.192	149.100.1.2		Eth1
149.100.1.128	255.255.255.192	149.100.1.1	G	Eth1

9

Таблица маршрутизации M2 может иметь следующий вид (внизу слайда).

Выполнив поиск в этой таблице маршрутизатор M2 найдет, что наиболее полно согласуется с адресом назначения маршрут, определенный в строке 5 и отправит пакет к маршрутизатору M1 через интерфейс Eth1.

IP – маршрутизация



Пример сети, объединенной маршрутизаторами

Таблица маршрутизации **M1** :

Адрес назначения	Маска	Следующий узел	Флаг	Интерфейс
127.0.0.0	255.0.0.0	127.0.0.1	H	lo0
default		149.100.1.2	G	Eth1
149.100.2.0	255.255.255.0	149.100.1.2	G	Eth1
149.100.1.0	255.255.255.192	149.100.1.1		Eth1
149.100.1.159	255.255.255.255	149.100.1.159		Eth0
149.100.1.128	255.255.255.192	149.100.1.129		Eth0

10

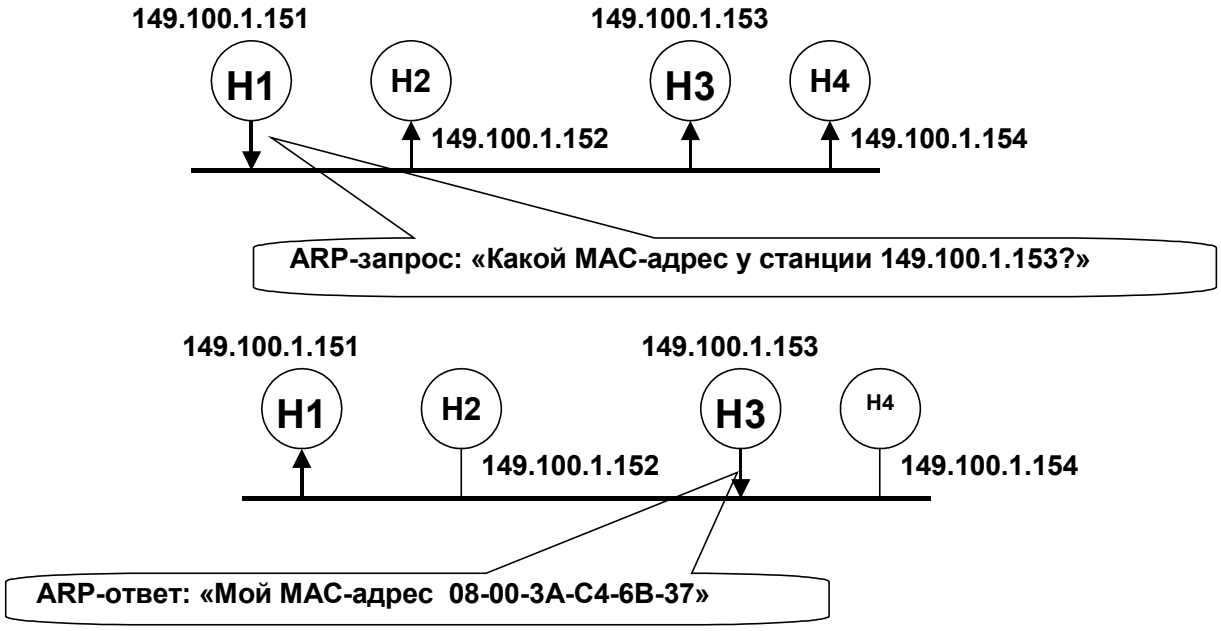
Таблица маршрутизации M1 представлена внизу в таблице.

В ней представлен маршрут с адресом назначения точно соответствующим адресу назначения пакета. Поэтому последний передается на интерфейс Eth0 и доставляется станции C2 с IP-адресом 149.100.1.159.

Разрешение IP-адресов в физические адреса сетевых устройств. Протокол ARP.

Для доставки IP-пакета к станции назначения, или от одного маршрутизатора к другому, необходимо передать его протоколу канального уровня, например Ethernet. Последний же «умеет» передавать кадры только по физическим адресам устройств, подключенных к среде передачи. В IP-сетях задачу преобразования сетевых адресов в физические решает протокол ARP (Address Resolution Protocol). Идея его функционирования иллюстрируется следующий слайд.

ARP (Address Resolution Protocol)



Разрешение IP-адреса в локальной сети

11

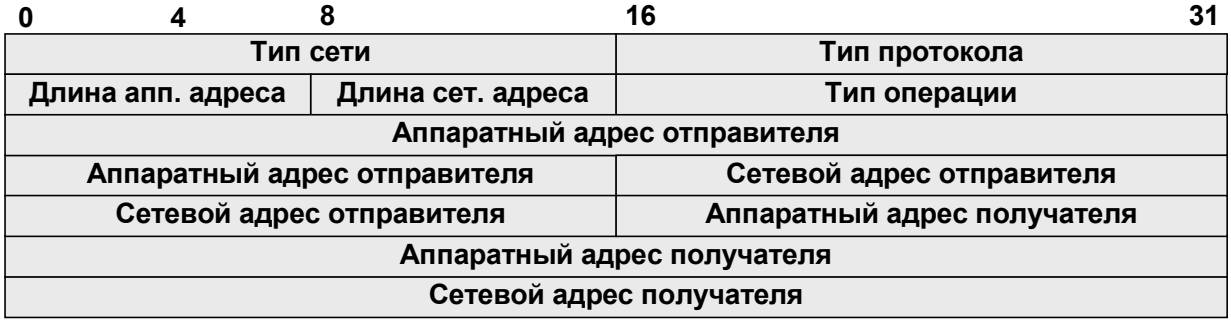
Пусть хост H1 хочет отослать пакет хосту H3, MAC-адрес которого не известен. Хост H1 генерирует так называемый ARP-запрос – специальный пакет, имеющий широковещательный адрес назначения. В теле этого запроса находится IP-адрес хоста, MAC-адрес которого необходимо узнать. Каждый хост сети получив такой пакет, сравнивает находящийся в нем IP-адрес со своим. Если совпадение обнаружено, то этот хост посылает запрашивающей станции ответный пакет (ARP-ответ), содержащий его физический адрес. На всех остальных станциях сети пакеты ARP-запросов уничтожаются. Для того, чтобы уменьшить количество ARP-запросов, каждое сетевое устройство имеет специальную буферную память, в которой хранится ARP-таблица. Последняя пополняется каждый раз, когда хост получает ARP-ответ.

В ARP-таблице могут быть как статические, так и динамические записи. Статические записи добавляются администратором и сохраняются в таблице до перезагрузки устройства. Кроме того, в таблице всегда содержится широковещательный адрес (%FFFFFFFFFFFF), который позволяет принимать широковещательные запросы. Динамические записи добавляются и удаляются автоматически. Каждая такая запись имеет потенциальное время жизни. После добавления записи в таблицу включается специальный таймер и, если в течении первых двух минут запись не используется, то она удаляется; в противном случае, время жизни такой записи составляет некое предустановленную величину (обычно 10 минут). Далее приведен фрагмент ARP-таблицы маршрутизатора.

ARP (Address Resolution Protocol)

IP-address	Port	Type	Media Address	Address Source
20.0.0.0	2	Broadcast	%FFFFFFFFFFFF	Static
10.0.0.0	1	Broadcast	%FFFFFFFFFFFF	Static
10.0.0.10	1	Local	%080002145DE6	Static
20.0.0.10	2	Local	%080002145DE5	Static
10.0.0.1	1	External	%080002A56BC0	ARP
20.0.0.1	2	External	%080002A719DD	ARP

Фрагмент ARP-таблицы маршрутизатора



Формат ARP-пакета

12

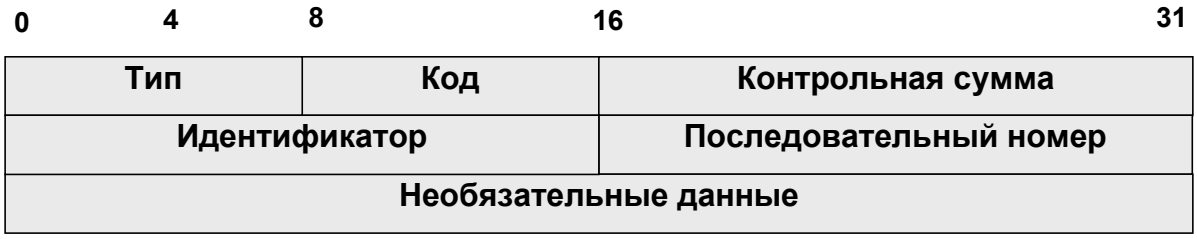
Протокол ARP достаточно универсален, и его можно применять в сетях, использующих любые технологии на сетевом и канальном уровнях. Заголовок ARP-пакета имеет формат, отличный от IP-заголовка и поэтому не передается маршрутизаторами. На рисунке внизу представлен формат сообщения ARP.

В поле «Тип сети» для Ethernet указывается значение 1. Для других типов сетей его значения определяются соответствующим стандартом. Поля «Тип протокола», «Длина аппаратного адреса» и «Длина сетевого адреса» обеспечивают отмеченную выше универсальность формата ARP-пакета. В поле «Тип операции» для ARP-запроса указывается 1, для ARP-ответа – 2.

Устройство, отправляющее ARP-запрос, заполняет в этом пакете все поля, кроме искомого аппаратного адреса. Значение этого поля заполняется станцией, опознавшей свой IP-адрес, указанный в этом пакете в поле «IP-адрес получателя».

ICMP – протокол обмена управляющей информацией

Тип	Содержание	Тип	Содержание
0	Ответ на эхо	11	Превышено время жизни пакета
3	Получатель недостижим	12	Ошибка параметров в пакете
4	Подавление источника	17	Запрос маски адреса
5	Изменение маршрута	18	Ответ на запрос маски адреса
8	Запрос эха		



Формат сообщений «Запрос эха» и «Ответ на эхо»

Выше упоминалось, что если маршрутизатор не может по каким-то причинам отправить пакет к узлу назначения, то он отсылает соответствующее сообщение узлу-отправителю. Эти функции выполняет протокол ICMP – Internet Control Message Protocol, описание которого приведено в документе RFC 792. Хотя его сообщения инкапсулируются в IP-пакет, протокол ICMP является протоколом сетевого уровня. Рассматриваемый протокол обеспечивает взаимодействие между программным обеспечением протокола IP, функционирующим на разных сетевых узлах. Однако, протокол не способен информировать промежуточные узлы о возникших ошибках, поскольку в IP-пакете нет поля для записи маршрута. Соответственно, когда пакет пришел на некий маршрутизатор и в ходе его обработки обнаружилась необходимость отправки сообщения ICMP, то единственным получателем такого сообщения будет узел – отправитель исходного пакета.

Протокол ICMP генерируют два вида сообщений: управляющие и сообщения об ошибках, но он не содержит никаких средств исправления ошибок; эта задача решается программным обеспечением узла-отправителя.

Сообщения ICMP начинаются тремя обязательными полями: «Тип», «Код» и «Контрольная сумма». Кроме этого, большинство сообщений включают в себя заголовок и первые 64 бита пакета, в котором была обнаружена ошибка, сообщение о которой несет данный пакет ICMP. Это сделано для более точной идентификации источника ошибки на узле-отправителе. Поле «Тип» определяет содержание сообщения и его формат. На слайде приведены некоторые значения этого поля.

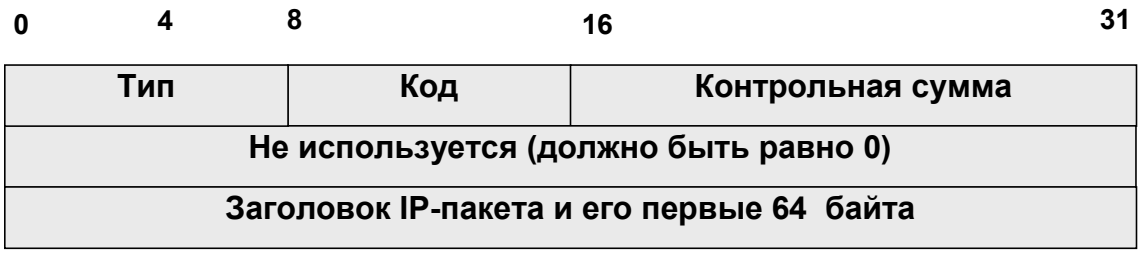
Сообщения «Запрос эха» и «Ответ на эхо» являются одними из самых используемых (команда **ping**). Поскольку эти сообщения передаются в IP-пакетах, то успешный прием ответа свидетельствует о работоспособности основных компонентов сетевой транспортной системы, т.е. успешной

маршрутизации, работоспособности IP-модулей стека протоколов на станциях отправителе и получателе. На рисунке внизу слайда представлен их формат.

Поля «Идентификатор» и «Последовательный номер» используются отправителем для проверки соответствия между ответом и запросом. Поле «Необязательные данные» имеет переменную длину и содержит данные, которые необходимо вернуть отправителю.

ICMP – протокол обмена управляющей информацией

Значение «Код»	Причины	Значение «Код»	Причины
0	Сеть недостижима	5	Ошибка при маршрутизации от источника
1	Устройство недостижимо	6	Сеть назначения неизвестна
2	Протокол недостижим	11	Сеть недостижима из-за класса обслуживания
3	Порт недостижим	12	Устройство недостижимо из-за класса обслуживания
4	Требуется фрагментация		



Формат сообщения «Получатель недостижим»

14

Сообщение «Получатель недостижим» генерируется маршрутизатором, в ситуации, когда он не может доставить пакет по назначению. Это сообщение содержит поле «Код», которое уточняет причину невозможности доставки пакета. Некоторые из них представлены в таблице.

Сообщение «Изменение маршрута» (***Слайд 13***) отправляется маршрутизатором отправителю, находящемуся с ним в одной физической сети, и свидетельствует об использовании последним неоптимального маршрута к узлу-назначения. Это сообщение содержит адрес маршрутизатора, использование которого является предпочтительным. Таким образом, начиная работу, отправитель может знать лишь один маршрут. В дальнейшем, его таблица маршрутизации, посредством этих сообщений ICMP, будет дополнена более точной информацией об «оптимальных» маршрутах. Весь этот механизм позволяет узлу отправителю минимизировать размер своей таблицы маршрутов.

Обработка IP-пакетов маршрутизатором.

Можно выделить две основные функции маршрутизатора:
1. определение оптимального маршрута пакета и
2. коммутацию пакетов с одного физического интерфейса на другой.

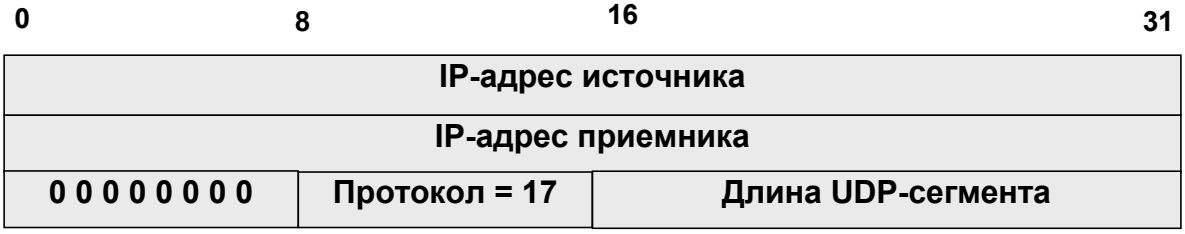
Определение маршрута, как правило, реализуется программным образом, а передача (коммутация) пакетов с одного интерфейса на другой в современных маршрутизаторах выполняется аппаратно. Программное обеспечение маршрутизатора включают в себя набор протоколов маршрутизации, процедуры управления базой данных (маршрутной таблицей) и процедуры поддержки определенных сервисов (фрагментация, фильтрация и т.п.).

Протокол UDP.

Два протокола транспортного уровня, UDP и TCP, обеспечивают IP-сетям механизмы взаимодействия прикладных процессов, выполняющихся на конечных станциях. Напомним, что собственно протокол IP «умеет» доставлять пакеты данных взаимодействующим хостам, но не «знает» как обеспечить взаимосвязь приложений и не имеет почти никаких средств обеспечения надежности доставки сообщений - он проверяет лишь целостность заголовка пакета.

Протокол UDP

Формат UDP-сегмента



Формат псевдозаголовка UDP-сегмента

15

Протокол UDP (User Datagram Protocol) описан в документе RFC 768. Он ориентирован на сервис без установления соединений и не обеспечивает надежную передачу сегментов между сетевыми приложениями. Это очень простой протокол, который развивает возможности IP-протокола лишь в части демультиплексирования потока пакетов по признаку принадлежности их определенному приложению и контроля целостности данных. Взаимодействие между прикладными процессами UDP реализует посредством механизма **протокольных портов**. Протокольный порт можно определить как абстрактную точку присутствия конкретной прикладной программы, выполняющейся на конкретном хосте. Когда рабочая станция получает пакет, в котором указан ее IP-адрес, она может направить его определенной программе, используя уникальный номер порта, назначенный этой программе в ходе выполнения процедуры установления соединения. Таким образом, в стеке протоколов TCP/IP порт

является механизмом поддержания рабочей станцией одновременного выполнения нескольких прикладных процессов.

Каждый порт (прикладной процесс) идентифицируется целым положительным числом (номером порта). Номера портов приложения, выполняющегося на разных станциях, указываются в заголовке UDP-сегмента. Эта информация дополняется на сетевом уровне IP-адресами взаимодействующих станций. Благодаря этому, создается видимость непосредственного обмена данными между процессами.

Сегмент данных протокола UDP (иногда его называют пользовательской дейтограммой) состоит из двух частей: заголовка и области данных (см. слайд). Заголовок имеет четыре 16-битных поля, определяющих порт отправителя, порт получателя, длину сегмента и контрольную сумму.

Поле «Длина UDP-сегмента» содержит количество байтов в дейтограмме с учетом длины ее заголовка.

Вычисление контрольной суммы дейтограммы UDP является опциональным. При работе в надежных локальных сетях она не вычисляется и тогда это поле заполняется нулями. Процедура подсчета контрольной суммы содержит две особенности. Первая состоит в дополнении дейтограммы нулевыми битами до размера, кратного 16. Это делается только на время вычисления контрольной суммы, и незначащие нули не передаются. Второй особенностью является дополнение, на период подсчета контрольной суммы, заголовка сегмента **псевдозаголовком.** Его формат представлен на рисунке внизу.

Псевдозаголовок включается перед заголовком дейтограммы; он имеет длину 12 байтов; поле «Протокол» содержит тип протокола сетевого уровня (IP, ICMP); его значение, как и значения полей с IP-адресами, должны быть извлечены из заголовка IP пакета. Поле «Контрольная сумма» на время ее вычисления заполняется нулями.

Такое дополнение дейтограммы выполняется как на передающей, так и на приемной станции и оно служит гарантией, что если контрольные суммы совпали, то дейтограмма достигла нужной станции и нужного порта. Еще раз подчеркну, что псевдозаголовок и дополнение нулями не передаются.

Если контрольные сумма, вычисленная приемником, не совпала с контрольной суммой, указанной в дейтограмме, то UDP-сегмент уничтожается и никаких уведомлений передающей станции об этом не передается.

Протокол TCP.

Протоколы TCP и IP являются «рабочими лошадками» Интернет. В этом разделе будут рассмотрены механизмы, посредством которых TCP обеспечивает надежный, ориентированный на установление соединений потоковый сервис в дейтограмной сети IP. В число этих механизмов входит вариант алгоритма ARQ с выборочным повторением и алгоритм управления перегрузками на базе индикации потерь сегментов. Подробно протокол описан в документах RFC 793 и RFC 1122.

Надежный потоковый сервис.

Задачей протокола TCP является организация и поддержание логического полнодуплексного соединения между двумя приложениями, выполняющимися на конечных станциях, соединенных ненадежной дейтограмной сетью, которое обеспечивает обмен упорядоченным потоком байтов и управление его интенсивностью. Дуплексность соединения позволяет вести передачу данных в одном из направлений и после того, как соединение противоположного направления было закрыто.

Аналогично UDP, каждый прикладной процесс для TCP-модуля представляется номером порта. Структура из пары переменных (порт, IP-адрес) называется **сокетом.** Соединение между отправителем и получателем однозначно определяется двумя сокетами. Для хранения всей информации, необходимой для установления и поддержания соединения, определена специальная структура данных – **блок управления передачей (Transmission Control Block - TCB)**. В эту структуру, кроме двух сокетов, входят флаги безопасности и приоритета соединения, указатели буферов отправителя и получателя, указатели номеров очередного сегмента и сегмента повторной посылки, а также ряд других переменных.

TCP не сохраняет границы сообщений и рассматривает данные, которые поступают ему от приложения, как поток байтов. Свои сегменты он формирует так, как считает необходимым, но с учетом свойств протокола сетевого уровня (сегмент должен полностью поместиться в IP-пакет). Таким

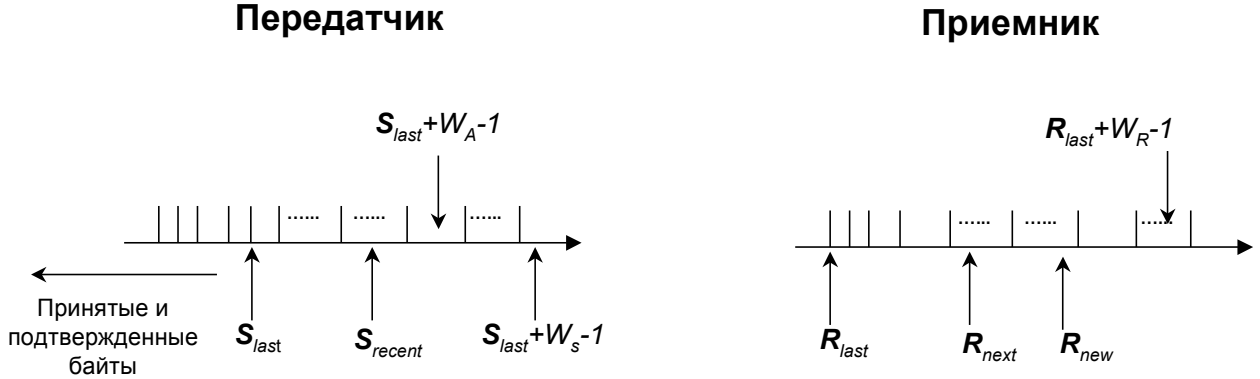
образом, если приложение отсылает сообщение, размер которого составляет 1000 байтов, то на приемной стороне оно может быть представлено двумя частями по 500 байт, тремя частями по 300, 300 и 400 байт и.т.д.

Адаптационные механизмы протокола.

Надежный потоковый сервис протокол TCP обеспечивает посредством специальных адаптационных механизмов. Напомним, что этот протокол предполагает наличие сетевой среды, в которой пакеты могут теряться, дублироваться, приходить в узел назначения с нарушением порядка их отправки. Для идентификации таких нарушений в протоколе введена последовательная нумерация **байтов** сегмента. Порядковый номер первого байта сегмента передается в его заголовке и он считается порядковым номером сегмента. Номер первого байта для каждого **соединения** выбирается случайным образом в период установления соединения. Поскольку каждый байт сегмента оказывается пронумерованным, то легко решается задача опознания на приемной стороне нарушений порядка их доставки. Механизм подтверждения приема носит накопительный характер, т.е. подтверждение приема байта с номером N, означает, что байты с номерами N-1, N-2, … успешно получены. Диапазон возможных номеров байтов ограничен числами от 0 до 2^{31}-1. Заметим, что при скорости передачи в 100 Мбит/с полный цикл использования всего допустимого диапазона номеров составляет 5,4 минуты, что вполне достаточно для того, чтобы любой сегмент был получен приемной станцией, а все его дубли были уничтожены.

Протокол TCP

Slast **- указатель наименьшего номера неподтвержденных байтов**
Sresent **- указатель номера последнего из отправленных байтов**
Slast+Ws–1 **– указатель наибольшего номера байта**
который передающий модуль может принять от приложения (но не отправить в сеть)
Rlast **- указатель наименьшего номера байта, который еще не был передан приложению**
Rnext **- указатель номера следующего ожидаемого к приему байта**
Rnew **- указатель наибольшего номера корректно принятого байта.**



Окно передачи и приема в TCP-модулях

16

Передача данных в протоколе TCP регулируется на основе алгоритма скользящего окна.

Окно передачи определяется тремя указателями:

- S_{last} - указатель наименьшего номера неподтвержденных байтов,
- S_{resent} - указатель номера последнего из отправленных байтов,

- $S_{last}+W_s-1$ – указатель наибольшего номера байта, который передающий модуль может принять от приложения (но не отправить в сеть).

Заметим, что передатчик отправит сегмент в сеть лишь после получения от приложения некоторого числа байтов, превышающего заранее установленный порог, либо после истечения определенного интервала времени. При этом, величина сегмента должна быть такой, что наибольший номер его байта, не должно превосходить величину $S_{last}+W_A-1$, где W_A - это так называемое объявляемое окно, параметр определяемый размером свободного буферного пространства приемного приложения.

Величина окна передачи (W_s) является механизмом адаптации интенсивности передатчика к требованиям сети и изменяется под воздействием целого ряда факторов.

В свою очередь, приемный модуль протокола TCP формирует свое окно, которое также управляется тремя параметрами:

- R_{last} - указатель наименьшего номера байта, который еще не был передан приложению
- R_{next} - указатель номера следующего ожидаемого к приему байта,
- R_{new} - указатель наибольшего номера корректно принятого байта.

Отметим, что R_{new} может оказаться больше R_{next}. Такая ситуация является следствием того, что приемный модуль не отбрасывает байты, принятые в пакетах, не содержащих ошибок, но пришедших с нарушением порядка их следования. Если размер буферной памяти приемного модуля равен W_R байт (максимальный размер приемного окна), то всегда существует возможность принять байты с номерами, не превышающими $R_{last}+W_R-1$. При этом, мы предполагаем, что приложение не обязательно читает байты немедленно после их приема.

Прибывший сегмент проверяется на целостность данных и при положительном результате такой проверки его байты, если их число не превышает размер W_R, записываются в соответствующую область памяти приемного буфера. Если принятый сегмент имеет номер R_{next}, то значение этого указателя обновляется, а передающему приложению отсылается положительное подтверждение с новым номером $R_{next.}$, что свидетельствует об успешном приеме всех сегментов с порядковыми номерами меньшими R_{next}. Это подтверждение используется передающим модулем для обновления значения своего указателя S_{last} и, тем самым, передающее окно перемещается вперед, позволяя отправлять сегменты с более высокими порядковыми номерами.

TCP располагает механизмом **управления потоком**, предотвращающим переполнение приемного буфера. Этот регулятор представляет собой, так называемое, **объявляемое окно (advertised window) –** специальное поле в заголовке сегмента, передаваемого от приемника к передающему узлу. Это поле информирует передающий модуль о доступной величине буферной памяти приемника. Величина объявляемого окна определяется выражением

$$W_A = W_R - (R_{new} - R_{last}).$$

Получив значение W_A, передающий модуль может из накопленных в его буфере данных сформировать сегмент, величина которого не превышает W_A, т.е.

$$S_{recent} - S_{last} \le W_A.$$

Для иллюстрации эффекта регулирования интенсивности потока, генерируемого приложением, рассмотрим ситуацию, когда приложение на приемной стороне прекращает читать данные из буфера своего модуля TCP. Ясно, что величина R_{new} будет увеличиваться, а R_{last} оставаться постоянной; такая динамика приводит к постоянному уменьшению величины объявляемого окна и передающий модуль TCP вынужден уменьшать размер отправляемых сегментов. В конце концов, W_A станет равным нулю и передающий модуль TCP прекратит отправку данных. Однако, запись данных приложения в буфер передающего модуля будет продолжаться до тех пор, пока он не примет W_s байт. В этот момент работа приложения будет полностью заблокирована.

Надежность передачи данных обеспечивается применением алгоритма повторной передачи с выборочным повторением. Когда протокол станции-отправителя передает сегмент с номером M, он помещает его копию в очередь повторной передачи и запускает специальный таймер повторной передачи; при получении подтверждения о благополучном приеме сегмента M, его копия уничтожается. В противном случае, после истечении таймера повторной передачи модуль TCP снова отправит этот

сегмент в сеть. Время доставки сегментов в сетевой среде, в которой работает протокол TCP (Интернет, например), может изменяться в значительных пределах даже в течении одного соединения. Поэтому выбор значения таймера повторной передачи представляет собой непростую задачу. В протоколе TCP для этого предусмотрена адаптивная процедура, базирующаяся на постоянном измерении интервала времени от посылки сегмента до получения квитанции о его приеме (τ_n). Эти значения непрерывно усредняются с некими весовыми коэффициентами, уменьшающимися от последнего к предыдущему замером. В результате получают значение параметра **«полное время оборота»** (**round-trip time**, t_{RTT}).

$$t_{RTT}(new) = \alpha t_{RTT}(old) + (1 - \alpha)\tau_n,$$

где $0 < \alpha \le 1$, типичное значение $\alpha = 7/8$. Значение таймера повторной передачи (t_{out}) учитывает также дисперсию значения t_{RTT}:

$$t_{out} = t_{RTT} + k\sigma_{RTT},$$

где k некоторая константа. Таким образом, если девиация значений полного времени оборота будет значительная, то и значения таймера повторной передачи будут увеличены в сравнении со средним значением t_{RTT}. Поскольку вычисление значения стандартного уклонения относительно трудоемко, то в практически используемых процедурах оценка вариации значений t_{RTT} базируется на вычислении абсолютного уклонения $|\tau_n - t_{RTT}|$:

$$t_{out} = t_{RTT} + 4d_{RTT},$$

$$d_{RTT}(new) = \beta d_{RTT}(old) + (1 - \beta)|\tau_n - t_{RTT}|,$$

при типичном значении $\beta = 0.25$.

Функционирование протокола TCP.

Протокол TCP

Формат TCP-сегмента

В этом разделе будут рассмотрены структура сегмента TCP, механизмы установления соединения, передачи данных и ликвидации соединения. Формат TCP сегмента представлен на слайде. Его заголовок содержит 20-байтную фиксированную часть и опциональную часть переменной длины.

«Порт источника» и **«Порт назначения» -** определяют передающее и приемное приложения, соответственно.

«Порядковый номер сегмента» - определяет позицию первого байта данных сегмента в байтовом потоке источника при значении флага SYN=0 (в режиме передачи данных). Напомним, что TCP нумерует байты, а не сегменты и если порядковый номер текущего сегмента равен 567, а поле данных содержит 12 байт, то следующий сегмент будет иметь порядковый номер 579. В режиме установления соединения, когда флаг SYN установлен в 1, в этом поле содержится начальный номер последовательности номеров байтов данного потока **(ISN – initial sequence number**); значение номера первого байта данных этого потока будет ISN+1. Отметим также, что соединения TCP являются дуплексными и в каждом из направлений передачи устанавливается своя нумерация.

«Порядковый номер подтверждения» - это поле в режиме с установленным флагом ACK (режим передачи данных) содержит порядковый номер байта данных, который передающий модуль ожидает получить от приемного узла; тем самым подтверждается правильность приема всех предыдущих байтов. В режиме установления соединения (ACK=0) значение этого поля не учитывается.

«Смещение данных» - поле определяет длину заголовка сегмента в 32-битных словах; эта информация позволяет приемному модулю определить начало поля данных, т.к. заголовок может содержать опциональное поле переменной длины.

«Резерв» - поле в настоящее время не используется и заполняется нулями.

«Контрольные биты» - поле длиной 6 бит, каждый из которых является флагом; их последовательность и смысл следующие:

URG – флаг срочности передачи сегмента

ACK – флаг указывающий на достоверность значений в поле **«Порядковый номер подтверждения»**

PSH – включена функция «проталкивания» сегмента, т.е. модуль TCP должен передать сегмент приложению немедленно

RST – указание приемному модулю разорвать соединение по причине каких-то аномалий; используется для перезагрузки соединения

SYN – флаг установления соединения, синхронизации порядковых номеров сегментов

FIN – флаг, индицирующий, что у передающего модуля нет данных для передачи; передающее приложение остается в соединении с приемным и принимает данные последнего.

«Размер окна» - поле опрделяет количество байтов, которое модуль TCP может принять *(W_a).*

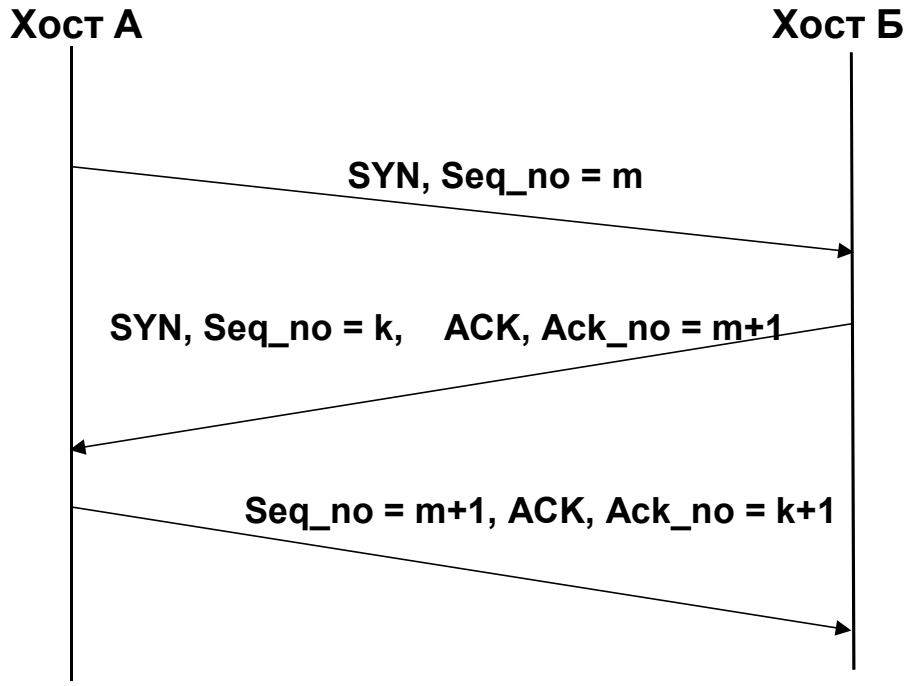
«Контрольная сумма» - значение этого поля рассчитывается по всему сегменту с дополнением его нулями до размера кратного 16 битам и 96 битным псевдозаголовком, включаемым перед заголовком TCP и содержащим сетевые адреса отправителя и получателя, тип протокола и длину TCP сегмента. Эти дополнения используются только для расчета контрольной суммы и не передаются.

«Указатель срочности» - значение этого поля при установленном флаге URG, будучи добавленным к значению поля **«Порядковый номер сегмента»,** определяет последний байт срочных данных. Поскольку приемный модуль TCP передает приложению байты строго по порядку, то все байты, содержащиеся в приемном буфере, вплоть до байта с определенным, как указано выше номером, будут рассматриваться как срочные.

«Опции» - поле используется для определения других, не предусмотренных заголовком, функций. Так например, это поле часто используется для определения **максимального размера сегмента** (maximum segment size - MSS). При использовании протокола в высокоскоростных сетях это поле используется для задания таких параметров как **«Коэффициент масштабирования окна»** (до 2^{14}) и **«Временная метка».** Последние важны в ситуации когда полный цикл нумерации байт может быть пройден за время существования соединения. Наличие временных меток в каждом сегменте позволяет также вычислить время полного оборота (*RTT*).

Теперь рассмотрим работу протокола в различных фазах жизни соединения.

Протокол TCP



Процедура установления TCP-соединения

18

Фаза установления соединения.

Фаза установления соединения, предшествующая фазе передачи данных, содержит следующие действия.

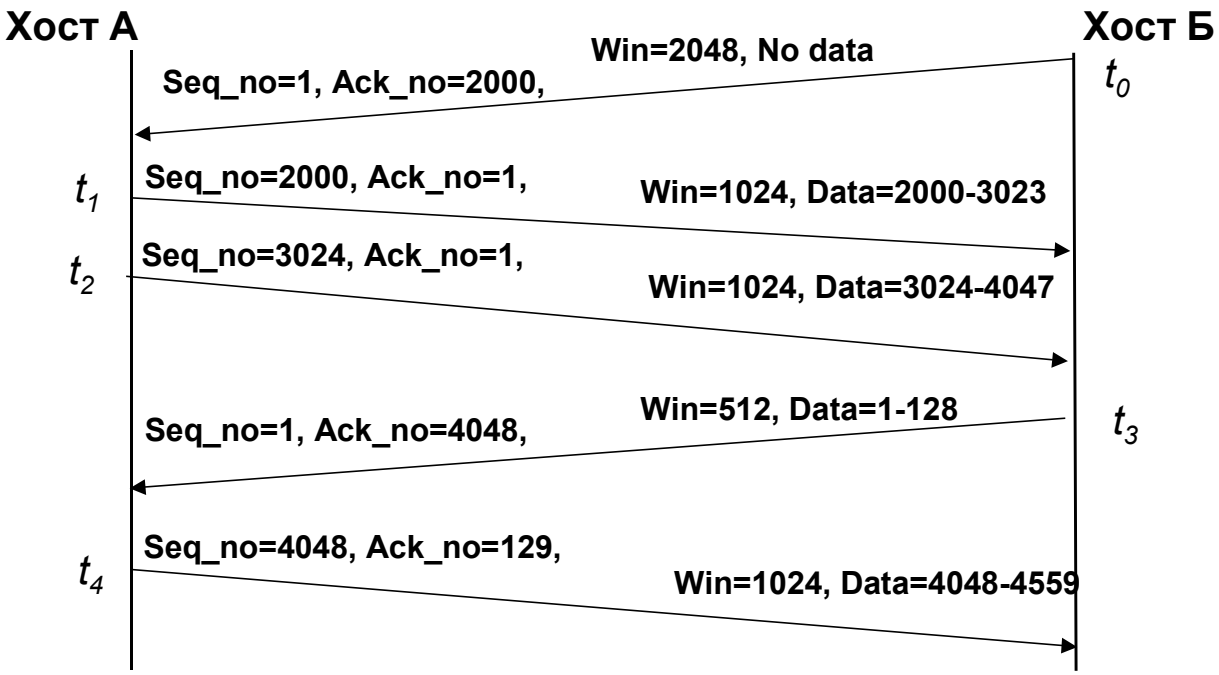
1. Хост **А** отправляет хосту **Б** запрос соединения посредством установки флага SYN и инициализирует значение начального номера нумерующей последовательности (Seq_no = m).
2. Хост **Б** отвечает на этот запрос установкой флага ACK и определяет поле «**Порядковый номер подтверждения»** значением на единицу большим m (Ack_no = m+1); одновременно, хост **Б** в своем ответе **А** отправляет запрос соединения (SYN) и также инициализирует значение начального номера своей нумерующей последовательности (Seq_no = k).
3. Хост **А** отвечает на запрос соединения от хоста **Б** установкой флага ACK и подтверждением ожидания следующего байта данных с порядковым номером k+1 (Ack_no = k+1); при этом, значение поля «**Порядковый номер сегмента»** устанавливается в значение m+1 (Seq_no = m+1).

Такая трехэтапная процедура установления соединения гарантирует согласование начальных значений нумерующих последовательностей взаимодействующих TCP-модулей, что принципиально важно для последующего функционирования соединения. Случайный характер выбора начальных значений Seq_no является достаточно надежной мерой, предупреждающей установление на обоих концах соединения одинаковых начальных номеров.

Заметим, что в фазе установления соединения каждый SYN-сегмент может содержать опциональные параметры и любой хост может отказать в соединении посредством отсылки сегмента с установленным флагом RST.

Протокол TCP поддерживает два типа соединения – активное и пассивное. Так, в приложениях, построенных по клиент-серверной архитектуре, сервер выполняет пассивное соединение (формирует свой сокет и переходит в режим «прослушивания»), сообщая тем самым своему модулю TCP о готовности принять запрос соединения. Когда клиент желает установить связь с сервером, он выполняет процедуру активного соединения. В нее входит создание сокета на клиентской стороне и выполнение описанных выше процедур TCP-соединения.

Протокол TCP



Передача данных по TCP-соединению

19

Фаза передачи данных.

Предоставление приложениям сервиса надежной доставки данных в протоколе TCP обеспечивается использованием алгоритма ARQ с выборочным повторением и механизма скользящего окна. При этом, особенностью протокола TCP является реализация скользящего окна не на уровне сегментов, а на уровне байтов. Протокол также обеспечивает управление потоком в фазе передачи данных посредством регулирования величины объявляемого окна и величины окна передачи. Слайд иллюстрирует процесс управления интенсивностью потока.

Пусть в момент t_0 TCP-модуль хоста **В** объявил величину своего окна равной 2048 байт и номер следующего ожидаемого байта 2000. Такой размер окна позволяет хосту **А** отправить без подтверждения 2 Кбайта данных, однако в его выходном буфере имеется лишь 1024 байта данных. Хост **А** отправляет эти данные нумеруя байты начиная с 2000. Одновременно, он объявляет величину своего окна равной 1024 байта и подтверждает, что номер ожидаемого первого байта от хоста **Б** должен быть равен 1. Хост **Б** задерживает выдачу подтверждения на прибывший сегмент данных, полагая, что у него

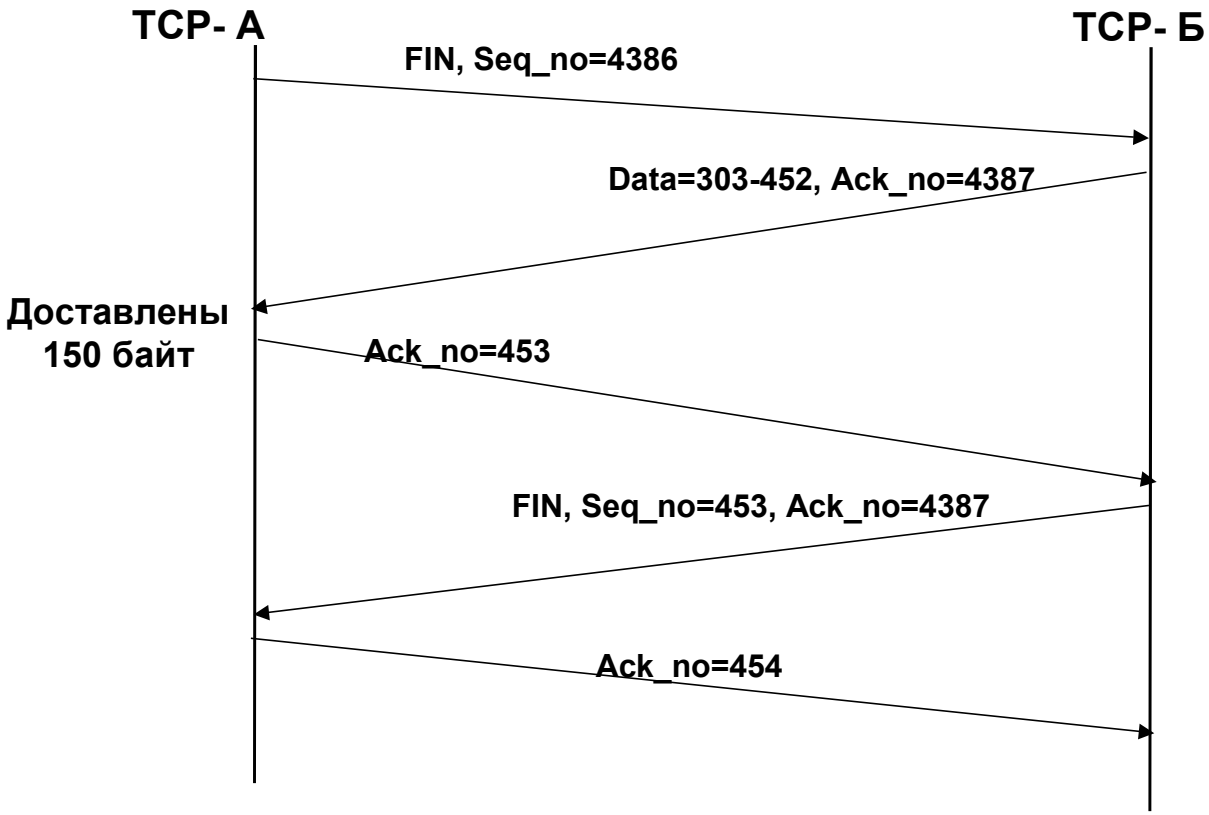
появятся данные для отправки хосту **А,** вместе с которыми он отправит и подтверждение. Тем временем, в момент t_2 модуль TCP хоста **А** снова получил от своего приложения 1024 байт данных и передал их хосту **Б**. После этого величина окна отправки на хосте **А** стала равной нулю и дальнейшая отправка им данных до получения подтверждения от хоста **Б** оказывается невозможной. В момент t_3 модуль TCP хоста **Б** получил 128 байт данных для отправки; вместе с ними он отправляет подтверждение получения от хоста **А** двух сегментов данных, указывая Ack_no=4048. К этому моменту в буферной памяти модуля TCP хоста **Б** оказывается свободными лишь 512 байт, поэтому он объявляет величину своего окна приема равной 512. Когда хост **А** получит этот сегмент, он установит величину окна отсылки равной 512 байт и, несмотря на то, что в момент t_4 в его буфере имеется 2048 байт данных, он сможет отослать только 512 байт и не перегрузит буфер хоста **Б**.

Предыдущее рассмотрение показывает, что задержка отправки подтверждения позволяет более экономно использовать пропускную способность канала. Это особенно актуально в ситуациях, подобных login-сессиям, когда пользователь вводит свои аутентификационные данные по одному символу. Пересылка каждого из них, сама по себе весьма затратна (на один байт информации приходится не менее 40 байтов заголовков TCP и IP уровней), а выдача подтверждения на каждый такой сегмент еще более усугубляет ситуацию. Решение, экономящее пропускную способность канала связи, было предложено Нейглом (Nagle). Идея алгоритма достаточно проста. Когда интерактивное приложение требует отсылки символа, модуль TCP отправляет его и ждет подтверждения. Поступающие в этот период времени от приложения символы не передаются, а буферизируются; они отправляются все вместе в одном сегменте после получения подтверждения на первый отправленный символ. В локальных сетях, где задержки доставки сегментов относительно малы и каналы связи являются достаточно широкополосными, применение рассмотренного алгоритма является нецелесообразным.

Существует еще одна ситуация, в которой механизм регулирования окна протокола TCP может приводить к неэффективному использованию полосы пропускания. Пусть передающее приложение имеет большой объем данных для передачи, а принимающее приложение может читать буфер своего TCP модуля лишь небольшими порциями. Ясно, что, в конце концов, это приведет к объявлению приемным модулем малого значения окна; соответственно, передающий модуль будет формировать сегменты с малым объемом данных, что и увеличит «накладные» расходы протокола при их передаче. Это явление часто называют «синдромом ленивого окна». Для уменьшения его негативного влияния ограничивают возможность объявлять малые значения приемного окна. А именно, пока размер свободного буферного пространства приемника не достигнет половины его объема, либо половины размера максимально допустимого сегмента, размер приемного окна не объявляется (т.е. считается равным нулю). Данные приложения на передающей стороне буферизируются и отправляются после объявления ненулевого приемного окна уже достаточно большими сегментами.

Рассмотрим еще проблему зацикливания нумерующей последовательности, характерную для работы протокола TCP в высокоскоростных средах. Исходная спецификация протокола предполагала, что максимальное время жизни сегмента составляет 2 минуты. Последовательность из 2^{32} номеров способна пронумеровать 4294967296 байт. Для их передачи по линии с пропускной способностью 2048 Кбит/с потребуется $(2^{32} \times 8)/(2{,}048 \times 10^6) = 4{,}5$ часа, а по линии OC-48 (2,4 Гбит/с) всего 14 секунд. Ясно, что на современных высокоскоростных линиях протокол TCP может терять работоспособность. Решение этой проблемы было найдено посредством определения в опциональном поле заголовка сегмента 4-х байтовой временной метки. Приемный модуль включает ее в свой ACK-сегмент, и таким образом объединяются два способа разметки последовательности отправляемых сегментов. Это увеличивает период нумерующей последовательности до 2^{64}. Ясно, что для реализации этого механизма таймер временных меток должен изменять свое значение, по крайней мере, один раз за время необходимое для отправки 2^{31} байт (максимальное значение окна передачи). Это требование определяет нижнюю границу частоты этого таймера. Эта частота не должна быть и настолько высокой, чтобы полный цикл показаний таймера был пройден за время, меньшее максимального времени жизни сегмента. Значения в диапазоне от 1 Гц до 1 кГц удовлетворяют этим ограничениям. Например, при частоте в 1кГц протокол будет успешно работать на линиях с пропускной способностью до 8 Тбит/с. [RFC 1323].

Протокол TCP



Фаза ликвидации соединения.

Протокол TCP реализует процедуру поэтапной ликвидации соединения, предполагающую независимое его закрытие в обоих направлениях. Необходимость в закрытии соединения возникает, когда приложение сообщает своему модулю TCP об отсутствии у него данных для отправки. TCP модуль завершает передачу данных, находящихся в его буфере, ожидает получения подтверждения об их успешном приеме и отправляет приемному модулю сегмент с установленным флагом FIN. Получив этот сегмент, приемный модуль информирует свое приложение о завершении поступления данных от передающего приложения, но продолжает отсылать данные (если они есть) в противоположном направлении. Получив подтверждение на отправленные данные, модуль TCP отправляется сегмент FIN в противоположном направлении и, после получения на него подтверждения ACK, соединение считается ликвидированным.

Хост **А** инициализирует процедуру разрыва соединения, отправляя сегмент с флагом FIN. Модуль TCP хоста **Б** подтверждает прием этого сегмента и передает извещение о запросе на закрытие соединения своему приложению. Одновременно, располагая данными для хоста **А,** модуль TCP хоста **Б** отправляет сегмент со 150 байтами данных хосту **А**, и получает подтверждение их приема. Получив от своего приложения подтверждение разрыва соединения протокольный модуль хоста **Б** отправляет встречный сегмент FIN и получает на него подтверждение. Модуль TCP хоста **А** переходит в состояние ожидания и запускает таймер TIME_WAIT с начальным значением задержки равным удвоенному максимальному времени жизни сегмента. В этот период единственным сегментом, который может прийти на хост **А**, является повторный сегмент FIN от хоста **Б** (если соответствующий сегмент ACK от хоста **А** был утерян). Если такой сегмент приходит, то хост **А** повторно отсылает сегмент ACK и вновь перезапускает таймер TIME_WAIT. При достижении этим таймером значения нуль, хост **А** ликвидирует соединение и удаляет запись о нем из таблицы соединений.

Состояние ожидания обеспечивает выполнение еще одной задачи, а именно, оно защищает будущие реализации соединения между этими же прикладными процессами от обработки задержавшихся в сети сегментов предыдущего соединения. За двойное время жизни все, не доставленные сегменты этого соединения, будут уничтожены.

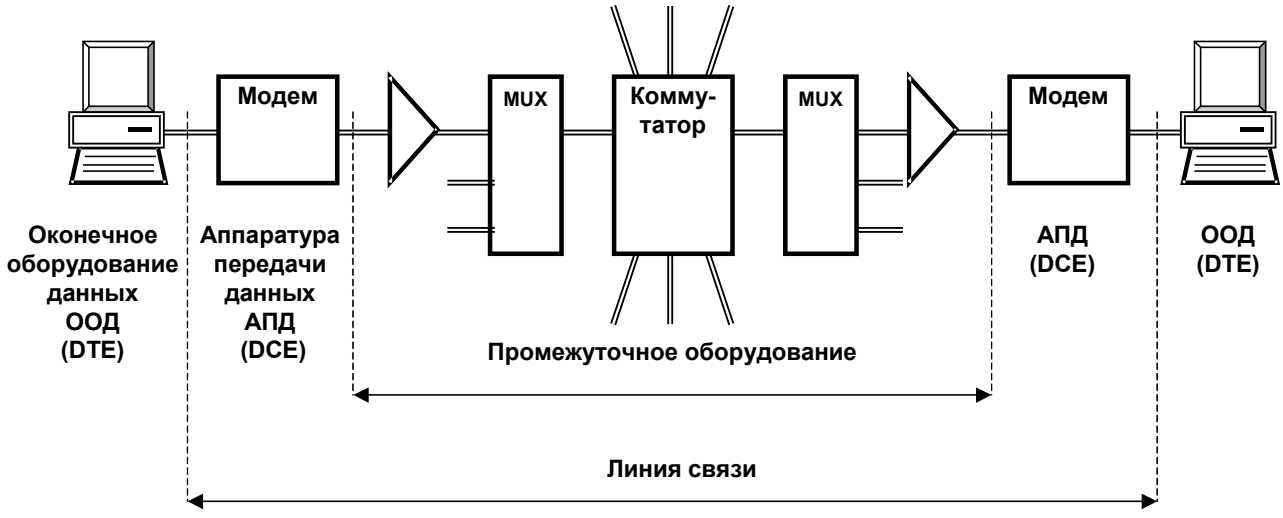
Протокол TCP располагает также механизмом срочной ликвидации соединения посредством отправки RST-сегмента. Отправка этого сегмента является на получение сегмента, адресованного приложению, которого на данном порте нет. Передающий модуль TCP, получив сегмент RST, уничтожает все данные, находящиеся в его буфере. Приемный модуль TCP, получив сегмент RST, информирует о ликвидации соединения соответствующий прикладной процесс.

Лекция 4. Основы передачи дискретных данных.

Любая сетевая технология должна обеспечить надежную и быструю передачу дискретных данных на физическом уровне по линиям связи. И все сетевые технологии при всех их отличиях базируются на общих принципах передачи дискретных данных. Эти принципы находят свое воплощение в:

- методах представления двоичных сигналов (т.е. нулей и единиц) с помощью импульсных или синусоидальных сигналов в линиях связи различной физической природы,
- методах обнаружения и коррекции ошибок,
- методах компрессии и
- методах коммутации.

Линии связи



Состав линии связи

2

Линии связи.

Линия или канал связи состоит в общем случае из:

- физической среды (media), по которой передаются электрические сигналы,
- аппаратуры передачи данных,
- промежуточной аппаратуры.

В зависимости от среды линии делятся на:

- проводные или воздушные,
- кабельные (медные и волоконно – оптические),
- радиоканалы наземной и спутниковой связи.

Проводные (воздушные) линии - провода без каких-либо изолирующих или экранирующих оплеток, проложенные между столбами и висящие в воздухе. Традиционно передают телефонные или телеграфные сигналы, но при отсутствии других возможностей эти линии используются и для передачи компьютерных данных. Скоростные качества и помехозащищенность плохие. Сегодня быстро вытесняются кабельными.

Кабельные линии - проводники, заключенных в несколько слоев изоляции: электрической, электромагнитной, механической, а также, возможно, климатической. Три основных типа кабеля:

- кабели на основе скрученных пар медных проводов,
- коаксиальные кабели с медной жилой,
- волоконно-оптические кабели.

Витая пара существует в экранированном варианте (Shielded Twisted Pair, STP), когда пара медных проводов обертывается в изоляционный экран, и неэкранированном (Unshielded TwistedPair, UTP), когда изоляционная обертка отсутствует. Скручивание проводов снижает влияние внешних помех на сигналы.

Коаксиальный кабель - несимметричная конструкция и состоит из внутренней медной жилы и оплетки, отделенной от жилы слоем изоляции. Существует несколько типов коаксиального кабеля — для локальных сетей, для глобальных сетей, для кабельного телевидения и т. п.

Волоконно-оптический кабель состоит из тонких (5-60 микрон) волокон, по которым распространяются световые сигналы. Это наиболее качественный тип кабеля — он обеспечивает передачу данных со скоростью до 10 Гбит/с и выше и к тому же лучше других типов обеспечивает защиту данных от внешних помех.

Радиоканалы наземной и спутниковой связи образуются с помощью передатчика и приемника радиоволн. Это могут быть каналы в диапазоне коротких, средних и длинных волн (КВ, СВ и ДВ), называемые также диапазонами амплитудной модуляции. Обеспечивают дальнюю связь, но при невысокой скорости передачи данных.

УКВ, для которых характерна частотная модуляция (Frequency Modulation, FM), а также диапазонах сверхвысоких частот (СВЧ или microwaves). В диапазоне СВЧ (свыше 4 ГГц) сигналы уже не отражаются ионосферой Земли и для устойчивой связи требуется наличие прямой видимости между передатчиком и приемником. Поэтому такие частоты используют либо спутниковые каналы, либо радиорелейные каналы, где это условие выполняется.

Аппаратура линий связи

Аппаратура передачи данных (DCE) непосредственно связывает компьютеры или локальные сети пользователя с линией связи и является **пограничным** оборудованием. Традиционно аппаратуру передачи данных включают в состав линии связи. Примеры DCE - модемы, терминальные адаптеры сетей ISDN, оптические модемы, устройства подключения к цифровым каналам. DCE работает на

физическом уровне, отвечая за передачу и прием сигнала нужной формы и мощности в физическую среду.

Оконечное оборудование данных (DTE). Примеры DTE - компьютеры или маршрутизаторы локальных сетей. Эту аппаратуру не включают в состав линии связи.

Разделение оборудования на классы DCE и DTE в локальных сетях является достаточно условным. Например, адаптер локальной сети можно считать как принадлежностью компьютера, то есть DTE, так и составной частью канала связи, то есть DCE.

Промежуточная аппаратура обычно используется на линиях связи большой протяженности и решает две основные задачи:

- улучшение качества сигнала;
- создание постоянного составного канала связи между двумя абонентами сети.

В глобальных сетях необходимо обеспечить качественную передачу сигналов на расстояния в сотни и тысячи километров. Поэтому без усилителей сигналов, установленных через определенные расстояния, построить территориальную линию связи невозможно.

В глобальной сети необходима также и промежуточная аппаратура другого рода — мультиплексоры, демультиплексоры и коммутаторы. Эта аппаратура решает вторую указанную задачу, то есть создает между двумя абонентами сети составной канал из некоммутируемых отрезков физической среды — кабелей с усилителями.

Важно отметить, что приведенные на слайде мультиплексоры, демультиплексоры и коммутаторы образуют составной канал на долговременной основе, например на месяц или год, причем абонент не может влиять на процесс коммутации этого канала — эти устройства управляются по отдельным входам, абоненту недоступным (на рисунке не показаны). Наличие промежуточной коммутационной аппаратуры избавляет создателей глобальной сети от необходимости прокладывать отдельную кабельную линию для каждой пары соединяемых узлов сети. Вместо этого между мультиплексорами и коммутаторами используется высокоскоростная физическая среда, например волоконно-оптический или коаксиальный кабель, по которому передаются одновременно данные от большого числа сравнительно низкоскоростных абонентских линий. Высокоскоростной канал обычно называют **уплотненным каналом.**

Промежуточная аппаратура канала связи прозрачна для пользователя, он ее не замечает и не учитывает в своей работе. Для него важны только качество полученного канала.

В действительности же промежуточная аппаратура образует сложную сеть, которую называют **первичной сетью**, так как сама по себе она никаких высокоуровневых служб (например, файловой или передачи голоса) не поддерживает, а только служит основой для построения компьютерных, телефонных или иных сетей.

В зависимости от типа промежуточной аппаратуры все линии связи делятся на аналоговые и цифровые.

В аналоговых линиях промежуточная аппаратура предназначена для усиления аналоговых сигналов, то есть сигналов, которые имеют непрерывный диапазон значений. Такие линии связи традиционно применялись в телефонных сетях для связи АТС между собой. Для создания высокоскоростных каналов, которые мультиплексируют несколько низкоскоростных аналоговых абонентских каналов, при аналоговом подходе обычно используется техника частотного мультиплексирования (FDM).

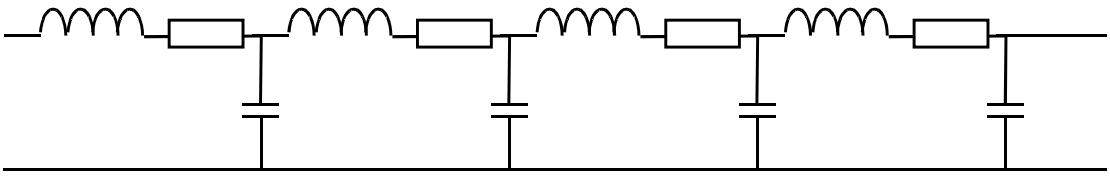
В цифровых линиях связи передаваемые сигналы имеют конечное число состояний. Как правило, элементарный сигнал, то есть сигнал, передаваемый за один такт работы передающей аппаратуры, имеет 2 или 3 состояния, которые передаются в линиях связи импульсами прямоугольной формы

Промежуточная аппаратура образования высокоскоростных цифровых каналов (мультиплексоры, демультиплексоры, коммутаторы) работает по принципу временного мультиплексирования каналов (TDM), когда каждому низкоскоростному каналу выделяется определенная доля времени (**тайм-слот**) высокоскоростного канала.

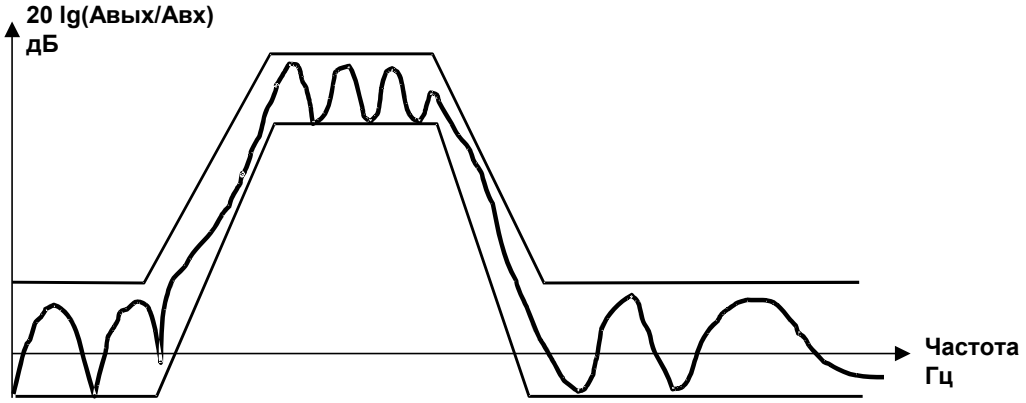
Аппаратура передачи дискретных компьютерных данных по аналоговым и цифровым линиям связи существенно отличается, так как в первом случае линия связи предназначена для передачи сигналов произвольной формы и не предъявляет никаких требований к способу представления единиц и

нулей аппаратурой передачи данных, а во втором — все параметры передаваемых линией импульсов стандартизованы. Другими словами, на цифровых линиях связи протокол физического уровня определен, а на аналоговых линиях — нет.

Характеристики линий связи



Представление линии связи как распределенной индуктивно-емкостной нагрузки



3

Характеристики линий связи.

Линия связи искажает передаваемые данные т.к. ее физические параметры отличаются от идеальных. Линия связи представляет собой некую распределенную комбинацию активного сопротивления, индуктивной и емкостной нагрузки.

Типы характеристик и способы их определения.

К основным характеристикам линий связи относятся:

- амплитудно-частотная характеристика;
- полоса пропускания;
- затухание;
- помехоустойчивость;
- перекрестные наводки на ближнем конце линии;
- пропускная способность;
- достоверность передачи данных;
- удельная стоимость.

В первую очередь разработчика вычислительной сети интересуют пропускная способность и достоверность передачи данных, поскольку эти характеристики прямо влияют на производительность и надежность создаваемой сети. Пропускная способность и достоверность — это характеристики как

линии связи, так и способа передачи данных. Поэтому если способ передачи (протокол) уже определен, то известны и эти характеристики. Например, пропускная способность цифровой линии всегда известна, так как на ней определен протокол физического уровня, который задает битовую скорость передачи данных — 64 Кбит/с, 2 Мбит/с и т. п.

Однако нельзя говорить о пропускной способности линии связи, до того как для нее определен протокол физического уровня.

Амплитудно-частотная характеристика, полоса пропускания и затухание

Амплитудно-частотная характеристика показывает, как затухает амплитуда синусоиды на выходе линии связи по сравнению с амплитудой на ее входе для всех возможных частот передаваемого сигнала. Вместо амплитуды в этой характеристике часто используют также такой параметр сигнала, как его мощность.

На практике вместо АЧХ применяются другие, упрощенные характеристики — полоса пропускания и затухание.

Полоса пропускания — это непрерывный диапазон частот, для которого отношение амплитуды выходного сигнала ко входному превышает некоторый заранее заданный предел, обычно 0,5. Ширина полосы пропускания в наибольшей степени влияет на максимально возможную скорость передачи информации по линии связи.

Затухание определяется как относительное уменьшение амплитуды или мощности сигнала при передаче по линии сигнала определенной частоты. Таким образом, затухание представляет собой одну точку из амплитудно-частотной характеристики линии. Часто при эксплуатации линии заранее известна основная частота передаваемого сигнала, то есть та частота, гармоника которой имеет наибольшую амплитуду и мощность. Поэтому достаточно знать затухание на этой частоте, чтобы приблизительно оценить искажения передаваемых по линии сигналов.

Затухание A обычно измеряется в децибелах и вычисляется по следующей формуле:

$$A = 10 \log (P_{вых}/P_{вх}),$$

Так как мощность выходного сигнала кабеля без промежуточных усилителей всегда меньше, чем мощность входного сигнала, затухание кабеля всегда является отрицательной величиной.

Например, кабель на витой паре категории 5 характеризуется затуханием не ниже -23,6 дБ для частоты 100 МГц при длине кабеля 100 м. Частота 100 МГц выбрана потому, что кабель этой категории предназначен для высокоскоростной передачи данных, сигналы которых имеют значимые гармоники с частотой примерно 100 МГц.

Кабель категории 3 предназначен для низкоскоростной передачи данных, поэтому для него определяется затухание на частоте 10 МГц (не ниже -11,5 дБ). Часто оперируют с абсолютными значениями затухания, без указания знака.

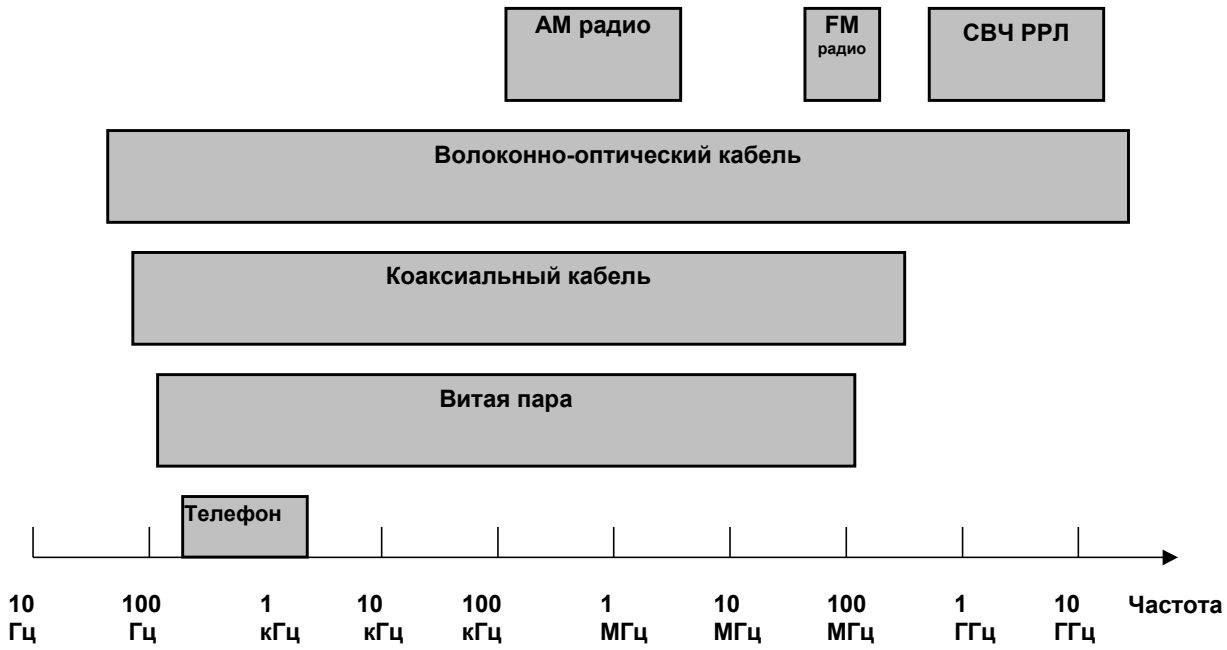
Абсолютный уровень мощности, например уровень мощности передатчика, также измеряется в децибелах. При этом в качестве базового значения мощности сигнала, относительно которого измеряется текущая мощность, принимается значение в 1 мВт. Таким образом, уровень мощности p вычисляется по следующей формуле:

$$p = 10 \log (P/1мВт) [дБм],$$

где P — мощность сигнала в милливаттах, а дБм (dBm) — это единица измерения уровня мощности (децибел на 1 мВт).

Таким образом, амплитудно-частотная характеристика, полоса пропускания и затухание являются универсальными характеристиками, и их знание позволяет сделать вывод о том, как через линию связи будут передаваться сигналы любой формы.

Характеристики линий связи



Полосы пропускания линий связи

4

Полоса пропускания зависит от типа линии и ее протяженности. На слайде показаны полосы пропускания линий связи различных типов, а также наиболее часто используемые в технике связи частотные диапазоны.

Пропускная способность линии

Пропускная способность линии характеризует максимально возможную скорость передачи данных по линии связи. Пропускная способность измеряется в битах в секунду — бит/с, а также в производных единицах, таких как килобит в секунду (Кбит/с), мегабит в секунду (Мбит/с), гигабит в секунду (Гбит/с) и т. д.

Выбор способа представления дискретной информации в виде сигналов, подаваемых на линию связи, называется физическим или линейным кодированием. От выбранного способа кодирования зависит спектр сигналов и, соответственно, пропускная способность линии. Таким образом, для одного способа кодирования линия может обладать одной пропускной способностью, а для другого — другой.

Теория информации говорит, что любое различимое и непредсказуемое изменение принимаемого сигнала несет в себе информацию. В соответствии с этим прием синусоиды, у которой амплитуда, фаза и частота остаются неизменными, информации не несет, так как изменение сигнала хотя и происходит, но является хорошо предсказуемым. Аналогично, не несут в себе информации импульсы на тактовой

шине компьютера, так как их изменения также постоянны во времени. А вот импульсы на шине данных предсказать заранее нельзя, поэтому они переносят информацию между отдельными блоками или устройствами.

Если сигнал изменяется так, что можно различить только два его состояния, то любое его изменение будет соответствовать наименьшей единице информации — биту. Если же сигнал может

иметь более двух различимых состояний, то любое его изменение будет нести несколько бит информации.

Количество изменений информационного параметра несущего периодического сигнала в секунду измеряется в бодах.

Пропускная способность линии в битах в секунду в общем случае не совпадает с числом бод. Она может быть как выше, так и ниже числа бод, и это соотношение зависит от способа кодирования.

Если сигнал имеет более двух различимых состояний, то пропускная способность в битах в секунду будет выше, чем число бод.

При использовании сигналов с двумя различимыми состояниями может наблюдаться обратная картина. Это часто происходит потому, что для надежного распознавания приемником пользовательской информации каждый бит в последовательности кодируется с помощью нескольких изменений информационного параметра несущего сигнала. Например, при кодировании единичного значения бита импульсом положительной полярности, а нулевого значения бита — импульсом отрицательной полярности физический сигнал дважды изменяет свое состояние при передаче каждого бита. При таком кодировании пропускная способность линии в два раза ниже, чем число бод, передаваемое по линии.

Например, если информационными параметрами являются фаза и амплитуда синусоиды, причем различаются 4 состояния фазы в 0, 90,180 и 270 градусов и два значения амплитуды сигнала, то информационный сигнал может иметь 8 различимых состояний. В этом случае модем, работающий со скоростью 2400 бод (с тактовой частотой 2400 Гц) передает информацию со скоростью 7200 бит/с, так как при одном изменении сигнала передается 3 бита информации.

На пропускную способность линии оказывает влияние не только физическое, но и логическое кодирование. Логическое кодирование выполняется до физического кодирования и подразумевает замену бит исходной информации новой последовательностью бит, несущей ту же информацию, но обладающей, кроме этого, дополнительными свойствами, например возможностью для приемной стороны обнаруживать ошибки в принятых данных.

Сопровождение каждого байта исходной информации одним битом четности — это пример очень часто применяемого способа логического кодирования при передаче данных с помощью модемов.

Другим примером логического кодирования может служить шифрация данных, обеспечивающая их конфиденциальность при передаче через общественные каналы связи.

При логическом кодировании чаще всего исходная последовательность бит заменяется более длинной последовательностью, поэтому пропускная способность канала по отношению к полезной информации при этом уменьшается.

Характеристики линий связи

$$C=F \log_2(1+Pc/Pш)$$

- формула Шеннона для максимальной пропускной способности

$$C=2F \log_2(M)$$

- соотношение Найквиста

F-ширина полосы пропускания линии связи
M –количество различимых состояний информационного параметра
Pc/Pш – соотношение мощностей сигнала и шума.

$$NEXT= 10 \lg(Pвых/Pнав)$$

– показатель перекрестных наводок на ближнем конце линии.

5

Связь между пропускной способностью линии и ее полосой пропускания

Чем выше частота несущего периодического сигнала, тем больше информации в единицу времени передается по линии и тем выше пропускная способность линии при фиксированном способе физического кодирования. Однако, с другой стороны, с увеличением частоты периодического несущего сигнала увеличивается и ширина спектра этого сигнала. Линия передает этот спектр синусоид с теми искажениями, которые определяются ее полосой пропускания. **Это не значит, что сигналы нельзя передавать.** Чем больше несоответствие между полосой пропускания линии и шириной спектра передаваемых информационных сигналов, тем больше сигналы искажаются и тем вероятнее ошибки в распознавании информации принимающей стороной, а значит, скорость передачи информации на самом деле оказывается меньше, чем можно было предположить.

Связь между полосой пропускания линии и ее максимально возможной пропускной способностью, вне зависимости от принятого способа физического кодирования, установил Клод Шеннон:

$$C = F \log_2 (1 + Pc/Pш),$$

где C — максимальная пропускная способность линии в битах в секунду,
F — ширина полосы пропускания линии в герцах,
Pc — мощность сигнала,
Pш — мощность шума.

Из этого соотношения видно, что хотя теоретического предела пропускной способности линии с фиксированной полосой пропускания не существует, на практике такой предел имеется. Действительно,

повысить пропускную способность линии можно за счет увеличения мощности передатчика или же уменьшения мощности шума (помех) на линии связи. Обе эти составляющие поддаются изменению с большим трудом. Повышение мощности передатчика ведет к значительному увеличению его габаритов и стоимости. Снижение уровня шума требует применения специальных кабелей с хорошими защитными экранами, что весьма дорого, а также снижения шума в передатчике и промежуточной аппаратуре, чего достичь весьма не просто.

К тому же влияние мощностей полезного сигнала и шума на пропускную способность ограничено логарифмической зависимостью, которая растет далеко не так быстро, как прямо-пропорциональная. Так, при достаточно типичном исходном отношении мощности сигнала к мощности шума в 100 раз повышение мощности передатчика в 2 раза даст только 15 % увеличения пропускной способности линии.

Близким по сути к формуле Шеннона является следующее соотношение, полученное Найквистом, которое также определяет максимально возможную пропускную способность линии связи, но без учета шума на линии:

$$C = 2F \log_2 M,$$

где M — количество различимых состояний информационного параметра.

Если сигнал имеет 2 различимых состояния, то пропускная способность равна удвоенному значению ширины полосы пропускания линии связи. Если же передатчик использует более чем 2 устойчивых состояния сигнала для кодирования данных, то пропускная способность линии повышается, так как за один такт работы передатчик передает несколько бит исходных данных, например 2 бита при наличии четырех различимых состояний сигнала.

Хотя формула Найквиста явно не учитывает наличие шума, косвенно его влияние отражается в выборе количества состояний информационного сигнала. Для повышения пропускной способности канала хотелось бы увеличить это количество до значительных величин, но на практике мы не можем этого сделать из-за шума на линии.

Например можно увеличить пропускную способность линии еще в два раза, использовав для кодирования данных не 4, а 16 уровней. Однако если амплитуда шума часто превышает разницу между соседними 16-ю уровнями, то приемник не сможет устойчиво распознавать передаваемые данные. Поэтому количество возможных состояний сигнала фактически ограничивается соотношением мощности сигнала и шума, а формула Найквиста определяет предельную скорость передачи данных в том случае, когда количество состояний уже выбрано с учетом возможностей устойчивого распознавания приемником.

Приведенные соотношения дают предельное значение пропускной способности линии, а степень приближения к этому пределу зависит от конкретных методов физического кодирования.

Помехоустойчивость и достоверность

Помехоустойчивость линии определяет ее способность уменьшать уровень помех, создаваемых во внешней среде, на внутренних проводниках. Помехоустойчивость линии зависит от типа используемой физической среды, а также от экранирующих и подавляющих помехи средств самой линии. Наименее помехоустойчивыми являются радиолинии, хорошей устойчивостью обладают кабельные линии и отличной — волоконно-оптические линии, малочувствительные ко внешнему электромагнитному излучению. Обычно для уменьшения помех, появляющихся из-за внешних электромагнитных полей, проводники экранируют и/или скручивают.

Перекрестные наводки на ближнем конце (Near End Cross Talk — NEXT) определяют помехоустойчивость кабеля к внутренним источникам помех, когда электромагнитное поле сигнала, передаваемого выходом передатчика по одной паре проводников, наводит на другую пару проводников сигнал помехи. Если ко второй паре будет подключен приемник, то он может принять наведенную внутреннюю помеху за полезный сигнал. Показатель NEXT, выраженный в децибелах, равен

$10 \log (P_{вых}/P_{нав})$, где $P_{вых}$ — мощность выходного сигнала, $P_{нав}$ — мощность наведенного сигнала.

Чем меньше значение NEXT, тем лучше кабель. Так, для витой пары категории 5 показатель NEXT должен быть меньше -27 дБ на частоте 100 МГц.

Показатель NEXT обычно используется применительно к кабелю, состоящему из нескольких витых пар, так как в этом случае взаимные наводки одной пары на другую могут достигать значительных величин. Для одинарного коаксиального кабеля (то есть состоящего из одной экранированной жилы) этот показатель не имеет смысла, а для двойного коаксиального кабеля он также не применяется вследствие высокой степени защищенности каждой жилы. Оптические волокна также не создают сколь-нибудь заметных помех друг для друга.

Достоверность передачи данных характеризует вероятность искажения для каждого передаваемого бита данных. Иногда этот же показатель называют интенсивностью битовых ошибок (Bit Error Rate, BER). Величина BER для каналов связи без дополнительных средств защиты от ошибок (например, самокорректирующихся кодов или протоколов с повторной передачей искаженных кадров) составляет, как правило, 10^{-4}-10^{-6}, в оптоволоконных линиях связи — 10^{-9}. Значение достоверности передачи данных, например, в 10^{-4} говорит о том, что в среднем из 10 000 бит искажается значение одного бита.

Искажения бит происходят как из-за наличия помех на линии, так и по причине искажений формы сигнала ограниченной полосой пропускания линии. Поэтому для повышения достоверности передаваемых данных нужно повышать степень помехозащищенности линии, снижать уровень перекрестных наводок в кабеле, а также использовать более широкополосные линии связи.

Стандарты кабелей

В стандартах кабелей оговаривается достаточно много характеристик, из которых наиболее важные перечислены ниже (первые две из них уже были достаточно детально рассмотрены).

- Затухание (Attenuation). Затухание измеряется в децибелах на метр для определенной частоты или диапазона частот сигнала,
- Перекрестные наводки на ближнем конце (Near End Cross Talk, NEXT). Измеряются в децибелах для определенной частоты сигнала.
- Импеданс (волновое сопротивление) — это полное (активное и реактивное) сопротивление в электрической цепи. Импеданс измеряется в Омах и является относительно постоянной величиной для кабельных систем (например, для коаксиальных кабелей, используемых в стандартах Ethernet, импеданс кабеля должен составлять 50 Ом). Для неэкранированной витой пары наиболее часто используемые значения импеданса — 100 и 120 Ом. В области высоких частот (100-200 МГц) импеданс зависит от частоты.
- Активное сопротивление — это сопротивление постоянному току в электрической цепи. В отличие от импеданса активное сопротивление не зависит от частоты и возрастает с увеличением длины кабеля.
- Емкость — это свойство металлических проводников накапливать энергию. Емкость является нежелательной величиной, поэтому следует стремиться к тому, чтобы она была как можно меньше (иногда применяют термин «паразитная емкость»). Высокое значение емкости в кабеле приводит к искажению сигнала и ограничивает полосу пропускания линии.
- Уровень внешнего электромагнитного излучения или электрический шум. Электрический шум — это нежелательное переменное напряжение в проводнике. Электрический шум бывает двух типов: фоновый и импульсный. Электрический шум можно также разделить на низко-, средне- и высокочастотный. Источниками фонового электрического шума в диапазоне до 150 кГц являются линии электропередачи, телефоны и лампы дневного света; в диапазоне от 150 кГц до 20 МГц — компьютеры, принтеры, ксероксы; в диапазоне от 20 МГц до 1 ГГц — телевизионные и радиопередатчики, микроволновые печи. Основными источниками импульсного электрического шума являются моторы, переключатели и сварочные агрегаты. Электрический шум измеряется в милливольтах.
- Диаметр или площадь сечения проводника. Для медных проводников достаточно употребительной является американская система AWG (American Wire Gauge), которая вводит

некоторые условные типы проводников, например 22 AWG, 24 AWG, 26 AWG. Чем больше номер типа проводника, тем меньше его диаметр.

Помимо универсальных характеристик, таких, например, как затухание, которые применимы для всех типов кабелей, существуют характеристики, которые применимы только к определенному типу кабеля. Например, параметр шаг скрутки проводов используется только для характеристики витой пары, а параметр NEXT применим только к многопарным кабелям на основе витой пары.

Основное внимание в современных стандартах уделяется кабелям на основе витой пары и волоконно-оптическим кабелям.

Волоконно-оптические кабели

Волоконно-оптические кабели состоят из центрального проводника света (сердцевины) — стеклянного волокна, окруженного другим слоем стекла — оболочкой, обладающей меньшим показателем преломления, чем сердцевина. Распространяясь по сердцевине, лучи света не выходят за ее пределы, отражаясь от покрывающего слоя оболочки. В зависимости от распределения показателя преломления и от величины диаметра сердечника различают:

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

Одномодовый кабель (Single Mode Fiber, SMF):

- центральный проводник очень малого диаметра, соизмеримого с длиной волны света - от 5 до 10 мкм. При этом практически все лучи света распространяются вдоль оптической оси световода, не отражаясь от внешнего проводника
- полоса пропускания очень широкая - до сотен гигагерц на километр
- изготовление тонких качественных волокон для одномодового кабеля представляет сложный технологический процесс, что делает одномодовый кабель достаточно дорогим
- в волокно такого маленького диаметра достаточно сложно направить пучок света, не потеряв при этом значительную часть его энергии.

Многомодовый кабель (Multi Mode Fiber, MMF):

- более широкие внутренние сердечники, которые легче изготовить технологически
- в многомодовых кабелях во внутреннем проводнике одновременно существует несколько световых лучей, отражающихся от внешнего проводника под разными углами. Угол отражения луча называется модой луча
- имеют более узкую полосу пропускания — от 500 до 800 МГц/км. Сужение полосы происходит из-за потерь световой энергии при отражениях, а также из-за интерференции лучей разных мод.

В качестве источников излучения света в волоконно-оптических кабелях применяются:

- светодиоды;
- полупроводниковые лазеры.

Для одномодовых кабелей применяются только полупроводниковые лазеры, так как при таком малом диаметре оптического волокна световой поток, создаваемый светодиодом, невозможно без больших потерь направить в волокно. Для многомодовых кабелей используются более дешевые светодиодные излучатели.

Методу передачи дискретных данных на -физическом уровне

1. Модуляция.
2. Цифровое кодирование.

При амплитудной модуляции для логической единицы выбирается один уровень амплитуды синусоиды несущей частоты, а для логического нуля — другой. Этот способ редко используется в чистом виде на практике из-за низкой помехоустойчивости, но часто применяется в сочетании с другим видом модуляции — фазовой модуляцией.

При частотной модуляции значения 0 и 1 исходных данных передаются синусоидами с различной частотой.

При фазовой модуляции значениям данных 0 и 1 соответствуют сигналы одинаковой частоты, но с различной фазой, например 0 и 180 градусов или 0,90,180 и 270 градусов.

В скоростных модемах часто используются комбинированные методы модуляции, как правило, амплитудная в сочетании с фазовой.

Спектр модулированного сигнала

Рассмотрим сначала спектр сигнала при потенциальном кодировании. Пусть логическая единица кодируется положительным потенциалом, а логический ноль — отрицательным потенциалом такой же величины. Для упрощения вычислений предположим, что передается информация, состоящая из бесконечной последовательности чередующихся единиц и нулей, в данном случае величины бод и бит в секунду совпадают.

Если дискретные данные передаются с битовой скоростью N бит/с, то спектр состоит из постоянной составляющей нулевой частоты и бесконечного ряда гармоник с частотами f0,3fo, 5f0,7f0,..., где f0 = N/2. Амплитуды этих гармоник убывают достаточно медленно — с коэффициентами 1/3, 1/5,1/7,... от амплитуды гармоники f0. В результате спектр потенциального кода требует для качественной передачи широкую полосу пропускания. Кроме того, нужно учесть, что реально спектр сигнала постоянно меняется в зависимости от того, какие данные передаются по линии связи. Например, передача длинной последовательности нулей или единиц сдвигает спектр в сторону низких частот, а в крайнем случае, когда передаваемые данные состоят только из единиц (или только из нулей), спектр состоит из гармоники нулевой частоты. При передаче чередующихся единиц и нулей постоянная составляющая отсутствует. Поэтому спектр результирующего сигнала потенциального кода при передаче произвольных данных занимает полосу от некоторой величины, близкой к 0 Гц, до примерно 7fo (гармониками d частотами выше 7fo можно пренебречь из-за их малого вклада в результирующий сигнал).

Для канала тональной частоты (телефон) верхняя граница при потенциальном кодировании достигается для скорости передачи данных в 971 бит/с, а нижняя неприемлема для любых скоростей, так как полоса пропускания канала начинается с 300 Гц. В результате потенциальные коды на каналах тональной частоты никогда не используются.

Цифровое кодирование.

Применяют потенциальные и импульсные коды. Потенциальные для представления сигнала используют уровень, импульсные – импульс либо перепад.

Требования к методам цифрового кодирования:

- Битовая синхронизация приемника и передатчика
- Распознавание ошибочных символов
- Отсутствие постоянной составляющей для обеспечения гальванической развязки
- Экономичное использование частотного спектра

Синхронизация.

- отдельной тактирующей линии связи (на небольших расстояниях)
- самосинхронизирующиеся коды, сигналы которых несут для передатчика указания о том, в какой момент времени нужно осуществлять распознавание очередного бита (или нескольких бит, если код ориентирован более чем на два состояния сигнала). Любой резкий перепад

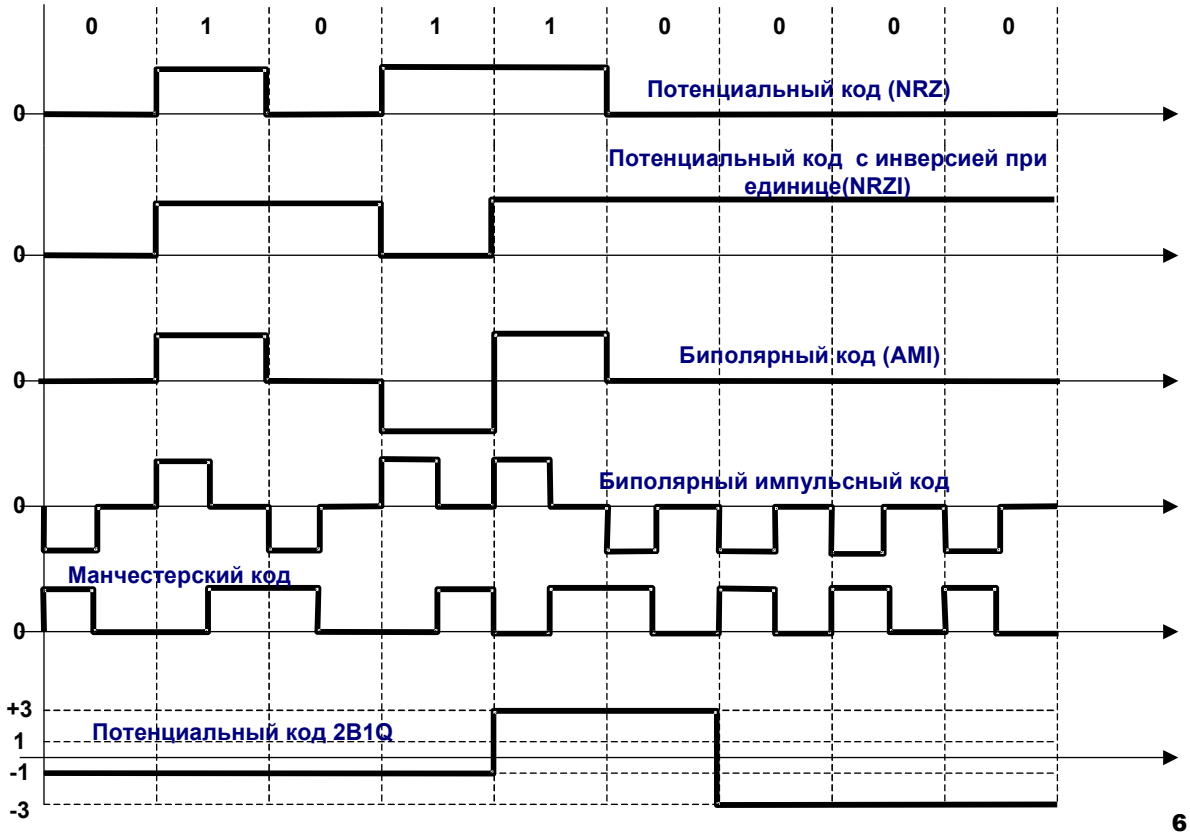
сигнала — так называемый фронт — может служить хорошим указанием для синхронизации приемника с передатчиком.

При использовании синусоид в качестве несущего сигнала результирующий код обладает свойством самосинхронизации, так как изменение амплитуды несущей частоты дает возможность приемнику определить момент появления входного кода.

Распознавание и коррекцию искаженных данных сложно осуществить средствами физического уровня, поэтому чаще всего эту работу берут на себя протоколы, лежащие выше: канальный, сетевой, транспортный или прикладной. С другой стороны, распознавание ошибок на физическом уровне экономит время, так как приемник не ждет полного помещения кадра в буфер, а отбраковывает его сразу при распознавании ошибочных бит внутри кадра.

Требования, предъявляемые к методам кодирования, являются взаимно противоречивыми, поэтому каждый из рассматриваемых ниже популярных методов цифрового кодирования обладает своими преимуществами и своими недостатками по сравнению с другими.

Дискретное кодирование данных



Потенциальный код без возвращения к нулю

Плюсы

- прост в реализации,
- обладает хорошей распознаваемостью ошибок (из-за двух резко отличающихся потенциалов),

Минусы

- не обладает свойством самосинхронизации. При передаче длинной последовательности единиц или нулей сигнал на линии не изменяется, поэтому приемник лишен возможности определять по входному сигналу моменты времени, когда нужно в очередной раз считывать данные. Даже при наличии высокоточного тактового генератора приемник может ошибиться с моментом

съема данных, так как частоты двух генераторов никогда не бывают полностью идентичными. Поэтому при высоких скоростях обмена данными и длинных последовательностях единиц или нулей небольшое рассогласование тактовых частот может привести к ошибке в целый такт и, соответственно, считыванию некорректного значения бита.

- наличие низкочастотной составляющей, которая приближается к нулю при передаче длинных последовательностей единиц или нулей. Из-за этого многие каналы связи, не обеспечивающие прямого гальванического соединения между приемником и источником, этот вид кодирования не поддерживают. В результате в чистом виде код NRZ в сетях не используется. Тем не менее используются его различные модификации, в которых устраняют как плохую самосинхронизацию кода NRZ, так и наличие постоянной составляющей. Привлекательность кода NRZ, из-за которой имеет смысл заняться его улучшением, состоит в достаточно низкой частоте основной гармоники fo, которая равна N/2 Гц, как это было показано в предыдущем разделе. У других методов кодирования, например манчестерского, основная гармоника имеет более высокую частоту.

Метод биполярного кодирования с альтернативной инверсией

Для кодирования логического нуля используется нулевой потенциал, а логическая единица кодируется либо положительным потенциалом, либо отрицательным, при этом потенциал каждой новой единицы противоположен потенциалу предыдущей.

- частично ликвидирует проблемы постоянной составляющей и отсутствия самосинхронизации, присущие коду NRZ. Это происходит при передаче длинных последовательностей единиц. В этих случаях сигнал на линии представляет собой последовательность разнополярных импульсов с тем же спектром, что и у кода NRZ, передающего чередующиеся нули и единицы, то есть без постоянной составляющей и с, основной гармоникой N/2 Гц (где N — битовая скорость передачи данных). Длинные же последовательности нулей также опасны для кода AMI, как и для кода NRZ — сигнал вырождается в постоянный потенциал нулевой амплитуды. Поэтому код AMI требует дальнейшего улучшения, хотя задача упрощается — осталось справиться только с последовательностями нулей.
- для различных комбинаций бит на линии использование кода AMI приводит к более узкому спектру сигнала, чем для кода NRZ, а значит, и к более высокой пропускной способности линии. Например, при передаче чередующихся единиц и нулей основная гармоника fo имеет частоту N/4 Гц. Код AMI предоставляет также некоторые возможности по распознаванию ошибочных сигналов. Так, нарушение строгого чередования полярности сигналов говорит о ложном импульсе или исчезновении с линии корректного импульса.
- в коде AMI используются не два, а три уровня сигнала на линии. Дополнительный уровень требует увеличение мощности передатчика примерно на 3 дБ для обеспечения той же достоверности приема бит на линии, что является общим недостатком кодов с несколькими состояниями сигнала по сравнению с кодами, которые различают только два состояния.

Потенциальный код с инверсией при единице

Существует код, похожий на AMI, но только с двумя уровнями сигнала. При передаче нуля он передает потенциал, который был установлен в предыдущем такте (то есть не меняет его), а при передаче единицы потенциал инвертируется на противоположный. Этот код называется потенциальным кодом с инверсией при единице

Этот код удобен в тех случаях, когда использование третьего уровня сигнала весьма нежелательно, например в оптических кабелях, где устойчиво распознаются два состояния сигнала - свет и темнота. Для улучшения потенциальных кодов, подобных AMI и NRZI, используются два метода. Первый метод основан на добавлении в исходный код избыточных бит, содержащих логические единицы. Очевидно, что в этом случае длинные последовательности нулей прерываются и код

становится самосинхронизирующимся для любых передаваемых данных. Исчезает также постоянная составляющая, а значит, еще более сужается спектр сигнала. Но этот метод снижает полезную пропускную способность линии, так как избыточные единицы пользовательской информации не несут.

Другой метод основан на предварительном «перемешивании» исходной информации таким образом, чтобы вероятность появления единиц и нулей на линии становилась близкой. Устройства, или блоки, выполняющие такую операцию, называются скрэмблерами (scramble — свалка, беспорядочная сборка). При скремблировании используется известный алгоритм, поэтому приемник, получив двоичные данные, передает их на дескрэмблер, который восстанавливает исходную последовательность бит.

Избыточные биты при этом по линии не передаются. Оба метода относятся к логическому, а не физическому кодированию, так как форму сигналов на линии они не определяют.

Биполярный импульсный код

- обладает отличными самосинхронизирующими свойствами
- но постоянная составляющая может присутствовать, например, при передаче длинной последовательности единиц или нулей
- кроме того, спектр у него шире, чем у потенциальных кодов. Так, при передаче всех нулей или единиц частота основной гармоники кода будет равна N Гц, что в два раза выше основной гармоники кода NRZ и в четыре раза выше основной гармоники кода AMI при передаче чередующихся единиц и нулей. Из-за слишком широкого спектра биполярный импульсный код используется редко.

Манчестерский код

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

- для кодирования единиц и нулей используется перепад потенциала, то есть фронт импульса
- каждый такт делится на две части, а информация кодируется перепадами потенциала, происходящими в середине каждого такта. В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько единиц или нулей подряд
- так как сигнал изменяется по крайней мере один раз за такт передачи одного бита данных, то манчестерский код обладает хорошими самосинхронизирующими свойствами
- полоса пропускания манчестерского кода уже, чем у биполярного импульсного
- у него также нет постоянной составляющей, а основная гармоника в худшем случае (при передаче последовательности единиц или нулей) имеет частоту N Гц, а в лучшем (при передаче чередующихся единиц и нулей) она равна N/2 Гц, как и у кодов AMI или NRZ.
- в среднем ширина полосы манчестерского кода в полтора раза уже, чем у биполярного импульсного кода, а основная гармоника колеблется вблизи значения 3N/4.
- Манчестерский код имеет еще одно преимущество перед биполярным импульсным кодом. В последнем для передачи данных используются три уровня сигнала, а в манчестерском — два.

Потенциальный код 2B1Q

На рис. 2.16, д показан потенциальный код с четырьмя уровнями сигнала для кодирования данных. Это код 2B1Q название которого отражает его суть — каждые два бита (2B) передаются за один такт сигналом, имеющим четыре состояния (1Q). Паре бит 00 соответствует потенциал -2,5 В, паре бит 01 соответствует потенциал -0,833 В, паре 11 — потенциал +0,833 В, а паре 10 — потенциал +2,5 В.

При этом способе кодирования требуются дополнительные меры по борьбе с длинными последовательностями одинаковых пар бит, так как при этом сигнал превращается в постоянную

составляющую. При случайном чередовании бит спектр сигнала в два раза уже, чем у кода NRZ, так как при той же битовой скорости длительность такта увеличивается в два раза.

Таким образом, с помощью кода 2B1Q можно по одной и той же линии передавать данные в два раза быстрее, чем с помощью кода AMI или NRZI.

Однако для его реализации мощность передатчика должна быть выше, чтобы четыре уровня четко различались приемником на фоне помех.

Логическое кодирование

Логическое кодирование используется для улучшения потенциальных кодов типа AMI, NRZI или 2Q1B.

Логическое кодирование должно заменять длинные последовательности бит, приводящие к постоянному потенциалу, вкраплениями единиц.

Для логического кодирования характерны два метода — избыточные коды и скрэмблирование.

Избыточные коды

Основаны на разбиении исходной последовательности бит на порции, которые часто называют символами. Затем каждый исходный символ заменяется на новый, который имеет большее количество бит, чем исходный.

Логическое кодирование

Исходный код	Результирующий код	Исходный код	Результирующий код
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

7

Например, логический код 4B/5B, используемый в технологиях FDDI и Fast Ethernet, меняет исходные символы длиной в 4 бита на символы длиной в 5 бит. Так как результирующие символы содержат избыточные биты, то общее количество битовых комбинаций в них больше, чем в исходных. Так, в коде 4B/5B результирующие символы могут содержать 32 битовых комбинации, в то время как исходные символы — только 16. Поэтому в результирующем коде можно отобрать 16 таких

комбинаций, которые не содержат большого количества нулей, а остальные считать запрещенными кодами (code violation).

Кроме устранения постоянной составляющей и придания коду свойства самосинхронизации, избыточные коды позволяют приемнику распознавать искаженные биты. Если приемник принимает запрещенный код, значит, на линии произошло искажение сигнала.

Код 4B/5B затем передается по линии с помощью физического кодирования по одному из методов потенциального кодирования, чувствительному только к длинным последовательностям нулей.

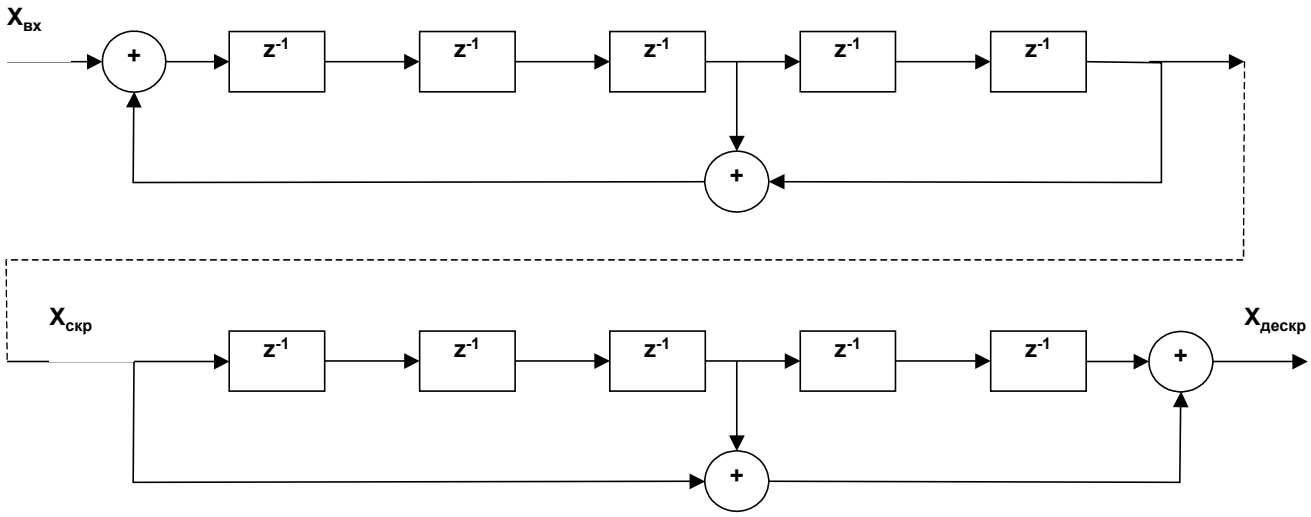
Символы кода 4B/5B длиной 5 бит гарантируют, что при любом их сочетании на линии не могут встретиться более трех нулей подряд.

Буква B в названии кода означает, что элементарный сигнал имеет 2 состояния — от английского binary — двоичный. Имеются также коды и с тремя состояниями сигнала, например, в коде 8B/6T для кодирования 8 бит исходной информации используется код из 6 сигналов, каждый из которых имеет три состояния. Избыточность кода 8B/6T выше, чем кода 4B/5B, так как на 256 исходных кодов приходится 3^6=729 результирующих символов.

Использование таблицы перекодировки является очень простой операцией, поэтому этот подход не усложняет сетевые адаптеры и интерфейсные блоки коммутаторов и маршрутизаторов.

Для обеспечения заданной пропускной способности линии передатчик, использующий избыточный код, должен работать с повышенной тактовой частотой. Так, для передачи кодов 4B/5B со скоростью 100 Мб/с передатчик должен работать с тактовой частотой 125 МГц. При этом спектр сигнала на линии расширяется по сравнению со случаем, когда по линии передается чистый, не избыточный код. Тем не менее спектр избыточного потенциального кода оказывается уже спектра манчестерского кода, что оправдывает дополнительный этап логического кодирования, а также работу приемника и передатчика на повышенной тактовой частоте.

Дискретное кодирование данных



Скремблер и дескремблер информационной последовательности

Скрэмблирование

Перемешивание данных скрэмблером перед передачей их в линию с помощью потенциального кода является другим способом логического кодирования.

Методы скрэмблирования заключаются в побитном вычислении результирующего кода на основании бит исходного кода и полученных в предыдущих тактах бит результирующего кода. Например, скрэмблер может реализовывать следующее соотношение:

$B_i = A_i + B_{i-3} + B_{i-5}$,

где Bi — двоичная цифра результирующего кода, полученная на i-м такте работы скрэмблера, Ai - двоичная цифра исходного кода, поступающая на i-м такте на вход скрэмблера, Bi-з и Bi-5 — двоичные цифры результирующего кода, полученные на предыдущих тактах работы скрэмблера, соответственно на 3 и на 5 тактов ране» текущего такта, + операция исключающего ИЛИ (сложение по модулю 2).

Например, для исходной последовательности 110110000001 скрэмблер даст с дующий результирующий код:

B1 = A2 = 1 (первые три цифры результирующего кода будут совпадать с исходным, так как еще нет нужных предыдущих цифр)

B2 = A2 = 1

B3 = A3 = 0

B4= A4 + B1 = 1 + 1 = 0

B5 = A5 + B2 = 1 + 1 = 0
B6 = A6 + B3 + B1 = 0 + 0 + 1 = 1
B7 = A7 + B4 + B2 = 0 + 0 + 1 = 1
B8 = A8 + B5 + B3 = 0 + 0 + 0 = 0
B9 = A9 + B6 + B4 = 0 + 1 + 0 = 1
B10 = A10 + B7 + B5 = 0 + 1 + 0 = 1
B11 = A11 + B8 + B6 = 0 + 0 + 1 = 1
Bl2 = A12 + B9 + B7 = 1 + 1 + 1 = 1

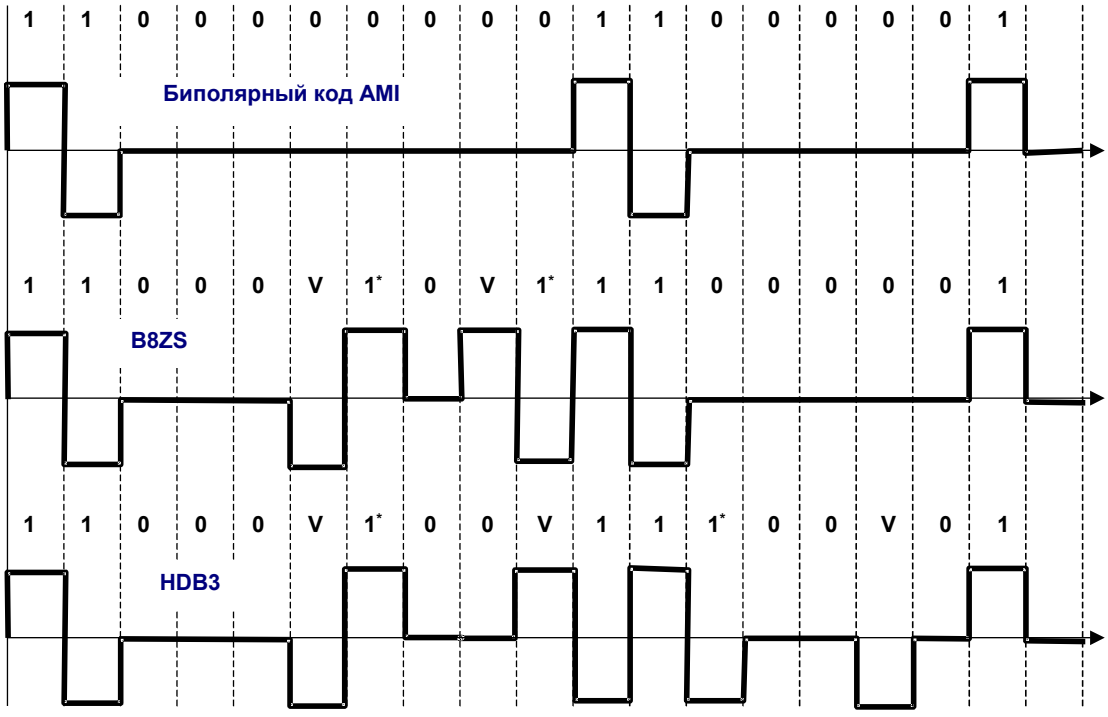
Таким образом, на выходе скрэмблера появится последовательность 110001101111, в которой нет последовательности из шести нулей, присутствовавшей в исходном коде.

После получения результирующей последовательности приемник передает ее дескрэмблеру, который восстанавливает исходную последовательность на основании обратного соотношения:

Ci = Bi + Bi-3 + Bi-5 = (Ai + Bi-3 + Bi-5) + Bi-3 + Bi-5 = A

Различные алгоритмы скрэмблирования отличаются количеством слагаемых, дающих цифру результирующего кода, и сдвигом между слагаемыми. Так, в сетях ISDN при передаче данных от сети к абоненту используется преобразование со сдвигами в 5 и 23 позиции, а при передаче данных от абонента в сеть — со сдвигами 18 и 23 позиции.

Дискретное кодирование данных



Линейные коды высокой плотности B8ZS и HDB3

9

Существуют и более простые методы борьбы с последовательностями единиц, также относимые к классу скрэмблирования.

Для улучшения кода Bipolar AMI используются два метода, основанные на искусственном искажении последовательности нулей запрещенными символами.

B8ZS (Bipolar with 8 Zeros Substitution) и метод HDB3 (High-Density Bipolar 3-Zeros) для корректировки кода AMI.

Исходный код состоит из двух длинных последовательностей нулей: в первом случае — из 8, а во втором — из 5.

Код B8ZS исправляет только последовательности, состоящие из 8 нулей. Для этого он после первых трех нулей вместо оставшихся пяти нулей вставляет пять цифр: V - l* - 0 –V - l*. V здесь обозначает сигнал единицы, запрещенной для данного такта полярности, то есть сигнал, не изменяющий полярность предыдущей единицы, 1* — сигнал единицы корректной полярности, а знак звездочки отмечает тот факт, что в исходном коде в этом такте была не единица, а ноль. В результате на 8 тактах приемник наблюдает 2 искажения — очень маловероятно, что это случилось из-за шума на линии или других сбоев передачи. Поэтому приемник считает такие нарушения кодировкой 8 последовательных нулей и после приема заменяет их на исходные 8 нулей.

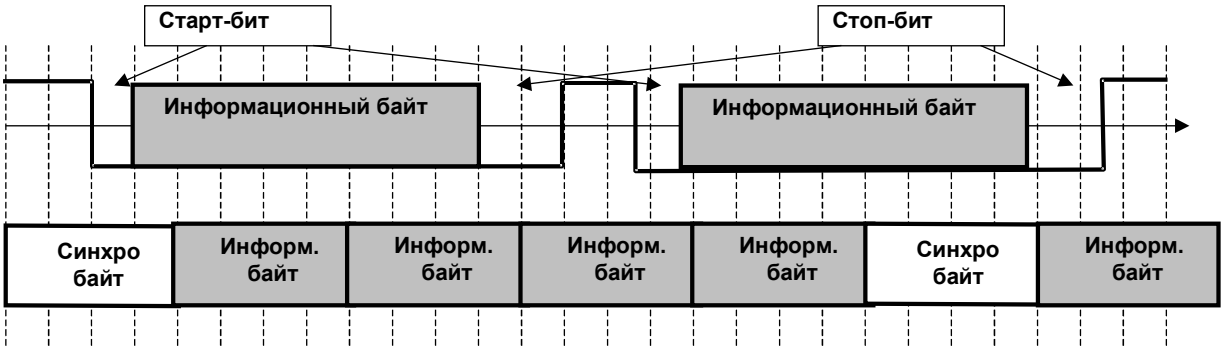
Код B8ZS построен так, что его постоянная составляющая равна нулю при любых последовательностях двоичных цифр.

Код HDB3 исправляет любые четыре подряд идущих нуля в исходной последовательности.

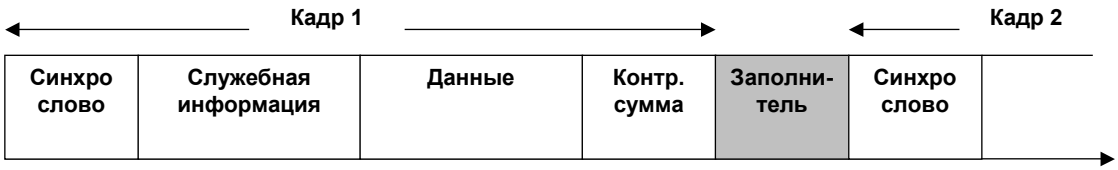
Правила формирования кода HDB3 более сложные, чем кода B8ZS. Каждые четыре нуля заменяются четырьмя сигналами, в которых имеется один сигнал V. Для подавления постоянной составляющей полярность сигнала V чередуется при последовательных заменах. Кроме того, для замены используются два образца четырехтактовых кодов. Если перед заменой исходный код содержал нечетное число единиц, то используется последовательность 000V, а если число единиц было четным — последовательность l*00V.

Улучшенные потенциальные коды обладают достаточно узкой полосой пропускания для любых последовательностей единиц и нулей, которые встречаются в передаваемых данных. Коды, полученные из потенциального путем логического кодирования, обладают более узким спектром, чем манчестерский, даже при повышенной тактовой частоте. Этим объясняется применение потенциальных избыточных и скрэмблированных кодов в современных технологиях, подобных FDDI, Fast Ethernet, Gigabit Ethernet, ISDN и т. п. вместо манчестерского и биполярного импульсного кодирования.

Дискретное кодирование данных



Асинхронная и синхронная передача на уровне байт



Кадровая структура синхронных протоколов

10

Асинхронная и синхронная передачи

В асинхронном режиме каждый байт данных сопровождается специальными сигналами «старт» и «стоп». Назначение этих сигналов состоит в том, чтобы, во-первых, известить приемник о приходе данных и, во-вторых, чтобы дать приемнику достаточно времени для выполнения некоторых функций, связанных с синхронизацией, до поступления следующего байта. Сигнал «старт» имеет продолжительность в один тактовый интервал, а сигнал «стоп» может длиться один, полтора или два такта, поэтому говорят, что используется один, полтора или два бита в качестве стопового сигнала, хотя пользовательские биты эти сигналы не представляют.

Асинхронным описанный режим называется потому, что каждый байт может быть несколько смещен во времени относительно побитовых тактов предыдущего байта.

При синхронном режиме передачи старт-стопные биты между каждой парой байт отсутствуют. Пользовательские данные собираются в кадр, который предваряется байтами синхронизации. Байт синхронизации — это байт, содержащий заранее известный код, который оповещает приемник о приходе кадра данных. При его получении приемник должен войти в байтовый синхронизм с передатчиком, то есть правильно понимать начало очередного байта кадра. Иногда применяется несколько синхробайт для обеспечения более надежной синхронизации приемника и передатчика.