

Вопросы к экзамену

1. Базовая эталонная модель взаимодействия открытых систем (модель OSI)

	Уровень (layer)	Тип данных (PDU ^[1])	Функции	Примеры
Host layers	7. Прикладной (application)		Доступ к сетевым службам	HTTP, FTP, SMTP
	6. Представительский (представления) (presentation)		Представление и шифрование данных	ASCII, EBCDIC, JPEG
	5. Сеансовый (session)		Управление сеансом связи	RPC, PAP, L2TP
	4. Транспортный (transport)	Сегменты (segment)/ Дейтаграммы (datagram)	Прямая связь между конечными пунктами и надёжность	TCP, UDP, SCTP
Media ^[2] layers	3. Сетевой (network)	Пакеты (packet)	Определение маршрута и логическая адресация	IPv4, IPv6, IPsec, AppleTalk
	2. Канальный (data link)	Биты (bit)/ Кадры (frame)	Физическая адресация	PPP, IEEE 802.2, Ethernet, DSL, ARP
	1. Физический (physical)	Биты (bit)	Работа со средой передачи, сигналами и двоичными данными	USB, витая пара, коаксиальный кабель, оптический кабель

Этот уровень

1.4. ЭТАЛОННАЯ МОДЕЛЬ OSI

Модель **OSI** основана на предложении **ISO** и называется **ISO OS Reference Model** – эталонная модель взаимодействия открытых систем **ISO**, поскольку она связывает открытые системы, т. е. системы, открытые для связи с другими системами.

Модель **OSI** имеет семь уровней (рис. 1.3), описывает только системные средства взаимодействия и не включает средства взаимодействия приложений конечных пользователей.

Модель **OSI** не является сетевой архитектурой, поскольку не описывает службы и протоколы.

Уровни модели **OSI**:

1. **Физический (Physical)** уровень занимается передачей битов по физическим каналам связи, определяет характеристики физических сред передачи, параметры сигналов.

2. **Канальный (Data Link)** уровень обеспечивает доставку данных между узлами только типовой топологии. Одна из задач – проверка доступности среды передачи и определение правил совместного использования физического уровня. Другой задачей является реализация механизма обнаружения и коррекции ошибок. Для этого группирует биты в блоки называемые **кадрами (frames)**. Канальный уровень определяет форму кадров, способы их выделения из потока битов. Для обнаружения ошибок используется добавляемая в кадр **контрольная сумма**. В протоколе канального уровня закладывается определенная структура связей между компьютерами и способами их адресации – определяется **локальный (физический) адрес узла**.

3. **Сетевой (Network)** уровень служит для образования единой транспортной системы, объединяющей несколько сетей, причем сети могут использовать различные принципы передачи.

Под сетью понимается совокупность компьютеров, соединенных в соответствии с одной из базовых топологий и использующих определенный для нее один из протоколов канального уровня.

Сообщения сетевого уровня принято называть **пакетами**. При организации передачи пакетов вводится понятие «номер сети». В этом случае сетевой адрес получателя состоит из **номера сети и адреса узла** в этой сети.

На сетевом уровне важнейшей задачей является продвижение пакетов через сеть – **маршрутизация пакетов**. Для ее решения выполняется сбор информации о топологии межсетевых соединений для определения **оптимальных маршрутов**.

Сетевой уровень отвечает за **преобразование адресов**, используемых на данном уровне, в локальные адреса узлов.

4. Транспортный (*Transport*) уровень обеспечивает доставку пакетов приложения на одном узле к приложению на другом (*транспортное соединение*). Обычно для каждого транспортного соединения создается отдельное сетевое соединение. Для повышения пропускной способности транспортный уровень может объединять несколько транспортных соединений в одном сетевом (*мультиплексирование/демультиплексирование соединений*). Подобное объединение/разъединение является «прозрачным» для смежных уровней.

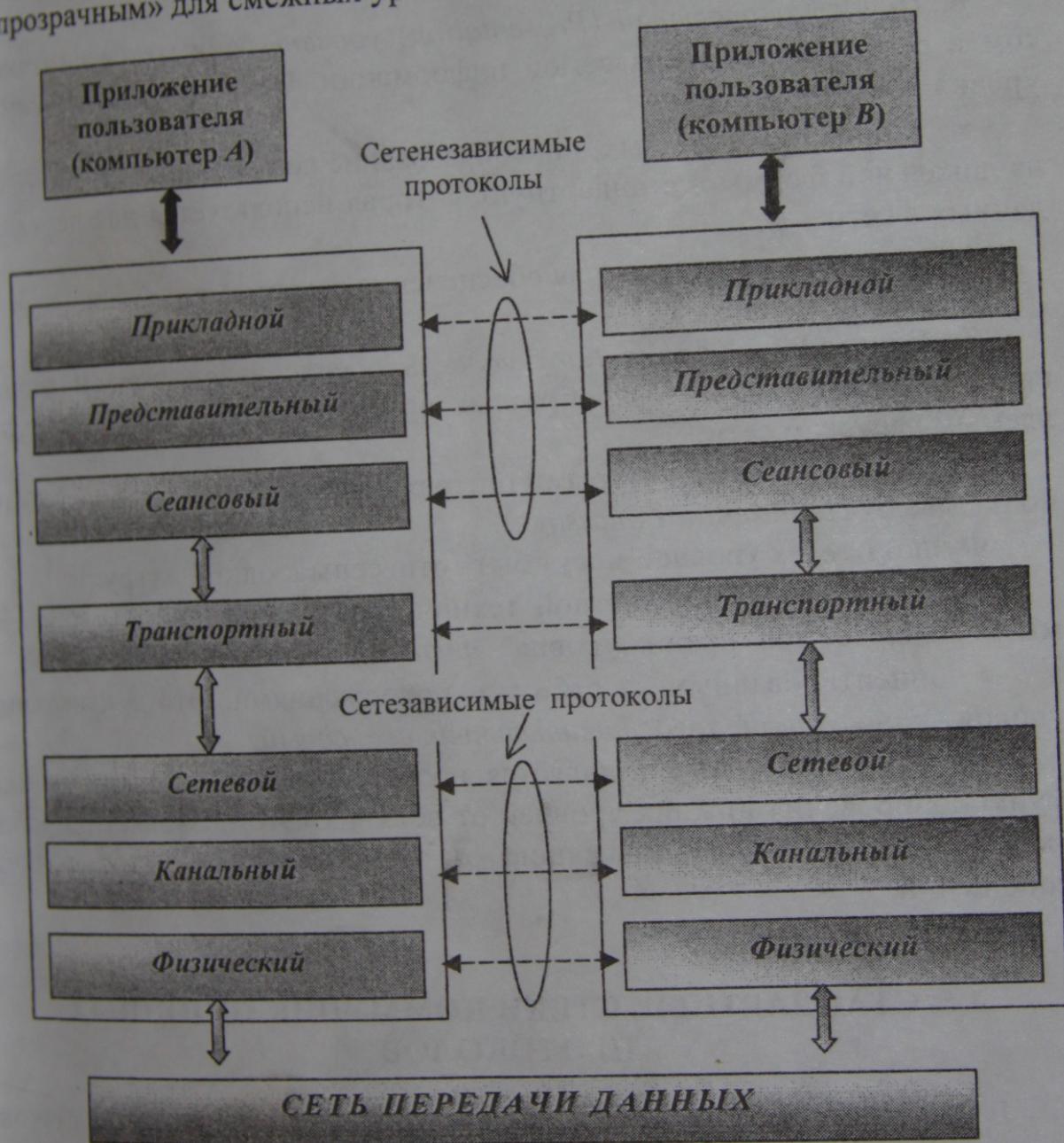


Рис. 1.3. Эталонная модель OSI

Транспортный уровень также определяет тип службы, предоставляемой пользователям сети, которая обеспечивает требуемую степень надежности передачи данных (*класс сервиса*). Важной задачей транспортного уровня является *управление потоком*.

5. *Сеансовый (Session)* уровень позволяет пользователям различным узлов устанавливать сеансы связи друг с другом: управляет диалогом, предоставляет средства синхронизации. В частности, позволяет вставлять контрольные точки в длинные передачи, чтобы в случае отката можно было вернуться к последней контрольной точке, а не начинать сеанс с начала.

6. *Представительный (Presentation)* уровень занимается синтаксисом и семантикой передаваемой информации. Основные задачи этого уровня представления:

- отображение данных – преобразование сообщений пользователей из локальной формы в стандартную, которая используется для передачи данных в сети;
- шифрование данных для обеспечения секретности обмена;
- сжатие данных.

7. *Прикладной (Application)* уровень обеспечивает сетевой сервис для пользователей – набор разнообразных протоколов для доступа к различным ресурсам сети.

Единицу данных, с которыми оперирует прикладной уровень, обычно называют *сообщением (message)*.

Функции всех уровней могут быть отнесены к одной из групп:

- зависящих от конкретной технической реализации. Сетезависимыми являются три нижних уровня – *физический, канальный, сетевой*;
- ориентированных на работу с приложениями. Это три верхних уровня – *прикладной, представительный, сеансовый*.

Транспортный уровень является промежуточным, скрывает детали функционирования нижних уровней от верхних. Это позволяет разрабатывать приложения, не зависящие от технических средств передачи сообщений.

2. Виды каналов связи: оптический, медный, беспроводной.

2.4. КАБЕЛЬНЫЕ СИСТЕМЫ

Кабельные системы являются самым распространенным типом каналов связи в современных КС.

2.4.1. Медные кабельные системы

В качестве линии связи используются, по крайней мере, два медных проводника, по которым информация передается с помощью электрических сигналов. Сигналы по проводникам передаются в *потенциальном* либо *токовом* представлении.

При потенциальном представлении информационным параметром является *уровень напряжения сигнала* между передатчиком и приемником, при этом передача может быть *асимметричной* или *дифференциальной* (симметричной).

При асимметричной передаче один из проводников назначается общим – его потенциал относительно земли остается постоянным. Информационным параметром является потенциал на другом (сигнальном) проводнике относительно общего проводника.

При дифференциальной передаче оба проводника равноправны, информационным параметром является разность потенциалов между ними.

При токовом представлении сигнала информационным параметром является наличие или отсутствие тока в цепи либо направление тока.

Для передачи сигналов используются две основные разновидности кабеля: *коаксиальный* и *витые пары* проводников.

Медные кабельные системы характеризуются набором параметров, распределенных по всей его длине: емкостью, сопротивлением изоляции между проводниками, индуктивностью и активным сопротивлением.

Важным комплексным параметром кабеля является *волновое сопротивление* (импеданс, *impedance*) – полное сопротивление, которое встречает электромагнитная волна при распространении вдоль однородной цепи.

Импеданс измеряется в омах и зависит в основном от геометрии проводников и диэлектрической проницаемости материала изоляции.

Пусть сигнал распространяется в отрезке однородной цепи от источника в точке *A* до приемника в точке *B* (рис. 2.2).

Сигнал распространяется по медной цепи со скоростью, которая обычно лежит в пределах 60–80 % от скорости света в вакууме. Развитие события в точке *B* зависит от импеданса нагрузки кабеля (в данном случае – приемника).

Pair cable, TP) являются симметричными и используются для дифференциальной передачи сигналов. При скручивании проводники идут под некоторым углом друг к другу, что снижает емкостную и индуктивную связь между ними. Кроме того, такой кабель для внешних помех оказывается симметричным (круглым), что снижает его чувствительность к различным наводкам.

Конструктивно кабели могут быть:

- на основе экранированной витой пары *STP* (*Shielded TP*);
- неэкранированной витой пары *UTP* (*Unshielded TP*).

Кабели на основе витой пары подразделяются на категории. Категория витой пары *CAT* (*Category*) определяет частотный диапазон ее эффективного применения. В табл. 2.1 приведены категории и соответствующие им частотные диапазоны для *UTP*-кабелей.

Кроме общепринятой классификации кабелей по категориям, фирмой *IBM* введена классификация по типам (*Type*).

Таблица 2.1

Классификация *UTP*-кабелей

<i>CAT</i>	Полоса частот, МГц
1	0,1
2	1
3	16
4	20
5	100
5e	125
6	200 (250)
7	600

2.4.2. Волоконно-оптические кабельные системы

В волоконно-оптических кабельных системах сигналы передаются несущей оптического диапазона волн по световодам.

Волоконный световод – это диэлектрическая структура, по которой распространяются оптические сигналы, с длинами волны $0,85\text{--}1,6 \text{ мкм}$, что соответствует диапазону частот $(2,3\text{--}1,2) \cdot 10^{14} \text{ Гц}$.

Принцип действия волоконного световода основан на использовании известных процессов отражения и преломления оптической волны на границе раздела двух сред с различными показателями преломления.

При падении луча на границу раздела двух сред в общем случае появляются преломленная и отраженная волны. Согласно закону Снеллиуса угол падения $\Phi_{\text{п}}$ связан с углами отражения $\Phi_{\text{отр}}$ и преломления $\Phi_{\text{пр}}$ следующим соотношением:

3. Методы передачи данных

3. МЕТОДЫ ПЕРЕДАЧИ ДИСКРЕТНЫХ ДАННЫХ

- ▶ Аналоговая модуляция
- ▶ Дискретная модуляция аналоговых сигналов
- ▶ Цифровое кодирование
- ▶ Логическое кодирование

Выбор способа представления информации с помощью сигналов, подаваемых в канал связи, называется **линейным или физическим кодированием**.

При передаче дискретных данных по каналам связи применяются два основных типа физического кодирования:

- **аналоговое кодирование (модуляция)** – на основе синусоидального несущего сигнала. Кодирование осуществляется за счет изменения параметров аналогового сигнала;
- **цифровое кодирование** – на основе последовательности прямугольных импульсов.

Оба способа отличаются шириной спектра результирующего сигнала и сложностью аппаратуры, необходимой для их реализации.

Процесс представления аналоговой информации в дискретной форме называется **дискретной модуляцией**.

3.1. АНАЛОГОВАЯ МОДУЛЯЦИЯ

Аналоговая модуляция применяется для передачи дискретных данных по каналам с узкой полосой частот.

Примером такого канала является телефонный канал **тональной частоты**, который имеет полосу пропускания 3,1 кГц (для приемлемого качества передачи речи используются частоты от 300 до 3400 Гц).

Устройство, которое выполняет функции **модуляции** несущей синусоиды на передающей стороне и **демодуляции** на приемной стороне, носит название **модем** (модулятор – демодулятор).

Методы аналоговой модуляции:

- **амплитудная модуляция** – для кодирования «1» выбирается один уровень амплитуды синусоиды несущей частоты, а для «0» – другой;
- **частотная модуляция** – значения «0» и «1» исходных данных передаются синусоидами с различной частотой – f_0 и f_1 .
- **фазовая модуляция** – значениям данных «0» и «1» соответствуют сигналы одинаковой частоты со сдвигом фазы на некоторый постоян-

ный угол, например 45, 135, 225 и 315 градусов, через равные интервалы времени.

Спектр результирующего модулированного сигнала зависит от времени модуляции и скорости модуляции.

Заметим, что модуляционные технологии в чистом виде позволяют передавать один отсчет (т. е. на бод) два бита информации. Повысить скорость передачи за счет увеличения частоты также невозможно (из формулы Найквиста следует, что канальная частота можно вести передачу с частотой не выше 6,2 кГц).

Для повышения скорости передачи данных используют комбинированные методы модуляции. В качестве примера рассмотрим схему QAM (Quadrature Amplitude Modulation) квадратурной амплитудной модуляции, которая применяется в стандарте модема V.32 9600 бит/с.

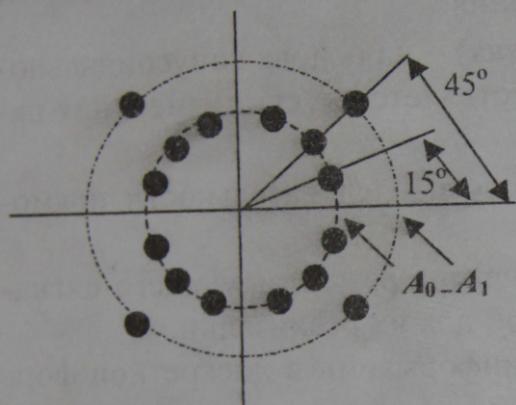


Рис. 3.1. Модуляция 4 бита/бод

ются запрещенными и используются для распознавания модемом ошибочных сигналов.

3.2. ДИСКРЕТНАЯ МОДУЛЯЦИЯ АНАЛОГОВЫХ СИГНАЛОВ

Рассмотрим принципы дискретной модуляции на примере импульско-кодовой модуляции ИКМ (Pulse Amplitude Modulation, PAM), которая широко применяется в цифровой телефонии (рис. 3.2):

1. Дискретизация по времени – с заданным периодом τ измеряется амплитуда исходной функции $s(t)$.
2. Оцифровывание – каждый замер амплитуды A_i представляется виде двоичного числа определенной разрядности (дискретизация по значениям функции).
3. Передача битовой последовательности оцифрованных замеров по каналам связи с использованием методов цифрового кодирования.

Эта схема использует комбинацию амплитудного и фазового методов модуляции. На рис. 3.1 показана диаграмма допустимых комбинаций амплитуды и фазы, называемая *диаграммой созвездия*.

В схеме использованы 2 уровня амплитуды (A_0 и A_1) и 12 значений сдвига фазы, из которых значащими выбраны 16 комбинаций, что позволяет передавать 4 бита/бод. Остальные комбинации являются запрещенными и используются для распознавания модемом ошибочных сигналов.

Устройство, которое выполняет эти функции, называется *аналогово-цифровым преобразователем (АЦП)*.

На приемной стороне с помощью специальной аппаратуры, называемой *цифро-аналоговым преобразователем (ЦАП)*, производится демодуляция оцифрованных амплитуд непрерывного сигнала, восстанавливая исходную непрерывную функцию времени.

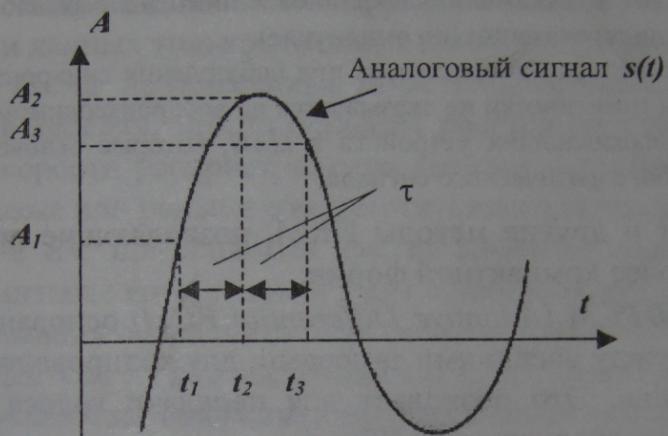


Рис. 3.2. Дискретная модуляция аналоговых сигналов

ИКМ основана на *теореме Найквиста*: аналоговая непрерывная функция $s(t)$, переданная в виде последовательности ее дискретных по времени значений A_i , может быть точно восстановлена, если частота дискретизации $f = 1/\tau \geq 2 * F$, где F — частота наивысшей гармоники спектра исходной функции.

Для передачи голоса в методе **ИКМ** используется частота квантования $f = 8000 \text{ Гц}$ ($\tau = 125 \text{ мкс}$). Эта частота квантования по теореме *Найквиста* достаточна для качественной передачи речи (для канала тональной частоты $F = 3400 \text{ Гц}$) и, кроме того, обеспечивает определенный запас качества.

Для оцифровывания замеров амплитуд обычно используется 7 или 8 бит кода. Соответственно, это дает 128 или 256 градаций звукового сигнала, что оказывается вполне достаточным для качественной передачи голоса.

Поэтому при использовании метода **ИКМ** для передачи одного 1-полосового канала необходима следующая пропускная способность:

- $7 * 8000 = 56\,000 \text{ бит/с} = 56 \text{ Кбит/с}$, если для оцифровывания используется 7 бит;
- $8 * 8000 = 64\,000 \text{ бит/с} = 64 \text{ Кбит/с}$ для случая восьми бит.

Цифровой канал 64 Кбит/с является стандартным и называется **стандартным каналом цифровых телефонных сетей**.

Передача непрерывного сигнала в дискретном виде требует синхронной передачи данных между узлами сети – соблюдения временного интервала $t = 125 \text{ мкс}$. При нарушении этого требования исходный сигнал восстанавливается неверно, что приводит к искажению информации, передаваемой по цифровым сетям.

На качество сигнала после ЦАП влияет также погрешность оцифровывания амплитуд, что приводит к искажению восстановленного непрерывного сигнала. Это называется шумом дискретизации (по амплитуде).

В то же время потеря одного замера при соблюдении синхронности между отдельными замерами практически не оказывается на воспроизводимом звуке. Это происходит за счет сглаживающих устройств в ЦАП, которые основаны на свойстве инерционности любого физического сигнала.

Существуют и другие методы ИКМ, позволяющие представить звуковые записи голоса в более компактной форме:

- Метод **ADPCM** (*Adaptive Differential PCM*) основан на нахождении разностей между соседними замерами, для кодирования которых используется 4 бита. Это позволяет для передачи голоса использовать цифровой канал со скоростью 32 Кбит/с.

- Метод **LPC** (*Linear Predictive Coding*) позволяет уменьшить число точек дискретизации, но использует прогнозирование направления изменения амплитуды. Данный метод позволяет снизить требования к скорости цифрового канала до 9,6 Кбит/с.

3.3. ЦИФРОВОЕ КОДИРОВАНИЕ

3.3. ЦИФРОВОЕ КОДИРОВАНИЕ

При цифровом кодировании дискретной информации на основе прямоугольных импульсов применяют следующие коды:

- **потенциальные** – для представления двоичных данных используется только значение потенциала сигнала;
- **импульсные** – двоичные данные представляются либо импульсами определенной полярности, либо частью импульса – перепадом потенциала определенного направления.

Выбор способа кодирования требует достижения нескольких целей:

- иметь наименьшую ширину спектра результирующего сигнала при одной и той же битовой скорости;
- обеспечивать синхронизацию между передатчиком и приемником;

- обладать способностью распознавать ошибки;
- обладать низкой стоимостью реализации.

Более узкий спектр сигналов позволяет достичь более высокой скорости передачи.

ности передачи данных на одном и том же канале. Кроме того, к спектру сигнала предъявляется требование отсутствия постоянной составляющей, т. е. наличия постоянного тока между передатчиком и приемником.

Синхронизация передатчика и приемника нужна для определения приемником точного момента времени считывания переданной информации с канала связи.

Эта проблема на небольших расстояниях решается с помощью схемы, основанной на отдельном тактирующем канале – информация снимается с линии данных только в момент прихода тактового импульса.

В КС эта схема не применима из-за неоднородности характеристик проводников в кабелях, что на больших расстояниях приводит к неравномерности скорости распространения сигнала – тактовый импульс может прийти позже или раньше соответствующего сигнала данных.

Поэтому в КС применяются так называемые *самосинхронизирующиеся коды*, сигналы которых несут для приемника указания на моменты времени считывания очередного бита (или нескольких бит, если код ориентирован более чем на два состояния сигнала). Примером такого указания для синхронизации приемника может служить любой резкий перепад сигнала, так называемый фронт сигнала.

Примечание. При использовании синусоидального несущего сигнала результатирующий код обладает свойством самосинхронизации, так как изменение амплитуды несущей частоты дает возможность приемнику определить момент появления входного кода.

Средствами физического уровня сложно решить проблему распознавания и коррекции искаженных данных. В большинстве сетевых технологий это решается протоколами выше лежащих уровней. Однако, если код позволяет обнаруживать ошибки, это существенно повышает производительность сетевой передачи.

Рассмотрим наиболее распространенные схемы кодирования, применяемые в КС.

1. Примеры потенциальных кодов:

- **NRZ (Non Return to Zero)** – потенциальный код без возврата к нулю. Это двухполярная схема, имеющая два варианта. В обычной схеме (рис. 3.3, а) бит «1» представляется напряжением $+V$, бит «0» – нулевым напряжением. В дифференциальном NRZ (рис. 3.3, б) состояние меняется в начале битового интервала «1» и не меняется для «0». Нет привязки «1» и «0» к определенному состоянию.

Метод NRZ прост в реализации, обладает хорошей распознаваемостью ошибок (из-за двух резко отличающихся потенциалов), но не обладает свойством самосинхронизации – возможны постоянные состав-

ляющие из нулей либо единиц. Достоинством кода **NRZ** является достаточно низкая частота основной гармоники $F = C/2 \Gamma_0$, где C – скорость передачи данных по каналу (бит/с).

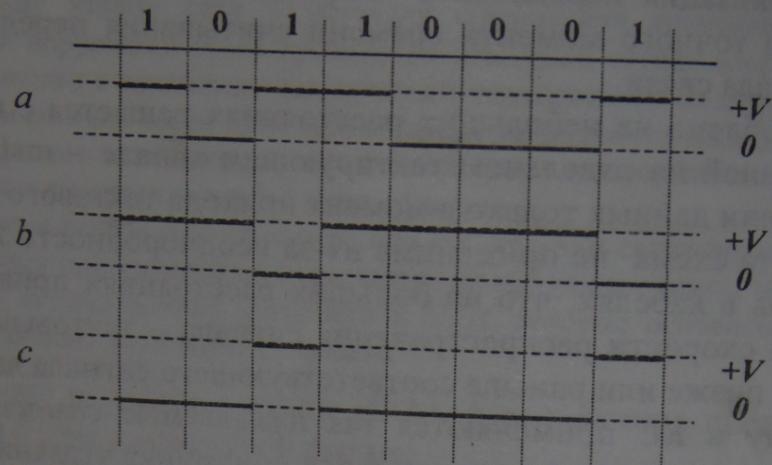


Рис. 3.3. Кодирование NRZ

- **NRZI** (*Non Return to Zero with ones Inverted*) – потенциальный код с инверсией при единице (модифицированный вариант **NRZ**). «0» передается на том же уровне потенциала, который был установлен в предыдущем такте, а при передаче «1» потенциал инвертируется на противоположный (рис. 3.3, *c*). Устраниены постоянные составляющие из «1», но возможен постоянный сигнал при передаче «0».

- **AMI** (*Alternate Mark Inversion*) – метод биполярного кодирования с альтернативной инверсией, в котором используются три уровня потенциала $-V$, 0 , $+V$. «0» кодируется потенциалом 0 , а «1» – либо $+V$, либо $-V$, при этом потенциал каждой новой единицы противоположен потенциалу предыдущей.

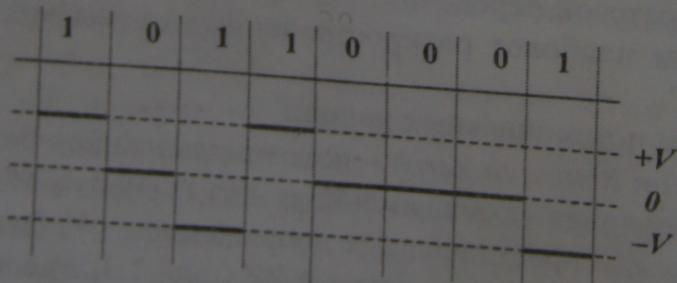


Рис. 3.4. Кодирование AMI

Код **AMI** частично решает проблему постоянной составляющей и не является полностью самосинхронизирующим.

По сравнению с **NRZ** код **AMI** имеет более узкий спектр: при передаче последовательности из чередующихся нулей и единиц основная

гармоника $F = C/4$ Гц. Кроме того, отсутствие строгого чередования полярности сигнала говорит об ошибке.

- **MLT-3** – несамосинхронизирующее трехуровневое кодирование $(-V, 0, +V)$, постоянное в течение каждого битового интервала (рис. 3.5). При передаче «0» значение не меняется, а при передаче «1» значения меняются по цепочке $+V, 0, -V, 0, +V$ и т. д.

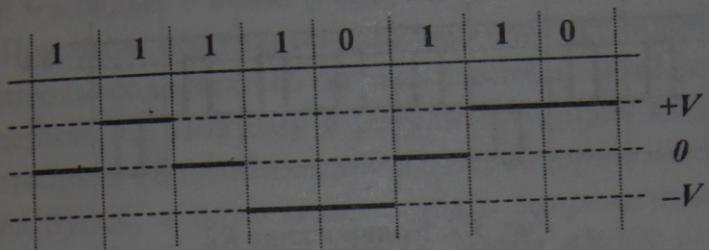


Рис. 3.5. Кодирование MLT-3

2. Примеры импульсных кодов:

- **Манчестерский код (Manchester encoding).** Текущий бит определяется по направлению смены состояния в середине битового интервала: от $-V$ к $+V$ – «1», от $+V$ к $-V$ – «0» (рис. 3.6, a). Спектр кода: в среднем основная гармоника $F = 3/4 * C$ Гц.

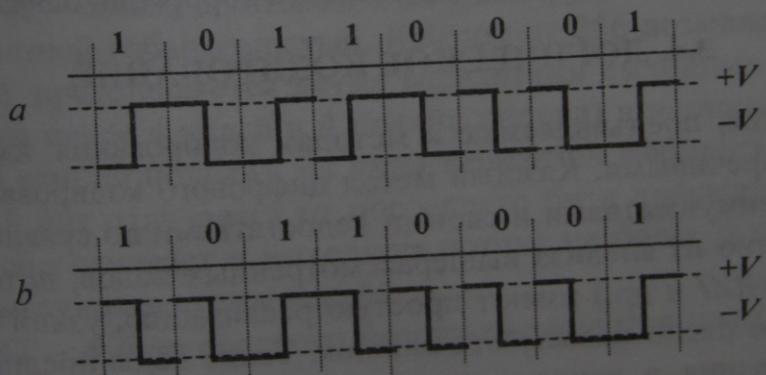


Рис. 3.6. Манчестерское кодирование

В дифференциальном манчестерском коде (рис. 3.6, b) текущий бит определяется по наличию перехода в начале битового интервала: «0» – есть переход, «1» – нет перехода. Возможно и противоположное определение «0» и «1». Сложная реализация кода. В связи с тем что на каждом такте сигнал меняет свое состояние, то для передачи одного бита требуется удвоенная частота тактового генератора (имеем 2 бод на 1 бит).

- **RZ** (*Return to Zero*) – самосинхронизирующая схема с тремя состояниями ($-V$, 0 , $+V$).

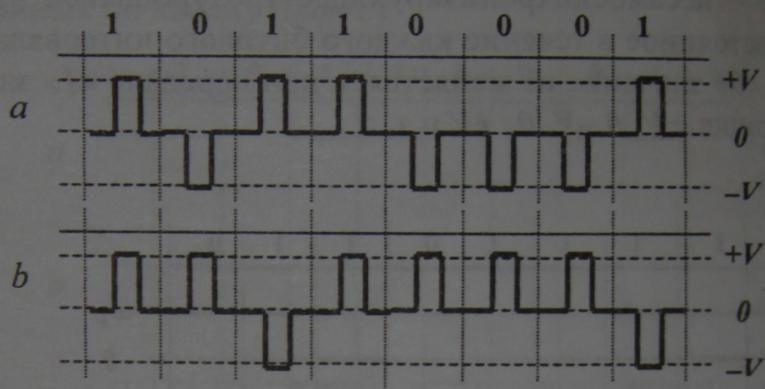


Рис. 3.7. Кодирование RZ

В определенный момент битового интервала состояние всегда возвращается к нулю. Имеет обычный вариант (рис. 3.7, *a*) и дифференциальный (рис. 3.7, *b*), в котором нет привязки «0» и «1» к определенному состоянию.

3. Примеры кодов с несколькими состояниями:

- **PAM 5** (*Pulse Amplitude Modulation*) – пятиуровневое биполярное кодирование, при котором пара бит, в зависимости от предыстории, представляется одним из 4 уровней потенциала ($-2V$, $-1V$, $+1V$, $+2V$), пятый ($0V$) используется для обнаружения и коррекции ошибок.

5. Управление каналом связи

4. УПРАВЛЕНИЕ КАНАЛАМИ СВЯЗИ

- Формирование кадров
- Службы передачи кадров
- Обнаружение и коррекция ошибок
- Методы восстановления искаженных и потерянных кадров
- Управление потоком
- Компрессия данных
- Протокол HDLC

Управление каналами связи решается с помощью протоколов канального уровня. Протоколы канального уровня оперируют с блоками данных, которые называются *кадрами* (*Frame*).

Основными задачами протоколов канального уровня являются:

- формирование кадров для передачи пакетов данных между физически связанными узлами *КС* (основная служба, предоставляемая сетевому уровню);
- нахождение границ кадра в потоке бит, передаваемых на физическом уровне;
- обработка ошибок передачи и управление потоком кадров.

Примечание. Протоколы канального уровня имеют локальный смысл, они предназначены для доставки кадров данных, как правило, в пределах сетей с простой топологией связей и однотипной или близкой технологией.

4.1. ФОРМИРОВАНИЕ КАДРОВ

В зависимости от метода передачи протоколы, работающие на канальном уровне, могут быть разделены на две группы – *асинхронные* и *синхронные*.

4.1.1. Асинхронные протоколы

Асинхронные протоколы оперируют не с кадрами, а с отдельными символами. Символ представляет байт данных, сопровождаемый специальными сигналами «*Start*», «*Stop*» (рис. 4.1). Эти сигналы предназначены для синхронизации передатчика и приемника при передаче каждого байта.

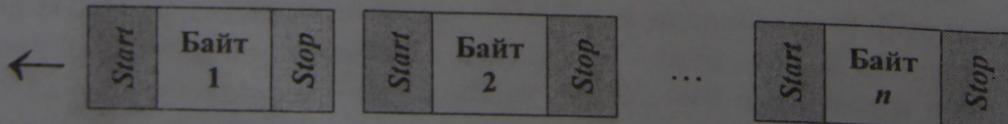


Рис. 4.1. Асинхронная передача

В асинхронных протоколах применяются стандартные наборы символов, чаще всего *ASCII* или *EBCDIC*. Так как первые 32 или 27 кодов в этих наборах являются специальными кодами, которые не отображаются на дисплее или принтере, то они используются асинхронными протоколами для управления режимом обмена данными.

Такой метод передачи пригоден для связи низкоскоростных устройств на небольших расстояниях, например для связи клавиатур или дисплеев с компьютером.

В асинхронных протоколах применяются и более сложные методы передачи: наряду с отдельными символами использовать целые блоки данных.

Пример. Протокол *XMODEM* обеспечивает передачу файлов между двумя компьютерами по асинхронному модему, в котором сочетается посылка отдельных символов с посылкой блоков данных. Алгоритм его работы следующий:

- принимающая сторона передает символ *NAK*, который означает готовность к приему;
- передающая сторона, приняв *NAK*, отправляет очередной блок файла, состоящий из 128 байт данных, заголовка и концевика. Заголовок состоит из специального символа *SOH* (*Start Of Header*) и номера блока. Концевик содержит контрольную сумму блока данных;
- приемная сторона, получив новый блок, проверяет его номер и контрольную сумму. В случае совпадения этих параметров с ожидаемыми, приемник отправлял символ *ACK*, а в противном случае – символ *NAK*, после чего передатчик должен был повторить передачу данного блока;
- в конце передачи файла передавался символ *EOX*.

Символы *NAK*, *SOH*, *ACK*, *EOX* выбраны из управляющих символов *ASCII*.

4.1.2. Синхронные протоколы

В синхронных протоколах обмен данными осуществляется кадрами, общий формат которых приведен на рис. 4.2. Все биты кадра передаются непрерывным синхронным потоком, что значительно ускоряет передачу данных.

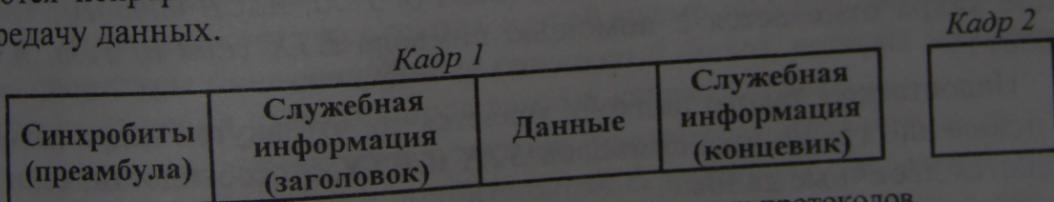


Рис. 4.2. Общий формат кадров синхронных протоколов

При синхронной передаче старт-стопные биты отсутствуют, поэтому для надежной синхронизации приемника и передатчика используются

байты (один или несколько) *синхронизации*. Синхробайты содержат известные коды, которые извещают приемник о приходе кадра данных. При его получении приемник должен войти в *режим байтовой синхронизации*, т. е. правильно понимать начало байта.

При передаче длинного кадра могут возникнуть проблемы с синхронизацией битов, в связи с чем необходимо применение на физическом уровне самосинхронизирующихся кодов.

Затем приемник должен найти начало и конец кадра, а также определить границы каждого поля кадра: адресную информацию и других служебных полей заголовка, поля данных и контрольной суммы, которая обычно составляет концевик кадра.

Большинство протоколов допускает использование в кадре поля данных переменной длины. Поэтому канальные протоколы определяют максимальное значение, которое может иметь длина поля данных. Эта величина называется *максимальной единицей передачи данных MTU (Maximum Transfer Unit)*.

В некоторых протоколах задается также минимальное значение, которое может иметь длина поля данных.

В зависимости от методов решения вышепоставленных задач синхронные протоколы канального уровня разделяют на два типа:

- *символьно-ориентированные;*
- *бит-ориентированные.*

Для обоих характерны одни и те же методы синхронизации бит. Главное различие между ними заключается в методе синхронизации символов и кадров.

Символьно-ориентированные протоколы используются в основном для передачи блоков отображаемых символов, например текстовых файлов. В качестве *синхробайтов* используются два или более управляющих символа, называемых символами *SYN* (в коде ASCII символ *SYN* имеет двоичное значение *0010110*). Границы начала кадра указываются с помощью специального символа *STX* (*Start of TeXt*, ASCII *0000010*). Окончание кадра отмечается с помощью символа *ETX* (*End of TeXt*, ASCII *0000011*).

Недостатком такого подхода является то, что внутри кадра возможно появление граничных символов *STX* и *ETX*, в особенности, если передаются двоичные данные.

Кодопрозрачность протокола, т. е. его способность отличать граничные символы от символов данных кадра, совпадающих по кодам с граничными, достигается за счет процедуры, называемой *байт-страффингом* (*stuff* – заполнитель): перед символами *STX* и *ETX* *внутри*

кадра всегда вставлялся символ **DLE** (*Data Link Escape*). А если в поле данных кадра встречается последовательность **DLE ETX**, то передатчик порождает последовательность **DLE DLE ETX**, а приемник, встретив подряд два символа **DLE DLE**, всегда удалял первый, а оставшиеся символы **DLE ETX** считает пользовательскими данными.

Отметим, что символьно-ориентированные протоколы имеют большую избыточность за счет дополнительных символов **DLE**, что приводит к неэффективности передачи двоичных данных. Кроме того, формат управляющих символов для разных кодировок различен, поэтому этот метод допустим только с определенным типом кодировки, даже если кадр содержит чисто двоичные данные.

Бит-ориентированные протоколы реализуют более универсальный метод передачи. Рассмотрим базовые схемы их построения, которые, по сути, различаются лишь способом определения начала и конца кадра.

Схема 1. Начало и конец каждого кадра отмечается одной и той же 8-битовой последовательностью – **01111110**, называемой *флагом* (рис. 4.3).

Термин «бит-ориентированный» используется потому, что принимаемый поток бит сканируется приемником на побитовой основе для обнаружения начального флага, а затем во время приема для обнаружения конечного флага. Поэтому длина кадра в этом случае не обязательно должна быть кратна 8 бит.

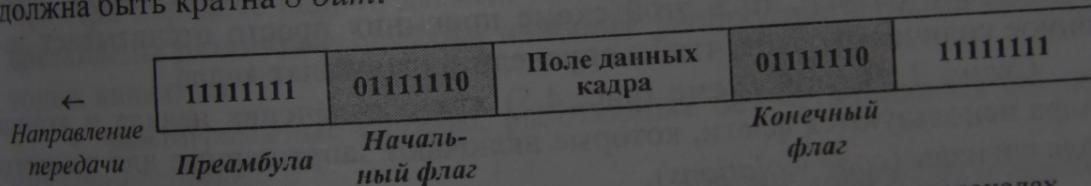


Рис. 4.3. Схема 1 формирования кадров в бит-ориентированных протоколах

Для синхронизации приемника и передатчика в качестве синхробайтов (*пreamble*) посыпается последовательность байтов *простоя*, каждый из которых равен **1111111**.

Кодопрозрачность протокола в этой схеме обеспечивается с помощью процедуры *бит-стаффинга* (вставка 0 бита), которая применяется только во время передачи поля данных кадра:

- если передатчик обнаруживает, что подряд передано пять 1, то автоматически вставляет дополнительный 0, даже если после этих пяти 1 шел 0. Поэтому флаговая последовательность **01111110** никогда не появится в поле данных кадра;
- приемник выполняет обратную функцию – если после пяти 1 обнаруживает 0, то автоматически удаляет его из поля данных кадра.

Бит-стаффинг имеет меньшую избыточность, чем байт-стаффинг, так как вместо лишнего байта вставляется один бит.

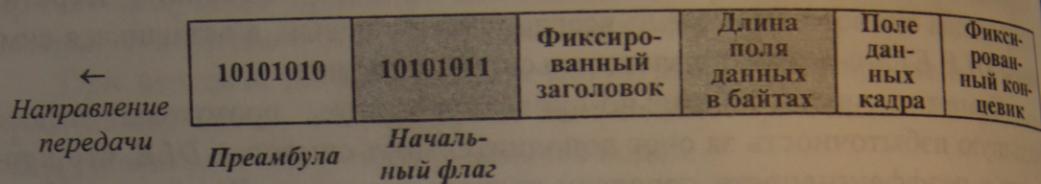


Рис. 4.4. Схема 2 формирования кадров в бит-ориентированных протоколах

Схема 2. В этой схеме (рис. 4.4) для обозначения начала кадра имеется только *начальный флаг*, а для определения конца кадра используется *поле длины кадра*, которое при фиксированных размерах заголовка и концевика чаще всего имеет смысл *длины поля данных кадра*.

Эта схема наиболее применима в локальных сетях, в которых незаданность среды определяется по отсутствию несущей частоты в канале.

Для обеспечения *битовой синхронизации* посылающая станция предваряет содержимое кадра последовательностью бит, известной как *преамбула*, которая состоит из чередования единиц и нулей.

Войдя в битовую синхронизацию, приемник исследует входной поток на побитовой основе, пока не обнаружит *флаговый байт* начала кадра. Поскольку служебные поля кадра (заголовок и концевик) имеют фиксированный размер, то в этой схеме приемник просто отсчитывает заданное количество байт, чтобы определить окончание кадра.

Схема 3. В этой схеме (рис. 4.5) для обозначения начала и конца кадра используются флаги, которые включают запрещенные для данного кода сигналы (*code violations*).

Пример. При манчестерском кодировании начало кадра отмечается последовательностью *JK0JK000*, а конец – последовательностью *JK1JK100*, где *J* и *K* – запрещенные для данного метода кодирования сигналы: вместо обязательного изменения полярности сигнала в середине тактового интервала уровень сигнала *J* остается неизменным и низким, а уровень сигнала *K* – неизменным и высоким.

Эта схема очень экономична: не требует ни бит-стаффинга, ни поля длины данных. Недостатком схемы 3 является ее зависимость от принятого метода физического кодирования.

При использовании избыточных кодов роль сигналов *J* и *K* играют запрещенные символы, например, в коде *4B/5B* этими символами являются коды *11000* и *10001*.

4.1.3. Протоколы с гибким форматом кадра

Однако существует ряд протоколов, в которых кадры имеют более гибкую структуру. Кадры таких протоколов состоят из *неопределенного* количества полей, каждое из которых может иметь *переменную* длину.

Начало такого кадра отмечается некоторым стандартным образом, например с помощью начального флага, а затем протокол последовательно просматривает поля кадра и определяет их количество и размеры.

Способ представления данных в таких протоколах получил название *TLV* (*Type*, *Length*, *Value*), в котором каждому полю кадра (*Type* – значение) обычно предшествуют два дополнительных поля фиксированного размера (*Type* – тип и *Length* – длина).

Поскольку количество таких полей неизвестно, для определения общей длины кадра используется либо общее поле «Длина», которое помещается в начале кадра и относится ко всем полям данных, либо конечный флаг.

По такой схеме построен прикладной протокол управления сетями *SNMP* (*Simple Network Management Protocol*), а также протокол канального уровня *PPP* (*Point_to_Point Protocol*), используемый для соединений типа «точка-точка».

6. Обнаружение и коррекция ошибок

4.3. ОБНАРУЖЕНИЕ И КОРРЕКЦИЯ ОШИБОК

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

4.3.1. Методы обнаружения ошибок

Все методы обнаружения ошибок основаны на передаче в составе кадра данных *служебной избыточной информации*, по которой можно судить с некоторой степенью вероятности о достоверности принятых данных. Эту служебную информацию принято называть *контрольной суммой* или *последовательностью контроля кадра FCS (Frame Check Sequence)*.

Коды, которые применяются в алгоритмах вычисления контрольной суммы, называются *помехоустойчивыми*. Они позволяют обнаруживать ошибки либо их исправлять. Коды, исправляющие ошибки, называются *корректирующими* или кодами с исправлением ошибок.

Рассмотрим наиболее распространенные коды, используемые для обнаружения ошибок:

1. *Контроль по паритету*. Метод заключается в *суммировании по модулю 2* всех бит контролируемого блока данных. Результат суммирования (один бит данных – *бит паритета*) пересыдается вместе с контролируемой информацией. Такой код позволяет обнаруживать только одиночные ошибки. Применение контроля по паритету к каждому байту контролируемого блока дает очень высокий коэффициент избыточности.

Модификацией данного метода является *вертикальный и горизонтальный контроль по паритету*. Контролируемый блок данных рассматривается как матрица шириной n бит и высотой k бит. Биты парите-

та рассчитываются отдельно для каждой строки и каждого столбца матрицы, что позволяет обнаруживать большую часть двойных ошибок. Однако большая избыточность кода ограничивает его применение на практике.

2. Циклический избыточный контроль **CRC** (*Cyclic Redundancy Check*). Это полиномиальный код, который наиболее широко используется в компьютерных сетях.

В основу кода положено представление битовых строк в виде полиномов с коэффициентами 0 или 1. Кадр из k бит рассматривается как полином степени $k - 1$. Например, строка 110001 представляется как полином 5-й степени $P(x) = x^5 + x^4 + x^0$.

На множестве полиномов $P(x)$ определим операции сложения и вычитания по модулю 2, при этом операция сложения не отличается от вычитания, перенос в следующий или предыдущий разряд не производится. Деление будем выполнять как деление обычных двоичных чисел, при котором вычитание производится по модулю 2.

Для использования кода **CRC** отправитель и получатель определяют образующий полином $G(x)$, старший и младший биты которого должны быть равны 1. Пусть r — степень полинома $G(x)$.

Алгоритм вычисления контрольной суммы для кадра $M(x)$, содержащего m бит, следующий:

1. Построим полином $M_1(x) = x^r M(x)$, который соответствует контролируемому кадру, дополненному справа r нулевыми битами (кадр теперь содержит $m + r$ бит).

2. Вычислим $R(x) = M_1(x) \bmod G(x)$, который будет иметь степень не более r .

3. Вычислим $T(x) = M_1(x) - R(x)$, который будет соответствовать передаваемому кадру.

4. Получатель, приняв кадр, вычисляет $R_1(x) = T(x) \bmod G(x)$. При успешной передаче кадра $R_1(x) = 0$, ненулевой $R_1(x)$ будет означать ошибку.

На практике в протоколах канального уровня под поле контрольной суммы отводится два или четыре байта, соответственно производящие полиномы имеют степень 16 или 32. Для того чтобы подчеркнуть используемый в протоколе размер контрольной суммы, применяют нотации **CRC-16** или **CRC-32**.

Проанализируем возможности данного метода. Пусть получатель принял ошибочный кадр $T_1(x)$, который можно представить как $T_1(x) = T(x) + E(x)$, где каждый бит I полинома $E(x)$ соответствует инвертированному (ошибочному) биту кадра. Если $E(x)$ содержит k бит, равных 1, это значит произошло k единичных ошибок.

Вычислим $R_1(x) = T_1(x) \bmod G(x) = (T(x) + E(x)) \bmod G(x) = E(x) \bmod G(x)$:

- если $E(x)$ окажется кратным $G(x)$, то ошибки не будут обнаружены;
- если произошла одиночная ошибка, т. е. $E(x) = x^i$, где i – номер ошибочного бита, то она будет обнаружена – $E(x)$ никогда не будет кратен $G(x)$;
- в случае двух изолированных ошибок представим $E(x) = x^i + x^j$ ($i > j$), в виде $E(x) = x^i(x^{i-j} + 1)$. Если предположить, что $G(x)$ не делится на x^i , то достаточным условием обнаружения всех двойных ошибок будет неделимость на $G(x)$ полинома $x^k + 1$ для любого k от 1 до максимального значения $i-j$.

Например, полином невысокой степени $x^{15} + x^{14} + 1$ не является делителем для $x^k + 1$ при $\forall k \subseteq [1, 32768]$;

- в случае нечетного числа ошибочных бит $E(x)$ будет содержать нечетное число членов. В системе арифметических операций по модулю 2 известно, что полиномы с нечетным числом членов не делятся на $x + 1$. Поэтому, если в качестве $G(x)$ выбрать полином, делящийся на $x + 1$, то с его помощью можно обнаружить все ошибки, состоящие из нечетного числа инвертированных бит;
- в случае пакета ошибок длиной k ($k \leq r$) $E(x)$ можно представить как $E(x) = x^i(x^{k-i} + \dots + 1)$, где i определяет, насколько далеко от правого края располагается пакет ошибок. Если $G(x)$ содержит x^0 , то x^i не будет его множителем. Поэтому, если выражение в скобках меньше степени $G(x)$, то $E(x) \bmod G(x) \neq 0$. Следовательно, CRC с r контрольными битами позволяет обнаруживать все пакеты ошибок длиной $\leq r$.

При длине пакета ошибок $r+1$ остаток от деления будет нулевым, если пакет идентичен $G(x)$. По определению пакета ошибок его первый и последний биты равны 1, поэтому будет ли совпадать он с $G(x)$, зависит от $r-1$ промежуточных бит. Если все комбинации считать равновероятными, то вероятность нераспознанной ошибки будет равна 0.5^{r-1} . Аналогично можно провести оценку и для пакетов ошибок длиннее $r+1$.

Примеры стандартных образующих полиномов:

$$\text{CRC-16} = x^{16} + x^{15} + x^2 + 1;$$

$$\text{CRC-ITU} = x^{16} + x^{12} + x^5 + 1;$$

$$\text{IEEE 802} = x^{32} + x^{26} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1.$$

Все они делятся на $x+1$.

К недостаткам метода можно отнести его высокую вычислительную сложность. Для повышения производительности метода Питерсон

(Peterson) и Браун (Brown) предложили простую схему для аппаратного подсчета и проверки контрольной суммы на основе сдвигового регистра, которая применяется почти во всей аппаратуре.

Отметим, что *CRC*-метод обладает также невысокой степенью избыточности. Например, для кадра *Ethernet* размером в 1024 байт контрольная информация длиной в 4 байт составляет только 0,4 %.

4.3.2. Методы восстановления ошибочных и потерянных кадров

Методы коррекции ошибок в КС основаны на *повторной* передаче кадра данных в случае его потери (не доходит до адресата) или если приемник обнаружил в нем искажение информации (не пройден контроль, например, по контрольной сумме). Для решения проблемы по повторной передаче кадра необходимо ввести:

- нумерацию отправляемых кадров;
- служебные кадры для подтверждения приема кадров данных;
- таймер, ограничивающий время ожидания подтверждения приема.

Служебные кадры, подтверждающие прием кадров данных, называются *квитанциями*. Квитанции могут быть положительными или отрицательными. Для обозначения *положительной квитанции* будем использовать символ *ACK* (*Acknowledgement*), *отрицательной* – *NAK* (*Not Acknowledgement*).

Поскольку время ожидания подтверждения ограничено, важную роль в процессе подтверждения играют *таймеры*. При отправке каждого кадра передатчик запускает таймер, и если по его истечении квитанция (*положительная* или *отрицательная*) не получена, кадр считается утерянным. Однако, если квитанция запаздывает, возникает проблема дублирования передаваемых кадров.

Рассмотрим некоторые подходы к организации процесса обмена квитанциями.

1. *Метод с простоями* (*Idle Source*). Источник, пославший кадр, ожидает получения квитанции (*положительной* или *отрицательной*) от приемника и только после этого посыпает следующий кадр (или повторяет искаженный). Если же квитанция не приходит в течение тайм-аута, то кадр (или квитанция) считается утерянным и его передача повторяется. Кроме того, каждый кадр должен иметь некоторый последовательный номер, что позволило бы приемнику контролировать приход всех кадров в требуемом порядке.

Этот метод имеет низкую производительность обмена данными за счет простоеев на ожидание подтверждений.

2. *Метод «скользящего окна» (sliding window).* В этом методе каждый передаваемый кадр содержит порядковый номер S , который варьируется от 0 до $2^n - 1$, где n – количество бит, отводимых под номер.

Для передачи и приема кадров используются посылающее и принимающее окна. Окно можно представить как циклический буфер, позволяющий разместить $M = 2^n$ кадров. Будем говорить, что задано окно по модулю M .

Сущность метода заключается в том, что отправитель может передать некоторое количество кадров в *непрерывном режиме* (без получения квитанции подтверждения), помещая их в посылающее окно. Максимальное количество кадров, которые можно отправить, ограничивается размером окна w ($1 \leq w \leq 2^n - 1$).

Отметим, что размер окна w может не использоваться. В этом случае отправитель полностью использует всю предоставляемую окном нумерацию для отправляемых кадров.

Если окно содержит w кадров для передачи, то отправитель не имеет права посылать следующие кадры. В этом случае мы говорим, что окно закрыто.

Аналогично получатель работает с принимающим окном, сохраняя в нем кадры, которые ему разрешено принимать, т. е. до тех пор, пока окно открыто. Получатель проверяет полученные кадры на корректность. Пусть R – номер последнего правильно принятого кадра. Правильно полученные кадры передаются службам для обработки, а в ответе получатель передает номер $R + 1$.

Заметим, что получателю нет необходимости посыпать квитанции на каждый принятый корректный кадр. Если пришло несколько кадров, то приемнику достаточно послать квитанцию только на последний кадр. При этом подразумевается, что все предыдущие кадры также получены корректно.

Для отправителя это означает, что получатель успешно принял все кадры с номерами до R и ожидает следующий кадр с номером $R + 1$. Получив такое подтверждение, посылающее окно сдвигается – нижняя граница окна устанавливается на кадр с номером R . Соответственно, верхняя граница также передвигается, предоставляя отправителю новое пространство в посылающем окне – отправитель теперь может передавать кадры с номерами до $R + w - 1$.

на который получена квитанция, номер кадра, который еще можно передать до получения новой квитанции.

При ошибочных и потерянных кадрах обычно используют две стратегии:

- *выборочный повтор* – получатель запрашивает повторить передачу ошибочных и потерянных кадров, сохраняя в буфере последние, правильно полученные кадры;
- *возврат на n* – получатель просто игнорирует все кадры, полученные вслед за ошибочным и требует повторить все кадры, начиная с ошибочного.

7. Коммутация и мультиплексирование

5. МЕТОДЫ КОММУТАЦИИ

- ▶ Коммутация каналов
- ▶ Частотное мультиплексирование
- ▶ Коммутация каналов на основе разделения времени
- ▶ Технология плотного волнового мультиплексирования
- ▶ Множественный доступ с кодовым разделением каналов
- ▶ Первичные сети: *PDH* и *SDH*
- ▶ Коммутация пакетов
- ▶ Коммутация сообщений

Существующие системы телекоммуникаций предназначены для создания коммутируемой инфраструктуры, которая позволяет быстро и гибко создавать *каналы связи* между пользовательскими устройствами. Их основу составляют *первичные*, или *опорные*, сети. *Компьютерные сети* строятся как *вторичные сети*, поскольку работают на основе инфраструктуры первичных сетей, поэтому их часто называют *наложенным* (*overlay*).

Любая сеть связи должна обеспечивать доступность имеющихся каналов связи для абонентов сети (удаленных компьютеров, локальных сетей и т. д.).

Существуют три принципиально различные схемы коммутации абонентов в сетях:

- *коммутация каналов* (*circuit switching*);
- *коммутация пакетов* (*packet switching*);
- *коммутация сообщений* (*message switching*).

5.1. КОММУТАЦИЯ КАНАЛОВ

Коммутация каналов подразумевает образование *непрерывного составного физического канала* из последовательно соединенных отдельных канальных участков для *прямой* передачи данных между узлами.

Отдельные каналы соединяются между собой специальной аппаратурой – *коммутаторами*, которые могут устанавливать связи между любыми конечными узлами сети. В сети с коммутацией каналов перед передачей данных всегда необходимо выполнить процедуру установления *соединения*, в процессе которой и создается составной канал.

Сети с коммутацией каналов могут быть разделены на два класса:

- *сети с динамической коммутацией*;
- *сети с постоянной коммутацией*.

В первом случае коммутация выполняется на время сеанса по инициативе одного из взаимодействующих пользователей при выполнении определенной работы (передачи файла, просмотра страницы текста, изображения и т. п.).

Во втором случае разрешается паре пользователей заказать соединение на длительный период времени, которое устанавливается персоналом, обслуживающим сеть. Режим постоянной коммутации в сетях с коммутацией каналов называется *сервисом выделенных (dedicated) или арендуемых (leased) каналов*.

Коммутаторы, а также соединяющие их каналы должны обеспечивать одновременную передачу данных нескольких абонентских каналов или нескольких сеансов связи. Эта задача решается с помощью техники **мультиплексирования**.

В настоящее время для мультиплексирования абонентских каналов используются две техники:

- техника частотного мультиплексирования **FDM** (*Frequency Division Multiplexing*);
- техника мультиплексирования с разделением времени **TDM** (*Time Division Multiplexing*);
- техника мультиплексирования по длине волны **WDM** (*Wave Division Multiplexing*).

5.2. ЧАСТОТНОЕ МУЛЬТИПЛЕКСИРОВАНИЕ

Рассмотрим особенности этой техники мультиплексирования на примере телефонной сети.

Основные гармоники речевого сигнала укладываются в диапазон от 300 до 3400 Гц, поэтому для качественной передачи речи достаточно образовать между двумя абонентами канал с полосой пропускания в 3100 Гц.

В то же время полоса пропускания кабельных систем, соединяющих коммутаторы между собой, может составлять до сотни мегагерц. Но передавать одновременно сигналы нескольких абонентских каналов по широкополосному каналу невозможно, так как все они работают в одном и том же диапазоне частот, и сигналы разных абонентов будут смешиваться между собой так, что разделить их будет невозможно.

Для разделения абонентских каналов используется *техника модуляции высокочастотного несущего синусоидального сигнала низкочастотным речевым сигналом*. В результате спектр модулированного сигнала переносится в другой диапазон, который симметрично распола-

В сетях на основе **FDM**-коммутации принято несколько уровней иерархии уплотненных каналов:

- **базовая группа** составляет первый уровень уплотнения;
- второй уровень уплотнения образуют 5 базовых групп, которые составляют **супергруппу**, с полосой частот шириной в 240 кГц и границами от 312 до 552 кГц. Супергруппа передает данные 60 абонентских каналов;
- десять супергрупп образуют **главную группу**, которая используется для связи между коммутаторами на больших расстояниях. Главная группа передает данные 600 абонентов одновременно и требует от канала связи полосу пропускания шириной не менее 2520 кГц с границами от 564 до 3084 кГц.

Коммутаторы **FDM** могут выполнять как динамическую, так и **постоянную** коммутацию. При **динамической** коммутации коммутатор выделяет данному абоненту одну из свободных полос своего уплотненного канала. При **постоянной** коммутации за абонентом полоса в 4 кГц закрепляется на длительный срок путем настройки коммутатора по отдельному входу, недоступному другим пользователям.

5.3. КОММУТАЦИЯ КАНАЛОВ НА ОСНОВЕ РАЗДЕЛЕНИЯ ВРЕМЕНИ

Коммутация на основе техники разделения частот разрабатывалась в расчете на передачу непрерывных сигналов, представляющих голос.

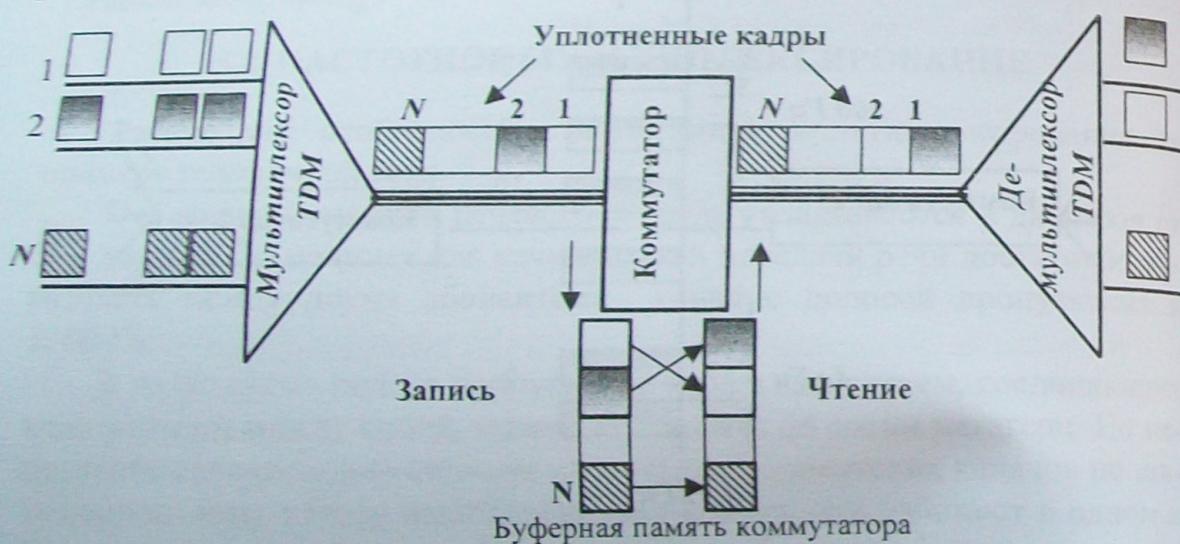


Рис. 5.2. Коммутация на основе *TDM*

При переходе к цифровой форме представления голоса была разработана новая техника мультиплексирования, ориентирующаяся на дискретный характер передаваемых данных.

Эта техника носит название мультиплексирования с разделением времени **TDM** (Time Division Multiplexing). Реже используется и другое ее название – техника синхронного режима передачи **STM** (Synchronous Transfer Mode).

Цикл работы оборудования **TDM** равен 125 мкс, что соответствует периоду следования замеров голоса в цифровом абонентском канале. Каждому соединению выделяется один квант времени цикла работы аппаратуры, называемый также **тайм-слотом**. Длительность тайм-слота зависит от числа абонентских каналов, обслуживаемых мультиплексором или коммутатором.

Мультиплексор TDM принимает информацию по N входным каналам от конечных абонентов, каждый из которых передает данные по абонентскому каналу со скоростью 64 Кбит/с – 1 байт каждые 125 мкс (рис. 5.2).

В каждом цикле мультиплексор выполняет следующие действия:

- прием от каждого канала очередного байта данных;
- составление из принятых байтов уплотненного кадра;
- передача уплотненного кадра на выходной канал с битовой скоростью, равной $N \cdot 64$ Кбит/с.

Порядок байт в уплотненном кадре соответствует номеру входного канала, от которого этот байт получен. Количество обслуживаемых мультиплексором абонентских каналов зависит от его быстродействия.

Например, мультиплексор **T1**, представляющий собой первый промышленный мультиплексор, работавший по технологии **TDM**, поддерживает 24 входных абонентских канала, создавая на выходе кадры стандарта **T1**, передаваемые с битовой скоростью 1,544 Мбит/с.

Демультиплексор выполняет обратную задачу – байты уплотненного кадра распределяются по своим выходным каналам, считая, что **порядковый номер** байта в кадре соответствует **номеру выходного канала**.

Коммутатор принимает уплотненный кадр по скоростному каналу от мультиплексора и записывает каждый байт из него в отдельную ячейку своей **буферной памяти**, причем в том порядке, в котором эти байты были упакованы в уплотненный кадр. Для выполнения операции коммутации байты извлекаются из буферной памяти не в порядке поступления, а в таком порядке, который соответствует поддерживаемым в сети соединениям абонентов. «Перемешивая» нужным образом байты в уплотненном кадре, коммутатор обеспечивает соединение конечных абонентов в сети.

Выделенный номер тайм-слота остается в распоряжении соединения.

5.4. ТЕХНОЛОГИЯ ПЛОТНОГО ВОЛНОВОГО МУЛЬТИПЛЕКСИРОВАНИЯ

Технология плотного волнового (спектрального) мультиплексирования **DWDM** (*Dense Wave Division Multiplexing*) предназначена для создания оптических магистралей сверхвысокой производительности. Сети **DWDM** работают по принципу коммутации каналов. Особенностью данной технологии является передача информации одновременно большим числом световых волн, каждая из которых представляет собой отдельный спектральный канал. Это достигается существенным уменьшением расстояния между длинами волн и использованием сигналов с минимально возможной шириной спектра несущей волны (расстояние между частотами соседних волн должно быть больше, чем спектр передаваемого сигнала). При этом на одной длине волны достигаются скорости передачи данных до 10 Гбит/с и более.

На сегодня определен ряд рекомендаций:

- разнесение частот между каналами 100 ГГц ($\Delta\lambda \approx 0,8$ нм), что обеспечивает применение 41 волны в диапазоне от 1528,77 нм (196,1 ТГц) до 1560,61 нм (192,1 ТГц);
- разнесение частот 50 ГГц ($\Delta\lambda \approx 0,4$ нм) – 81 волна.

Экспериментальное оборудование работает с разнесением частот между каналами в 25 ГГц.

5.5. МНОЖЕСТВЕННЫЙ ДОСТУП С КОДОВЫМ РАЗДЕЛЕНИЕМ КАНАЛОВ

Множественный доступ с кодовым разделением каналов **CDMA** (*Code Division Multiple Access*) основан на форме связи с расширенным спектром сигнала. Более подробно о технологиях уширения спектра говорим в главе 11.

18 В **CDMA** каждый битовый интервал разбивается на m элементарных сигналов или чипов (*chip*). На практике $m = 64$ или 128. Каждой станции

VC является единицей коммутации мультиплексора. *VC*: каждый мультиплексор содержит таблицу *кросс-соединений*, в которой для каждого *VC* указаны порты для сквозного пути к точкам присоединения оборудования пользователей. Схема мультиплексирования приведена на рис. 5.6.

Отметим особую важность использования концепции *указателей*. Указатель задает положение виртуального контейнера в структуре более высокого уровня – трибутарном блоке *TU* (*Tributary Unit*) или административном блоке *AU* (*Administrative Unit*). Применение указателя позволяет *VC* плавать внутри *TU* или *AU*.

Блоки *TU* объединяются в группы *TUG* (*TU Groups*), которые затем объединяются в административные группы *AUG* (*AU Groups*). Группа из *N* административных блоков составляет полезную нагрузку кадра *STM-N*.

Основное отличие блоков *TU* и *AU* от *VC* только в том, что на каждом шаге мультиплексирования добавляется *служебная информация*, задающая структуру блока и поле *указателя*, которое позволяет определить начало пользовательских данных.

5.7. КОММУТАЦИЯ ПАКЕТОВ

Коммутация пакетов – это техника коммутации абонентов, которая была специально разработана для эффективной передачи компьютерного трафика, который обычно имеет *пульсирующий* характер. Передача такого трафика на основе техники коммутации каналов *не позволяет достичь высокой общей пропускной способности сети*.

Передаваемые пользователем сообщения разбиваются в исходном узле на *небольшие* части, называемые *пакетами*.

Сообщение – логически завершенная группа данных, которая может иметь произвольную длину.

Каждый пакет снабжается заголовком, содержащим адрес узла назначения и некоторую нумерацию пакета, необходимую для корректной сборки сообщения. Пакеты транспортируются в сети как *независимые информационные блоки*.

Коммутаторы пакетной сети, в отличие от коммутаторов каналов, имеют внутреннюю *буферную память*, которая может быть использована для формирования *очередей пакетов* к выходным портам, которые в момент принятия пакета заняты передачей других пакетов.

Наличие буферной памяти позволяет *сглаживать пульсации* трафика на каналах связи между коммутаторами и тем самым использовать их

наиболее эффективным образом для повышения пропускной способности сети в целом.

Очевидно, что сеть с коммутацией пакетов может замедлять процесс взаимодействия конкретной пары абонентов из-за ожидания пакетов в коммутаторах. Однако общий объем передаваемых сетью компьютерных данных в единицу времени при технике коммутации пакетов будет выше, чем при технике коммутации каналов. Доказано, что пульсации отдельных абонентов в соответствии с законом больших чисел распределяются во времени, поэтому при большом числе абонентов коммутаторы постоянно и достаточно равномерно загружены работой.

Режим передачи пакетов между двумя конечными узлами сети, реализующий независимую маршрутизацию каждого пакета, называется **дейтаграммным**. В таких сетях каждый коммутатор выбирает маршрут в зависимости от состояния сети – работоспособности каналов и других коммутаторов, длины очередей пакетов и т. п.

При коммутации пакетов используется также режим передачи пакетов по предварительно построенному **виртуальному каналу VC** (*virtual circuit* или *virtual channel*). **VC** может быть динамическим или постоянным.

Динамический **VC** строится с помощью передачи специального пакета – **запроса на соединение**, при прохождении которого коммутаторы запоминают маршрут для данного соединения (присваивают **VC** специальную метку – номер виртуального канала).

Постоянные **VC** создаются администраторами сети путем ручной настройки коммутаторов. **VC** представляет собой **единственный** маршрут, по которому затем передаются данные. В случае отказа коммутатора или канала на пути **VC**, соединение разрывается, и **VC** прокладывается заново.

Отметим преимущества и недостатки обоих режимов передачи пакетов:

- дейтаграммный метод работает без задержки перед передачей данных, поскольку не требуется предварительного установления соединения, и быстрее адаптируется к изменениям в сети;
- при использовании метода виртуальных каналов время, затраченное на установление виртуального канала, компенсируется последующей быстрой передачей всего потока пакетов. Кроме того, распознавание принадлежности пакета к **VC** по номеру виртуального канала значительно проще и не требует анализа адресов конечных узлов, как это делается при дейтаграммном методе.

18/06/2012

5.8. КОММУТАЦИЯ СООБЩЕНИЙ

Под *коммутацией сообщений* понимается метод передачи сообщений в сетях с коммутацией пакетов или коммутацией каналов, при которой транзитные компьютеры сети обеспечивают временную их буферизацию в своей дисковой памяти. Такой способ коммутации абонентов с промежуточным хранением на диске называется режимом «хранение-и-передача» (*store-and-forward*) и применяется для передачи сообщений, не требующих немедленного ответа. Чаще всего этот способ используется при передаче сообщений электронной почты, новостей.

Применение режима коммутации сообщений, в основном для не оперативных служб, позволяет разгрузить сеть для передачи главного интерактивного трафика. Обычно в пакетной сети для временного хранения сообщений используется небольшое число компьютеров.

Техника коммутации сообщений по критерию пропускной способности сети уступает технике коммутации пакетов. Буферизация сообщений, размер которых может быть достаточно большим, требует существенных затрат времени. Кроме того, транзитные компьютеры, которые выступают в роли коммутаторов, должны иметь соответствующие объемы дисковой памяти, что может оказать влияние на стоимость сети.

8. Основы Ethernet

Ethernet ([ˈiːθə.net] от англ. *ether* [ˈiːθə] — «эфир» и англ. *network* — «сеть, цепь») — семейство технологий пакетной передачи данных для компьютерных сетей.

Стандарты Ethernet определяют проводные соединения и электрические сигналы на физическом уровне, формат кадров и протоколы управления доступом к среде — на канальном уровне модели OSI. Ethernet в основном описывается стандартами IEEE группы 802.3. Ethernet стал самой распространённой технологией ЛВС в середине 1990-х годов, вытеснив такие устаревшие технологии, как ARCNET и Token ring.

Название «Ethernet» (буквально «эфирная сеть» или «среда сети») отражает первоначальный принцип работы этой технологии: всё, передаваемое одним узлом, одновременно принимается всеми остальными (то есть имеется некое сходство с радиовещанием). В настоящее время практически всегда подключение происходит через коммутаторы (switch), так что кадры, отправляемые одним узлом, доходят лишь до адресата (исключение составляют передачи на широковещательный адрес) — это повышает скорость работы и безопасность сети.

В стандарте первых версий (Ethernet v1.0 и Ethernet v2.0) указано, что в качестве передающей среды используется коаксиальный кабель, в дальнейшем появилась возможность использовать [витую пару](#) и [оптический кабель](#).

Преимущества использования витой пары по сравнению с коаксиальным кабелем:

- возможность работы в [дуплексном](#) режиме;
- низкая стоимость кабеля витой пары;
- более высокая надёжность сетей: при использовании витой пары сеть строится по топологии «звезда», поэтому обрыв кабеля приводит лишь к нарушению связи между двумя объектами сети, соединёнными этим кабелем (при использовании коаксиального кабеля сеть строится по топологии «общая шина», для которой требуется наличие терминальных резисторов на концах кабеля, поэтому обрыв кабеля приводит к неисправности сегмента сети);
- уменьшен минимально допустимый радиус изгиба кабеля;
- большая помехоустойчивость из-за использования дифференциального сигнала;
- возможность питания по кабелю маломощных узлов, например, IP-телефонов (стандарт [Power over Ethernet](#), PoE);
- **гальваническая развязка трансформаторного типа.** В условиях СНГ, где, как правило, отсутствует заземление компьютеров, применение коаксиального кабеля часто приводило к выходу из строя сетевых карт в результате электрического пробоя.

Причиной перехода на оптический кабель была необходимость увеличить длину сегмента без повторителей.

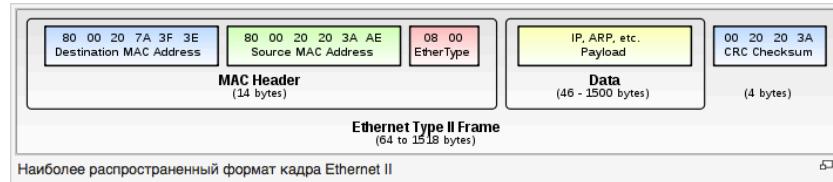
Метод управления доступом (для сети на коаксиальном кабеле) — **множественный доступ с контролем несущей и обнаружением коллизий** (CSMA/CD, Carrier Sense Multiple Access with Collision Detection), скорость передачи данных 10 Мбит/с, размер кадра от 64 до 1518 байт, описаны методы кодирования данных. Режим работы полу duplexный, то есть узел не может одновременно передавать и принимать информацию. Количество узлов в одном разделении сегмента сети ограничено предельным значением в 1024 рабочих станций (спецификации физического уровня могут устанавливать более жёсткие ограничения, например, к сегменту тонкого коаксиала может подключаться не более 30 рабочих станций, а к сегменту толстого коаксиала — не более 100). Однако сеть, построенная на одном разделении сегмента, становится неэффективной задолго до достижения предельного значения количества узлов, в основном по причине полу duplexного режима работы.

В 1995 году принят стандарт [IEEE 802.3](#) **Fast Ethernet** со скоростью 100 Мбит/с и появилась возможность работы в режиме [полный дуплекс](#). В 1997 году был принят стандарт [IEEE 802.3z](#) **Gigabit Ethernet** со скоростью 1000 Мбит/с для передачи по [оптическому волокну](#) и ещё через два года для передачи по витой паре.

Формат кадра [править | править вики-текст]

Существует несколько форматов Ethernet-кадра.

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



Наиболее распространенный формат кадра Ethernet II

- Novell — внутренняя модификация IEEE 802.3 без LLC ([Logical Link Control](#)).
- Кадр IEEE 802.3 LLC.
- Кадр IEEE 802.3 LLC/SNAP.
- Некоторые сетевые карты Ethernet, производимые компанией [Hewlett-Packard](#), использовали при работе кадр формата IEEE 802.12, соответствующий стандарту [100VG-AnyLAN](#).

В качестве дополнения Ethernet-кадр может содержать тег [IEEE 802.1Q](#) для идентификации [VLAN](#), к которой он адресован, и [IEEE 802.1p](#) для указания приоритетности.

Разные типы кадра имеют различный формат и значение [MTU](#).

MAC-адреса [править | править вики-текст]

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

Уникальность MAC-адресов достигается тем, что каждый производитель получает в координирующем комитете [IEEE Registration Authority](#) диапазон из шестнадцати миллионов (2^{24}) адресов, и по мере исчерпания выделенных адресов может запросить новый диапазон. Поэтому по трём старшим байтам MAC-адреса можно определить производителя. Существуют таблицы, позволяющие определить производителя по MAC-адресу; в частности, они включены в программы типа [arpalert](#).

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

Некоторое время назад, когда драйверы сетевых карт не давали возможность изменить свой MAC-адрес, а альтернативные возможности не были слишком известны, некоторые провайдеры Internet использовали его для идентификации машины в сети при учёте трафика. Программы из Microsoft Office, начиная с версии Office 97, записывали MAC-адрес сетевой платы в редактируемый документ в качестве составляющей уникального GUID-идентификатора.^[4]

5 Разновидности Ethernet

- 5.1 Ранние модификации Ethernet
- 5.2 10 Мбит/с Ethernet
- 5.3 Быстрый Ethernet (Fast Ethernet, 100 Мбит/с)
- 5.4 Гигабитный Ethernet (Gigabit Ethernet, 1 Гбит/с)
- 5.5 2,5- и 5-гигабитные варианты (NBASE-T, MGBASE-T)
- 5.6 10-гигабитный Ethernet (10G Ethernet, 10 Гбит/с)
- 5.7 40-гигабитный и 100-гигабитный Ethernet

9. Logical Link Control protocol

Logical Link Control (общепринятое сокращение — LLC) — подуровень управления логической связью — по стандарту IEEE 802 — верхний подуровень канального уровня модели OSI, осуществляет:

- управление передачей данных;
- обеспечивает проверку и правильность передачи информации по соединению.

Структура кадра [править | править вики-текст]

По своему назначению все кадры уровня LLC (называемые в стандарте IEEE 802.2 блоками данных — Protocol Data Unit, PDU) подразделяются на три типа — информационные, управляющие и ненумерованные:

- Информационные кадры предназначены для передачи информации в процедурах с установлением логического соединения и должны обязательно содержать поле информации. В процессе передачи информационных блоков осуществляется их нумерация в режиме скользящего окна.
- Управляющие кадры предназначены для передачи команд и ответов в процедурах с установлением логического соединения, в том числе запросов на повторную передачу искаженных информационных блоков.
- Ненумерованные кадры предназначены для передачи ненумерованных команд и ответов, выполняющих в процедурах без установления логического соединения передачу информации, идентификацию и тестирование LLC-уровня, а в процедурах с установлением логического соединения — установление и разъединение логического соединения, а также информирование об ошибках.

3 Сетевой уровень
2 Канальный уровень
Подуровень управления логической связью (LLC)
Подуровень управления доступом к среде (MAC)
1 Физический уровень
Шаблон: Просмотр • Обсуждение • Правка

Все типы кадров уровня LLC имеют единый формат. Они содержат четыре поля:

- адрес точки входа сервиса назначения (Destination Service Access Point, DSAP),
- адрес точки входа сервиса источника (Source Service Access Point, SSAP),
- управляющее поле (Control)
- поле данных (Data)

Флаг	DSAP	SSAP	Control	Data	Флаг
01111110	Адрес точки входа сервиса назначения	Адрес точки входа сервиса источника	Управляющее поле	Данные	01111110

Кадр LLC обрамляется двумя однобайтовыми полями «Флаг», имеющими значение 01111110. Флаги используются на MAC-уровне для определения границ блока. (Отметим, что формат кадров LLC, за исключением поля адреса точки входа сервиса источника, соответствует формату кадра HDLC, а также одного из вариантов протокола HDLC — протокола LAP-B, используемого в сетях X.25).

Поле данных кадра LLC предназначено для передачи по сети пакетов протоколов верхних уровней — IP, IPX, AppleTalk, DECnet, в редких случаях — прикладных протоколов, когда те не пользуются сетевыми протоколами, а вкладывают свои сообщения непосредственно в кадры канального уровня. Поле данных может отсутствовать в управляющих кадрах и некоторых ненумерованных кадрах.

Поле управления (один байт) используется для обозначения типа кадра данных — информационный, управляющий или ненумерованный. Кроме этого, в этом поле указываются порядковые номера отправленных и успешно принятых кадров, если подуровень LLC работает по процедуре LLC2 с установлением соединения. Формат поля управления полностью совпадает с форматом поля управления кадра LAP-B.

Поля DSAP и SSAP позволяют указать, какой сервис верхнего уровня пересыпает данные с помощью этого кадра. Программному обеспечению узлов сети при получении кадров канального уровня необходимо распознать, какой протокол вложил свой пакет в поле данных поступившего кадра, для того, чтобы передать извлеченный из кадра пакет нужному протоколу для последующей обработки. Например, в качестве значения DSAP и SSAP может выступать код протокола IPX или же код протокола покрывающего дерева Spanning Tree.

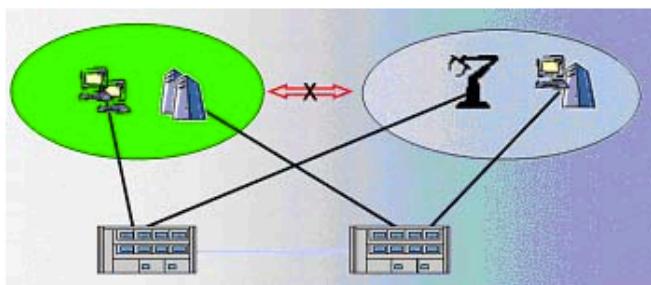
10. Virtual LAN

VLAN (Virtual LAN)

Virtual LAN – объединение портов коммутаторов и конечного оборудования в одно виртуальное информационное пространство (сеть второго уровня). Физически, устройства объединенные в VLAN могут находиться на достаточном удалении друг от друга (различные этажи зданий, различные здания). Объединение их в одну виртуальную сеть предоставляет возможность более гибкого и удобного управления, исчезает привязка оборудования и рабочих станций к конкретному месту. Исключается передача широковещательных сообщений между сетями, что снижает нагрузку на всю сеть.

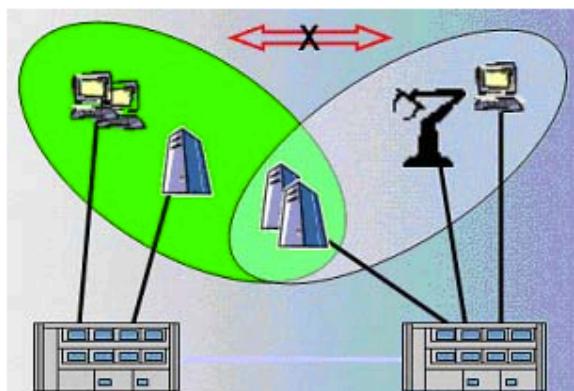
Виртуальные сети второго уровня могут строится по двум принципам:

Раздельные VLAN – при построении сети по данному принципу исключается любое взаимодействие устройств, принадлежащих разным виртуальным сетям.



Раздельные VLAN

Пересекающиеся VLAN – при построение сети по данному принципу между двумя виртуальными сегментами может быть организован общий ресурс (например файловый сервер или сервер БД) к которому могут обращаться устройства тех сегментов, в которые он входит. При этом сохраняется исключение передачи широковещательного трафика между сегментами VLAN.



Пересекающиеся VLAN

Построение виртуальных сетей возможно на оборудовании семейства Rail, MICE, MACH 3000.

Причем при использовании магистральных коммутаторов MACH 3000 возможно построение виртуальных сетей по MAC адресам. При этом отпадает какая-либо привязка оборудования к конкретному порту коммутатора, что дает возможность свободного перемещения оборудования без внесения изменений в настройки конфигурации.

11.FDDI

FDDI (англ. Fiber Distributed Data Interface – Волоконно-оптический интерфейс передачи данных) – стандарт передачи данных в [локальной сети](#), протянутой на расстоянии до 200 километров. Стандарт основан на [протоколе Token Ring](#). Кроме большой территории, сеть FDDI способна поддерживать несколько тысяч пользователей.

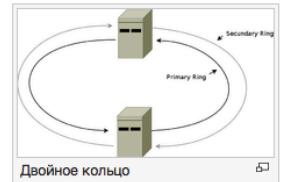
Основы технологии [править | править вики-текст]

Стандарт FDDI определяет 100 Мб/сек. LAN с двойным кольцом и передачей маркера, которая использует в качестве среды передачи волоконно-оптический кабель. Он определяет физический уровень и часть канального уровня, которая отвечает за доступ к носителю; поэтому его взаимоотношения с эталонной моделью OSI примерно аналогичны тем, которые характеризуют IEEE 802.3 и IEEE 802.5.

Хотя она работает на более высоких скоростях, FDDI во многом похожа на Token Ring. Обе сети имеют одинаковые характеристики, включая топологию (кольцевая сеть), технику доступа к носителю (передача маркера), характеристики надежности (например, сигнализация-beaconing), и др.

Одной из наиболее важных характеристик FDDI является то, что она использует световод в качестве передающей среды. Световод обеспечивает ряд преимуществ по сравнению с традиционной медной проводкой, включая защиту данных (оптоволокно не излучает электрические сигналы, которые можно перехватывать), надежность (оптоволокно устойчиво к электрическим помехам) и скорость (потенциальная пропускная способность световода намного выше, чем у медного кабеля).

При обрывах оптоволокна возможно частичное (при двух обрывах) или полное (при одном обрыве) восстановление связности сети.



Физические соединения [править | править вики-текст]

FDDI устанавливает применение двойных кольцевых сетей. Трафик по этим кольцам движется в противоположных направлениях. В физическом выражении кольцо состоит из двух или более двухточечных соединений между смежными станциями. Одно из двух колец FDDI называется первичным кольцом, другое – вторичным кольцом. Первичное кольцо используется для передачи данных, в то время как вторичное кольцо обычно является дублирующим.

"Станции Класса В" или "станции, подключаемые к одному кольцу" (SAS) подсоединены к одной кольцевой сети; "станции класса А" или "станции, подключаемые к двум кольцам" (DAS) подсоединены к обеим кольцевым сетям. SAS подключены к первичному кольцу через "концентратор", который обеспечивает связи для множества SAS. Концентратор отвечает за то, чтобы отказ или отключение питания в любой из SAS не прерывали кольцо. Это особенно необходимо, когда к кольцу подключен PC или аналогичные устройства, у которых питание часто включается и выключается.

Типы трафика [править | править вики-текст]

FDDI поддерживает распределение полосы пропускания сети в масштабе реального времени, что является идеальным для ряда различных типов прикладных задач. FDDI обеспечивает эту поддержку путем обозначения двух типов трафика: синхронного и асинхронного. Синхронный трафик может потреблять часть общей полосы пропускания сети FDDI, равную 100 Mb/сек; оставшую часть может потреблять асинхронный трафик. Синхронная полоса пропускания выделяется тем станциям, которым необходима постоянная возможность передачи. Например, наличие такой возможности помогает при передаче голоса и видеинформации. Другие станции используют оставшую часть полосы пропускания асинхронно. Спецификация SMT для сети FDDI определяет схему распределенных заявок на выделение полосы пропускания FDDI.



Распределение асинхронной полосы пропускания производится с использованием восьмиуровневой схемы приоритетов. Каждой станции присваивается определенный уровень приоритета пользования асинхронной полосой пропускания. FDDI также разрешает длительные диалоги, когда станции могут временно использовать всю асинхронную полосу пропускания. Механизм приоритетов FDDI может фактически блокировать станции, которые не могут пользоваться синхронной полосой пропускания и имеют слишком низкий приоритет пользования асинхронной полосой пропускания.

12.Беспроводные сети

Существует два основных направления применения беспроводных компьютерных сетей:

- Работа в замкнутом объеме (офис, выставочный зал и т. п.);
- Соединение удаленных [локальных сетей](#) (или удаленных сегментов локальной сети).

Для организации беспроводной сети в замкнутом пространстве применяются передатчики со всенаправленными антеннами. Стандарт [IEEE 802.11](#) определяет два режима работы сети — [Ad-hoc](#) и [клиент-сервер](#). Режим Ad-hoc (иначе называемый «[точка-точка](#)») — это простая сеть, в которой связь между станциями (клиентами) устанавливается напрямую, без использования специальной [точки доступа](#). В режиме клиент-сервер беспроводная сеть состоит, как минимум, из одной точки доступа, подключенной к проводной сети, и некоторого набора беспроводных клиентских станций. Поскольку в большинстве сетей необходимо обеспечить доступ к файловым серверам, принтерам и другим устройствам, подключенным к проводной локальной сети, чаще всего используется режим клиент-сервер. Без подключения дополнительной антенны устойчивая связь для оборудования IEEE 802.11b достигается в среднем на следующих расстояниях: открытое пространство — 500 м, комната, разделенная перегородками из неметаллического материала — 100 м, офис из нескольких комнат — 30 м. Следует иметь в виду, что через стены с большим содержанием металлической арматуры (в железобетонных зданиях таковыми являются

несущие стены) радиоволны диапазона 2,4 ГГц иногда могут вообще не проходить, поэтому в комнатах, разделенных подобной стеной, придется ставить свои точки доступа.

Для соединения удаленных локальных сетей (или удаленных сегментов локальной сети) используется оборудование с направленными [антеннами](#), что позволяет увеличить дальность связи до 20 км (а при использовании специальных усилителей и большой высоте размещения антенн — до 50 км). Причем в качестве подобного оборудования могут выступать и устройства [Wi-Fi](#), нужно лишь добавить к ним специальные антенны (конечно, если это допускается конструкцией). Комплексы для объединения локальных сетей по топологии делятся на «точку-точку» и «[звезду](#)». При топологии «точка-точка» (режим Ad-hoc в IEEE 802.11) организуется радиомост между двумя удаленными сегментами сети. При топологии «звезда» одна из станций является центральной и взаимодействует с другими удаленными станциями. При этом центральная станция имеет всенаправленную антенну, а другие удаленные станции — односторонние антенны. Применение всенаправленной антенны в центральной станции ограничивает дальность связи дистанцией примерно 7 км. Поэтому, если требуется соединить между собой сегменты локальной сети, удаленные друг от друга на расстояние более 7 км, приходится соединять их по принципу «точка-точка». При этом организуется беспроводная сеть с кольцевой или иной, более сложной топологией.

Мощность, излучаемая передатчиком точки доступа или же клиентской станции, работающей по стандарту IEEE 802.11, не превышает 0,1 Вт, но многие производители беспроводных точек доступа ограничивают мощность лишь программным путём, и достаточно просто поднять мощность до 0,2-0,5 Вт. Для сравнения — мощность, излучаемая [мобильным телефоном](#), на порядок больше(в момент звонка - до 2 Вт). Поскольку, в отличие от мобильного телефона, элементы сети расположены далеко от головы, в целом можно считать, что беспроводные компьютерные сети более безопасны с точки зрения здоровья, чем мобильные телефоны.

Если беспроводная сеть используется для объединения сегментов локальной сети, удаленных на большие расстояния, антенны, как правило, размещаются за пределами помещения и на большой высоте.

13.Структуризация сети: мосты и коммутаторы и их виды

Сетевой мост, бридж (с [англ. bridge](#)) — сетевое устройство [второго уровня модели OSI](#), предназначенное для объединения [сегментов \(подсети\) компьютерной сети](#) в единую сеть.

Термин «прозрачные» мосты объединяет большую группу устройств, поэтому их принято группировать в категории, базирующиеся на различных характеристиках изделий:

- Прозрачные мосты ([англ. transparent bridges](#)) объединяют сети с едиными протоколами канального и физического уровней модели OSI;
- Транслирующие мосты ([англ. translating bridges](#)) объединяют сети с различными протоколами канального и физического уровней;
- [Инкапсулирующие](#) мосты ([англ. encapsulating bridges](#)) соединяют сети с едиными протоколами канального и физического уровня через сети с другими протоколами.

Мост обеспечивает:

- ограничение [домена коллизий](#)
- задержку фреймов, адресованных узлу в сегменте отправителя
- ограничение перехода из домена в домен ошибочных фреймов:
 - [карликов](#) (фреймов меньшей длины, чем допускается по стандарту (64 байта))
 - фреймов с ошибками в [CRC](#)
 - фреймов с признаком «коллизия»
 - [затянувшихся фреймов](#) (размером больше, чем разрешено стандартом)

Мосты «изучают» характер расположения сегментов сети путём построения адресных таблиц вида «Интерфейс:[MAC-адрес](#)», в которых содержатся адреса всех сетевых устройств и сегментов, необходимых для получения доступа к данному устройству.

Мосты увеличивают [латентность](#) сети на 10-30 %. Это увеличение латентности связано с тем, что мосту при передаче данных требуется дополнительное время на принятие решения.

Мост рассматривается как устройство с функциями хранения и дальнейшей отправки, поскольку он должен проанализировать поле адреса пункта назначения фрейма и вычислить контрольную сумму [CRC](#) в поле контрольной последовательности фрейма перед отправкой фрейма на все порты.

Если порт пункта назначения в данный момент занят, то мост может временно сохранить фрейм до освобождения порта.

Для выполнения этих операций требуется некоторое время, что замедляет процесс передачи и увеличивает латентность.

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

Spanning Tree Protocol (STP, протокол оставного дерева) — канальный протокол. Основной задачей STP является устранение петель в топологии произвольной сети [Ethernet](#), в которой есть один или более [сетевых мостов](#), связанных избыточными соединениями. STP решает эту задачу, автоматически блокируя соединения, которые в данный момент для полной связности коммутаторов являются избыточными.

Необходимость устранения топологических петель в сети Ethernet следует из того, что их наличие в реальной сети Ethernet с [коммутатором](#) с высокой вероятностью приводит к бесконечным повторам передачи одних и тех же кадров Ethernet одним и более коммутатором, отчего пропускная способность сети оказывается почти полностью занятой этими бесполезными повторами; в этих условиях, хотя формально сеть может продолжать работать, на практике её производительность становится настолько низкой, что может выглядеть как полный отказ сети.

Суть работы протокола заключается в том, что поддерживающие его коммутаторы сети Ethernet обмениваются друг с другом информацией «о себе». На основании определённых условий (обычно в соответствии с настройками) один из коммутаторов выбирается «корневым» (или «root»), после чего все остальные коммутаторы по алгоритму оставного дерева выбирают для работы порты, «ближайшие» к «корневому» коммутатору (учитывается количество посредников и скорость линий). Все прочие сетевые порты, ведущие к «корневому» коммутатору, блокируются. Таким образом образуется несвязное дерево с корнем в выбранном коммутаторе.

- Выбирается один корневой мост ([англ. Root Bridge](#)).
- Далее каждый коммутатор, отличный от корневого, просчитывает кратчайший путь к корневому. Соответствующий порт называется корневым портом ([англ. Root Port](#)). У любого некорневого коммутатора может быть только один корневой порт.

3. После этого для каждого сегмента сети, к которому присоединён более чем один мост (или несколько портов одного моста), просчитывается кратчайший путь к корневому порту. Мост, через который проходит этот путь, становится **назначенным** для этой сети ([англ. Designated Bridge](#)), а соответствующий порт — **назначенным портом** ([англ. Designated port](#)).
4. Далее во всех сегментах, с которыми соединены более одного порта моста, все мосты блокируют все порты, не являющиеся корневыми и назначенными. В итоге получается древовидная структура (математический [граф](#)) с вершиной в виде корневого коммутатора.

После включения коммутаторов в сеть, по умолчанию каждый коммутатор считает себя корневым (root).

Каждый коммутатор начинает посыпать по всем портам конфигурационные Hello [BPDU](#) пакеты раз в 2 секунды.

Если мост получает [BPDU](#) с идентификатором моста (Bridge ID) меньшим, чем свой собственный, он прекращает генерировать свои BPDU и начинает ретранслировать BPDU с этим идентификатором. Таким образом в конце концов в этой сети Ethernet остаётся только один мост, который продолжает генерировать и передавать собственные BPDU. Он и становится *корневым мостом* (root bridge).

Остальные мосты ретранслируют BPDU корневого моста, добавляя в них собственный идентификатор и увеличивая счётчик стоимости пути (path cost).

Для каждого сегмента сети, к которому присоединены два и более портов мостов, происходит определение designated port — порта, через который BPDU, приходящие от корневого моста, попадают в этот сегмент.

После этого все порты в сегментах, к которым присоединены 2 и более портов моста, блокируются за исключением root port и designated port.

Корневой мост продолжает посылать свои Hello BPDU раз в 2 секунды.

Скорость передачи Стоимость (802.1D-1998) Стоимость (802.1W-2001)

4 Мбит/с	250	5 000 000
10 Мбит/с	100	2 000 000
16 Мбит/с	62	1 250 000
100 Мбит/с	19	200 000
1 Гбит/с	4	20 000
2 Гбит/с	3	10 000
10 Гбит/с	2	2 000

Коммутаторы используют и другие схемы построения коммутационных матриц, в частности на основе высокоскоростной шины или многовходовой разделяемой памяти.

Существует два подхода к коммутации – с полной буферизацией кадра (*store and forward*) или коммутация на «лету» (*on-the-fly cut-through*).

1. Технология с полной буферизацией предполагает, что *EPP* полностью буферизирует пришедший на порт кадр. Затем *EPP* анализирует его заголовок. Адрес отправителя (*SA*) используется для построения общей *AT* коммутатора. Кроме того, для повышения производительности каждый *EPP* имеет собственный кэш *AT*, содержащий текущие записи, с которыми он работает.

По адресу назначения (*DA*) *EPP* определяет выходной порт, в который кадр должен быть передан. *EPP* просматривает сначала собственный кэш *AT*, а если в нем не находит требуемого адреса – обращается к системному модулю, работающему в многозадачном режиме.

18/06/20

вающему все **EPP**. Системный модуль возвращает требуемую строку, которая буферизируется в кэше.

Если требуется отфильтровать кадр, то **EPP** очищает буфер и ожидает поступления следующего кадра.

Если необходимо передать кадр на другой порт, то **EPP** обращается к коммутационной матрице для построения канала к порту назначения. Это возможно, если порт назначения в данный момент свободен, иначе **EPP** ожидает его освобождения.

После того как канал построен, в него направляются буферизованные байты кадра, которые принимаются **EPP** выходного порта. Как только **EPP** выходного порта получает доступ к подключенному к нему сегменту *Ethernet* по протоколу *CSMA/CD*, байты кадра сразу же начинают передаваться в сеть.

В случае широковещательной или групповой передачи, а также при отсутствии необходимых записей в *AT*, производится передача во все остальные порты, по мере их освобождения.

Эта технология позволяет анализировать кадр (проверять *CRC*-код) и игнорировать ошибочные кадры. Недостатком ее является значительная задержка передачи кадров.

2. *Коммутация «на лету»* представляет конвейерную обработку кадра, совмещенную во времени несколько этапов его передачи, по возможности без полной буферизации. После принятия первых 6 байт (а это *DA*) **EPP** может уже начать пересылку кадра в выходной порт, если коммутационная матрица разрешает, при этом продолжая прием оставшихся байт кадра. Конечно, буферизация неизбежна, если выходной порт занят.

Коммутация «на лету» вносит минимальные задержки при передаче кадра, однако проверка *CRC* невозможна и передаются все кадры, в том числе и испорченные коллизией.

Отсюда следует, что основным методом повышения производительности при использовании коммутаторов является *параллельная обработка кадров*. В идеальном случае коммутатор с N портами обеспечивает $N/2$ независимых путей. Общая производительность коммутатора составляет $(N/2)*C$, где C – пропускная способность протокола доступа к среде. В этом случае говорят, что коммутатор предоставляет подключенным к его портам станциям или сегментам выделенную *пропускную способность протокола*.

В ЛС обычно используются неблокирующие (*non-blocking*) модели коммутаторов – коммутаторы, которые могут передавать кадры через свои порты с той же скоростью, с которой они на них поступают. Если же входной поток кадров (просуммированный по всем портам) в среднем

Наличие серьезных ограничений у протоколов канального уровня показывает, что построение на основе средств этого уровня больших неоднородных сетей является весьма проблематичным. Естественное решение в этих случаях – это привлечение средств более высокого, сетевого уровня.

Основная идея введения сетевого уровня состоит в следующем:

- сеть в общем случае рассматривается как совокупность нескольких сетей и называется *составной сетью* или *интерсетью* (*internetwork* или *internet*);
- сети, входящие в составную сеть, называются *подсетями* (*subnet*), составляющими сетями или просто сетями;
- подсети соединяются между собой *маршрутизаторами*.

Компонентами составной сети могут являться как *локальные*, так и *глобальные* сети. Все узлы в пределах одной подсети взаимодействуют, используя присущую им технологию, в том числе и собственную локальную адресацию.

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

Сетевой уровень необходим в качестве *координатора*, организующего работу всех подсетей по обеспечению транспортировки данных между любой произвольной парой узлов этой составной сети.

Для решения проблемы сетевому уровню необходима собственная *система адресации*, не зависящая от локальной в отдельных подсетях, которая позволила бы однозначно идентифицировать любой узел составной сети.

Сетевой адрес представляет собой структуру, которая включает:

- номер сети (подсети);
- номер узла.

В качестве номера узла может выступать либо локальный адрес этого узла (*такая принята в стеке IPX/SPX*), либо некоторое число, никак не связанное с локальной технологией, которое однозначно идентифицирует узел в пределах данной подсети (*схема стека TCP/IP*).

Данные, которые поступают на сетевой уровень для передачи через составную сеть, снабжаются *заголовком* сетевого уровня. Данные вместе с заголовком образуют *пакет*. Заголовок пакета сетевого уровня имеет унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, которые могут входить в объединенную сеть.

Сетевой уровень определяет маршрут перемещения пакета между подсетями. Явная нумерация сетей позволяет протоколам сетевого

уровня составлять точную карту межсетевых связей и выбирать рациональные маршруты при любой их топологии, в том числе альтернативные маршруты, если они имеются. Кроме номера сети, заголовок сетевого уровня может содержать и другую информацию, необходимую для успешного продвижения пакетов, например:

- *нумерацию фрагментов пакета*, необходимую для успешного проведения операций сборки-разборки пакетов при их транспортировке через сети с разными максимальными размерами пакетов;
- *время жизни пакета*, указывающее время нахождения пакета в интэрнете; это время может использоваться для уничтожения «заблудившихся» пакетов;
- *качество услуги* – критерий выбора маршрута при межсетевых передачах.

Маршрутизация ([англ. Routing](#)) — процесс определения маршрута следования информации в сетях связи.

Маршруты могут задаваться административно ([статические маршруты](#)), либо вычисляться с помощью [алгоритмов маршрутизации](#), базируясь на информации о [топологии](#) и состоянии сети, полученной с помощью [протоколов маршрутизации](#) (динамические маршруты).

Статическими маршрутами могут быть:

- маршруты, не изменяющиеся во времени;
- маршруты, изменяющиеся по расписанию;

Маршрутизация в компьютерных сетях выполняется специальными программно-аппаратными средствами — [маршрутизаторами](#); в простых конфигурациях может выполняться и компьютерами общего назначения, соответственно настроенными.

[Протокол маршрутизации](#) может работать только с пакетами, принадлежащими к одному из маршрутизуемых протоколов, например, [IP](#), [IPX](#) или [Xerox Network System](#), [AppleTalk](#).

Маршрутизуемые протоколы определяют формат пакетов (заголовков), важнейшей информацией из которых для маршрутизации является адрес назначения. Протоколы, не поддерживающие маршрутизацию, могут передаваться между сетями с помощью [туннелей](#). Подобные возможности обычно предоставляют программные маршрутизаторы и некоторые модели аппаратных маршрутизаторов.

16. Модель DOD. Классовая и бесклассовая адресация

Модель DOD (в отличие от семиуровневой модели [OSI](#)) состоит из следующих четырёх уровней (сверху вниз):

- уровень приложений или прикладной уровень ([англ. process/application](#); соответствует трём верхним уровням модели OSI ([прикладному уровню](#), [уровню представления](#) и [сеансовому уровню](#)));
- транспортный уровень ([англ. transport](#); соответствует [транспортному уровню](#) модели OSI);
- межсетевой уровень ([англ. internet](#); соответствует [сетевому уровню](#) модели OSI);
- уровень сетевого доступа ([англ. network access](#); соответствует двум нижним уровням модели OSI ([физическому уровню](#) и [канальному уровню](#))).

Каждый из четырёх уровней модели DOD выполняет функции соответствующих ему уровней модели [OSI](#).

Прикладной уровень

Прикладной уровень модели DOD включает [протоколы](#):

- обрабатывающие данные пользователей:
 - [Telnet \(удалённый доступ\)](#);
 - [FTP \(передача файлов\)](#);
 - [SMTP \(передача электронной почты\)](#);
- управляющие передачей данных между приложениями:
 - [SNMP \(управление сетевыми устройствами\)](#);
 - [BOOTP \(передача клиентам настройки сети\)](#);
 - [RARP \(получение MAC-адреса по IP-адресу\)](#);
 - [DNS \(получение IP-адресов по доменным именам\)](#) и наоборот).

На этом уровне стандартизируется представление данных.

Транспортный уровень

Транспортный уровень модели DOD содержит протоколы, ответственные за контроль [целостности передаваемых данных](#), установку и прекращение соединений:

- [TCP \(гарантированная доставка](#) (англ.); [установка и прекращение соединения](#) (англ.));
- UDP (доставка не гарантируется; [соединение не устанавливается](#) (англ.)).

[TCP](#) считается надёжным, так как получатель (приёмник) отправляет пакет — подтверждение всякий раз, как получает данные, размер которых равен заранее выбранному числу, называемому «[размером окна](#)». При потере пакетов отправитель (источник) отправит данные повторно. TCP — протокол с установкой соединения.

[UDP](#) — протокол без установки соединения и без механизмов, обеспечивающих гарантированную доставку. За счёт отсутствия дополнительных возможностей UDP работает быстрее TCP.

Межсетевой уровень

Межсетевой уровень модели DOD содержит протоколы, предназначенные для маршрутизации передаваемых данных:

- [IP \(доставка данных отправителя получателю\)](#);
- [ICMP \(диагностика, информирование о ошибках\)](#);
- [IGMP \(multicast\)](#).

Все протоколы транспортного уровня используют протокол [IP](#). IP — протокол, обеспечивающий адресацию в сети и связанные с ней функции без установки соединения и без механизмов контроля целостности. Протокол IP:

- определяет факт получения повреждённого пакета;
- определяет факт получения копии пакета;
- делит большие пакеты на фрагменты, собирает фрагменты в правильном порядке, определяет факт отсутствия утерянных фрагментов.

Гарантированную доставку данных могут обеспечить протоколы вышестоящих уровней.

Протокол [ICMP](#) ([англ. internet control message protocol](#)) — протокол для передачи сообщений об ошибках и диагностики, работающий поверх IP (использующий IP для доставки данных) и считающийся неотъемлемой частью протокола IP.

[IGMP](#) (lang-enlinternet group management protocol}) — протокол, используемый для объединения устройств в группы и обеспечивающий одновременную передачу данных всем устройствам внутри группы ([multicast](#)).

Уровень сетевого доступа

Уровень сетевого доступа содержит протоколы, предназначенные для физической передачи данных между устройствами сети. На этом уровне данные размещаются в кадре. Для различных типов сетей существуют различные протоколы этого уровня.

Изначально адресация в сетях [IP](#) осуществлялась на основе классов: первые биты определяли класс сети, а по классу сети можно было сказать — сколько бит было отведено под номер сети и номер узла. Всего существовало 5 классов:

Класс А	0	адрес сети (7 бит)	адрес хоста (24 бита)
Класс В	10	адрес сети (14 бит)	адрес хоста (16 бит)
Класс С	110	адрес сети (21 бит)	адрес хоста (8 бит)
Класс D	1110	Адрес многоадресной рассылки	
Класс Е	1111 ^[2]	Зарезервировано	

Адресация IP

Особенностью IP является гибкая система адресации. Плата за это — наличие централизованных служб типа DNS.

Адрес состоит из двух частей — номер сети и номер узла в сети. IP-адрес версии 4 имеет длину 4 байта, записывается в виде четырех десятичных чисел, разделенных точками.

Для определения, какие байты принадлежат номеру сети, а какие номеру узла существует несколько подходов.

Одним из подходов был классовый метод адресации.

Класс	Первые биты	Распределение байт (С — сеть, X — хост)	Число возможных адресов сетей	Число возможных адресов хостов	Маска подсети	Стартовый адрес	Конечный адрес
A	0	C.X.X.X	128	16 777 216	255.0.0.0	0.0.0.0	127.255.255.255
B	10	C.C.X.X	16 384	65 536	255.255.0.0	128.0.0.0	191.255.255.255
C	110	C.C.C.X	2 097 152	256	255.255.255.0	192.0.0.0	223.255.255.255
D	1110		Групповой адрес			224.0.0.0	239.255.255.255
E	1111		Зарезервировано			240.0.0.0	255.255.255.255

Нетрудно посчитать, что всего в пространстве адресов IP — 128 сетей по 16 777 216 адресов класса A, 16384 сети по 65536 адресов класса B и 2 097 152 сети по 256 адресов класса C, а также 268 435 456 адресов многоадресной рассылки и 268 435 456 зарезервированных адресов. С ростом сети Интернет эта система оказалась неэффективной и была дополнена [бесклассовой адресацией](#) (CIDR).

Бесклассовая адресация ([англ. Classless Inter-Domain Routing](#), [англ. CIDR](#)) — метод IP-адресации, позволяющий гибко управлять пространством IP-адресов, не используя жёсткие рамки [классовой адресации](#). Использование этого метода позволяет экономно использовать ограниченный ресурс IP-адресов, поскольку возможно применение различных [масок подсетей](#) к различным подсетям.

IP-адрес является массивом бит. Принцип IP-адресации — выделение множества (диапазона, блока, подсети) IP-адресов, в котором некоторые битовые разряды имеют фиксированные значения, а остальные разряды пробегают все возможные значения. Блок адресов задаётся указанием начального адреса и маски подсети. Бесклассовая адресация основывается на переменной длине маски подсети ([англ. variable length subnet mask](#), *VLSM*), в то время, как в классовой (традиционной) адресации длина маски строго фиксирована 0, 1, 2 или 3 установленными [октетами](#).

Пример подсети 192.0.2.32/27 с применением бесклассовой адресации:

Октеты IP-адреса	192	0	2	32
Биты IP-адреса	1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0			
Биты маски подсети	1 0 0 0 0			
Октеты маски подсети	255	255	255	224

В данном примере видно, что в маске подсети 27 бит слева выставлены в единицу. В таком случае говорят о длине префикса подсети в 27 бит и указывают через косую черту (знак /) после базового адреса.

17.Протоколы IP(v4,v6) и ARP

IP объединяет сегменты сети в единую сеть, обеспечивая доставку пакетов данных между любыми узлами сети через произвольное число промежуточных узлов ([маршрутизаторов](#)). Он классифицируется как протокол третьего уровня по [сетевой модели OSI](#). IP не гарантирует надёжной доставки пакета до адресата — в частности, пакеты могут прийти не в том порядке, в котором были отправлены, продублироваться (приходят две копии одного пакета), оказаться повреждёнными (обычно повреждённые пакеты уничтожаются) или не прийти вовсе. Гарантию безошибочной доставки пакетов дают некоторые протоколы более высокого уровня — [транспортного уровня](#) сетевой модели OSI, — например, [TCP](#), которые используют IP в качестве транспорта.

IPv4 использует 32-битные (четырёхбайтные) адреса, ограничивающие [адресное пространство](#) 4 294 967 296 (2^{32}) возможными уникальными адресами.

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

IPv6 ([англ. Internet Protocol version 6](#)) — новая версия [протокола IP](#), призванная решить проблемы, с которыми столкнулась предыдущая версия ([IPv4](#)) при её использовании в [Интернете](#), за счёт использования длины адреса 128 [бит](#) вместо 32. Протокол был разработан [IETF](#).

Из IPv6 убраны функции, усложняющие работу маршрутизаторов:

- Маршрутизаторы больше не должны фрагментировать пакет, вместо этого пакет отбрасывается с ICMP-уведомлением о превышении MTU. Передающая сторона в IPv6, таким образом, обречена на использование технологии [Path MTU discovery](#). Для лучшей работы протоколов, требовательных к потерям, минимальный [MTU](#) поднят до 1280 байт. Фрагментация поддерживается как опция (информация о фрагментации пакетов вынесена из основного заголовка в расширенные) и возможна только по инициативе передающей стороны.
- Из IP-заголовка исключена контрольная сумма. С учётом того, что канальные ([Ethernet](#)) и транспортные ([TCP](#) и [UDP](#)) протоколы имеют свои контрольные суммы, ещё одна контрольная сумма на уровне IP воспринимается как излишняя. Кроме того, модификация поля *hop limit* (или *TTL* в IPv4) на каждом маршрутизаторе в IPv4 приводила к необходимости её постоянного пересчёта.

Несмотря на огромный размер адреса IPv6, благодаря этим улучшениям заголовок пакета удлинился всего лишь вдвое: с 20 до 40 байт.

Улучшения IPv6 по сравнению с IPv4:

- В сверхскоростных сетях возможна поддержка огромных пакетов (джамбограмм) — до 4 гигабайт;
- [Time to Live](#) переименовано в [Hop Limit](#);
- Появились метки потоков и классы трафика;
- Появилось многоадресное вещание.

Адреса IPv6 отображаются как восемь четырёхзначных [шестнадцатеричных чисел](#) (то есть групп по четыре символа), разделённых двоеточием. Пример адреса:

`2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d`

ARP ([англ.](#) *Address Resolution Protocol* — протокол определения адреса) — протокол в компьютерных сетях, предназначенный для определения [MAC-адреса](#) по известному [IP-адресу](#).

Рассмотрим суть функционирования ARP на простом примере. Компьютер А (IP-адрес 10.0.0.1) и компьютер Б (IP-адрес 10.22.22.2) соединены сетью [Ethernet](#). Компьютер А желает переслать пакет данных на компьютер Б, IP-адрес компьютера Б ему известен. Однако сеть Ethernet, которой они соединены, не работает с IP-адресами. Поэтому компьютеру А для осуществления передачи через Ethernet требуется узнать адрес компьютера Б в сети Ethernet (*MAC-адрес* в терминах Ethernet). Для этой задачи и используется протокол ARP. По этому протоколу компьютер А отправляет широковещательный запрос, адресованный всем компьютерам в одном с ним [широковещательном домене](#). Суть запроса: «компьютер с IP-адресом 10.22.22.2, сообщите свой *MAC-адрес* компьютеру с MAC-адресом (напр. a0:ea:d1:f1:01)». Сеть Ethernet доставляет этот запрос всем устройствам в том же сегменте Ethernet, в том числе и компьютеру Б. Компьютер Б отвечает компьютеру А на запрос и сообщает свой *MAC-адрес* (напр. 00:ea:d1:f1:11). Теперь, получив *MAC-адрес* компьютера Б, компьютер А может передавать ему любые данные через сеть Ethernet.

Наибольшее распространение ARP получил благодаря повсеместности сетей [IP](#), построенных поверх Ethernet, поскольку практически в 100 % случаев при таком сочетании используется ARP. В семействе протоколов [IPv6](#) ARP не существует, его функции возложены на [ICMPv6](#).

Описание протокола было опубликовано в ноябре 1982 года в [RFC 826](#). ARP был спроектирован для случая передачи [IP-пакетов](#) через сегмент Ethernet. При этом общий принцип, предложенный для ARP, может, и был использован и для сетей других типов.

Существуют следующие типы сообщений ARP: запрос ARP (ARP request) и ответ ARP (ARP reply). Система-отправитель при помощи запроса ARP запрашивает физический адрес системы-получателя. Ответ (физический адрес узла-получателя) приходит в виде ответа ARP.

Перед тем как передать пакет сетевого уровня через сегмент Ethernet, [сетевой стек](#) проверяет кэш ARP, чтобы выяснить, не зарегистрирована ли в нём уже нужная информация об узле-получателе. Если такой записи в кэше ARP нет, то выполняется широковещательный запрос ARP. Этот запрос для устройств в сети имеет следующий смысл: «Кто-нибудь знает физический адрес устройства, обладающего следующим IP-адресом?» Когда получатель с этим IP-адресом примет этот пакет, то должен будет ответить: «Да, это мой IP-адрес. Мой физический адрес следующий: ...» После этого отправитель обновит свой кэш ARP и будет способен передать информацию получателю. Ниже приведён пример запроса и ответа ARP. <см. внизу страницы>

Записи в кэше ARP могут быть статическими и динамическими. Пример, данный выше, описывает динамическую запись кэша. Можно также создавать статические записи в таблице ARP. Это можно сделать при помощи команды:

```
arp -s <IP-адрес> <MAC-адрес>
```

В системах семейства Windows до NT 6.0 записи в таблице ARP, созданные динамически, остаются в кэше в течение 2-х минут. Если в течение этих двух минут произошла повторная передача данных по этому адресу, то время хранения записи в кэше продлевается ещё на 2 минуты. Эта процедура может повторяться до тех пор, пока запись в кэше просуществует до 10 минут. После этого запись будет удалена из кэша, и будет отправлен повторный запрос ARP¹¹. Сейчас же время хранения записей в ARP таблице и метод хранения выбирается программно, и при желании его можно изменить.

18.Протокол DHCP

DHCP ([англ.](#) *Dynamic Host Configuration Protocol* — протокол динамической настройки узла) — [сетевой протокол](#), позволяющий компьютерам автоматически получать [IP-адрес](#) и другие параметры, необходимые для работы в сети [TCP/IP](#). Данный протокол работает по модели «[клиент-сервер](#)». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому [серверу DHCP](#) и получает от него нужные параметры. [Сетевой администратор](#) может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве сетей TCP/IP.

DHCP является расширением протокола [БООТР](#), использовавшегося ранее для обеспечения [бездисковых рабочих станций](#) IP-адресами при их загрузке. DHCP сохраняет [обратную совместимость](#) с БООТР.

Протокол DHCP предоставляет три способа распределения [IP-адресов](#):

- *Ручное распределение.* При этом способе сетевой администратор сопоставляет аппаратному адресу (для Ethernet сетей это [MAC-адрес](#)) каждого [клиентского](#) компьютера определённый IP-адрес. Фактически, данный способ распределения адресов отличается от ручной настройки каждого компьютера лишь тем, что сведения об адресах хранятся централизованно (на [сервере](#) DHCP), и потому их проще изменять при необходимости.
- *Автоматическое распределение.* При данном способе каждому компьютеру на постоянное использование выделяется произвольный свободный IP-адрес из определённого администратором диапазона.
- *Динамическое распределение.* Этот способ аналогичен автоматическому распределению, за исключением того, что адрес выдаётся компьютеру не на постоянное пользование, а на определённый срок. Это называется *ареной адреса*. По истечении срока аренды IP-адрес вновь считается свободным, и клиент обязан запросить новый (он, впрочем, может оказаться тем же самым). Кроме того, клиент сам может отказаться от полученного адреса.

Некоторые реализации службы DHCP способны автоматически обновлять записи [DNS](#), соответствующие клиентским компьютерам, при выделении им новых адресов. Это производится при помощи протокола обновления DNS, описанного в [RFC 2136](#).

Помимо IP-адреса, DHCP также может сообщать клиенту дополнительные параметры, необходимые для нормальной работы в сети. Эти параметры называются *опциями DHCP*. Список стандартных опций можно найти в [RFC 2132](#).

Опции – строки переменной длины, состоящие из октетов. Первый октет - код опции, второй октет – количество следующих октетов, остальные октеты зависят от кода опции.

Например, опция “DHCP Message Type” при отправке сообщения “Offer” будет выглядеть так : 0x35,0x01,0x02, где 0x35 – код опции “DHCP Message Type”, 0x01 – означает, что далее идет только один октет, 0x02 – значение “Offer”.

Некоторыми из наиболее часто используемых опций являются:

- IP-адрес [маршрутизатора](#) по умолчанию;
- [маска подсети](#);
- адреса [серверов DNS](#);
- имя домена [DNS](#).

19.Методы маршрутизации. Функции маршрутизатора

Различают три вида маршрутизации - простую, фиксированную и адаптивную. Принципиальная разница между ними - в степени учета изменения топологии и нагрузки сети при решении задачи выбора маршрута.

Простая маршрутизация отличается тем, что при выборе маршрута не учитывается ни изменение топологии сети, ни изменение ее состояния (нагрузки). Она не обеспечивает направленной передачи пакетов и имеет низкую эффективность. Ее преимущества - простота реализации алгоритма маршрутизации и обеспечение устойчивой работы сети при выходе из строя отдельных ее элементов. Из этого вида некоторое практическое применение получили случайная и лавинная маршрутизации.

Случайная маршрутизация характеризуется тем, что для передачи пакета из узла связи выбирается одно, случайно выбранное свободное направление. Пакет "блуждает" по сети и с конечной вероятностью когда-либо достигает адресата. Естественно, что при этом не обеспечивается ни оптимальное время доставки пакета, ни эффективное использование пропускной способности сети.

Лавинная маршрутизация (или: заполнение пакетами всех свободных выходных направлений) - предусматривает передачу пакета из узла по всем свободным выходным линиям. Поскольку это происходит в каждом узле, имеет место явление "размножения" пакета, что резко ухудшает использование пропускной способности сети. Значительное ослабление этого недостатка достигается путем уничтожения в каждом узле дубликатов (копий) пакета и продвижения по маршруту только одного пакета. Основное преимущество такого метода - гарантированное обеспечение оптимального времени доставки пакета адресату, так как из всех направлений, по которым передается пакет, хотя бы одно обеспечивает такое время. Метод может использоваться в незагруженных сетях, когда требования по минимизации времени и надежности доставки пакетов достаточно высоки.

Фиксированная маршрутизация характеризуется тем, что при выборе маршрута учитывается изменение топологии сети и не учитывается изменение ее нагрузки. Для каждого узла назначения направление передачи выбирается по таблице маршрутов (каталогу), которая определяет кратчайшие пути. Каталоги составляются в центре управления сетью. Они составляются заново при изменении топологии сети. Отсутствие адаптации к изменению нагрузки приводит к задержкам пакетов сети. Различают однопутевую и многопутевую фиксированную маршрутизации. Первая строится на основе единственного пути передачи пакетов между двумя абонентами, что сопряжено с неустойчивостью к отказам и перегрузкам, а вторая - на основе нескольких возможных путей между двумя абонентами, из которых выбирается предпочтительный путь. Фиксированная маршрутизация применяется в сетях с мало изменяющейся топологией и установившимися потоками пакетов.

Основная функция маршрутизатора — чтение заголовков пакетов сетевых протоколов, принимаемых по каждому порту и принятие решения о дальнейшем маршруте следования пакета по его сетевому адресу.

20. Динамическая маршрутизация. Протокол RIP

Динамическая маршрутизация — вид маршрутизации, при котором таблица маршрутизации редактируется программно. В случае UNIX-систем *демонами маршрутизации*; в других системах — служебными программами, которые называются иначе, но фактически играют ту же роль.

Когда маршрутизатор отправляет обновление RIP, он добавляет к метрике маршрута, которую он использует, 1 и отправляет соседу. Сосед получает обновление, в котором указано какую метрику для полученного маршрута ему использовать.

Маршрутизатор отправляет каждые 30 секунд все известные ему маршруты соседним маршрутизаторам. Но, кроме этого, для предотвращения петель и для улучшения времени сходимости, используются дополнительные механизмы:

- **Split horizon** — если маршрут достижим через определенный интерфейс, то в обновление, которое отправляется через этот интерфейс не включается этот маршрут;
- **Triggered update** — обновления отправляются сразу при изменении маршрута, вместо того чтобы ожидать когда истечет Update timer;
- **Route poisoning** — это принудительное удаление маршрута и перевод в состояние удержания, применяется для борьбы с маршрутными петлями.
- **Poison reverse** — Маршрут помечается, как не достижимый, то есть с метрикой 16 и отправляется в обновлениях.

В обновлениях RIPv2 могут передаваться до 25 сетей.

Или так:

Протокол маршрутной информации ([англ. Routing Information Protocol](#)) — один из самых простых протоколов маршрутизации. Применяется в небольших [компьютерных сетях](#), позволяет [маршрутизаторам](#) динамически обновлять маршрутную информацию (направление и дальность в [хопах](#)), получая её от соседних маршрутизаторов.

[Алгоритм](#) маршрутизации RIP ([алгоритм Беллмана — Форда](#)) был впервые разработан в [1969 году](#), как основной для сети [ARPANET](#).

Прототип протокола RIP — [Gateway Information Protocol](#), часть пакета [PARC Universal Packet](#).

Версия RIP, которая поддерживает [протокол интернета](#) была включена в пакет [BSD](#) операционной системы [Unix](#) под названием *routed* (route daemon), а также многими производителями, реализовавшими свою версию этого протокола. В итоге протокол был унифицирован в документе [RFC 1058](#).

В период с 1993 по 1998 годы разрабатывался протокол RIP-2 ([RFC 2453](#)), который является расширением протокола RIP, обеспечивающим передачу дополнительной маршрутной информации в сообщениях RIP и повышающим уровень безопасности.

Для работы в среде [IPv6](#) была разработана версия [RIPng](#).

RIP — так называемый протокол [дистанционно-векторной маршрутизации](#), который оперирует [транзитными участками](#) в качестве метрики маршрутизации. Максимальное количество хопов, разрешенное в RIP — 15 (метрика 16 означает «бесконечно большую метрику»). Каждый RIP-маршрутизатор по умолчанию вешает в сеть свою полную таблицу маршрутизации раз в 30 секунд, довольно сильно нагружая низкоскоростные линии связи. RIP работает на 4 уровне (уровень приложения) стека [TCP/IP](#), используя [UDP](#) порт 520.

В современных сетевых средах RIP — не самое лучшее решение для выбора в качестве протокола маршрутизации, так как его возможности уступают более современным протоколам, таким как [EIGRP](#), [OSPF](#). Ограничение на 15 хопов не дает применять его в больших сетях. Преимущество этого протокола — простота конфигурирования.

21. Протокол OSPF

Тут приведено краткое описание работы протокола, которое подробнее описано ниже, в соответствующих разделах. Часть из этих этапов, специфичны для конкретной реализации, и указаны на соответствующих страницах настройки OSPF.

Задача этого раздела дать общее понимание того, как работает протокол. Не все пункты могут быть до конца понятны, но общее представление, скорее всего, появится.

1. Включить OSPF на маршрутизаторе
2. Маршрутизатор выбирает Router ID (的独特ное имя маршрутизатора)
3. Включить OSPF на интерфейсах (чтобы протокол знал о каких интерфейсах можно сообщать другим маршрутизаторам)
4. Обнаружение соседей с помощью Hello-пакетов
 1. Маршрутизаторы обмениваются hello-пакетами через все интерфейсы, на которых активирован OSPF.
 2. Маршрутизаторы, которые находятся в одном широковещательном сегменте, становятся соседями, когда они приходят к договоренности об определенных параметрах, указанных в их hello-пакетах.
5. Adjacency (отношения соседства, отношения смежности) это тип соседства между маршрутизаторами, по которому они синхронизируют LSDB. Установка этих отношений зависит от типа сети:
 1. Если маршрутизаторы находятся в сети с множественным доступом, они выбирают DR и выполняют синхронизацию LSDB с ним
 2. Если маршрутизаторы находятся в сети point-to-point, они приступают к синхронизации LSDB друг с другом
6. Синхронизация LSDB. Происходит в несколько этапов. По сформированным отношениям соседства происходит обмен такими пакетами:
 1. DBD (краткое описание LSA в LSDB). С помощью этих пакетов маршрутизаторы сообщают друг другу о том, какую информацию они знают, в сокращенном виде
 2. LSR. После обмена DBD-пакетами, с помощью LSR маршрутизаторы запрашивают у соседа недостающую информацию
 3. LSU (содержит полное описание LSA). В ответ на LSR, который ему прислал сосед, маршрутизатор отправляет LSU, с полным описанием информации, которой не хватает у соседа
 4. LSAck. После получения LSU от соседа, маршрутизатор отправляет подтверждение, что он получил информацию
 5. Если оба маршрутизатора должны запросить друг у друга информацию, то эта процедура повторяется и в другую сторону.
 6. После этого, LSDB синхронизирована, а значит, полностью одинакова между соседями
7. После синхронизации LSDB, маршрутизатор отправляет обновление далее, своим соседям в других широковещательных сегментах
8. Рассылая объявление через зону, все маршрутизаторы строят идентичную LSDB
9. Когда база данных построена, каждый маршрутизатор использует алгоритм SPF (shortest path first) для вычисления графа без петель, который будет описывать кратчайший путь к каждому известному пункту назначения с собой в качестве корня. Этот граф — дерево кратчайшего пути.
10. Каждый маршрутизатор строит таблицу маршрутизации, основываясь на своем дереве кратчайшего пути.

22. Протокол UDP. Передача данных без установления соединения

UDP ([англ.](#) *User Datagram Protocol* — протокол пользовательских [датаграмм](#)) — один из ключевых элементов [TCP/IP](#), набора сетевых протоколов для [Интернета](#). С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые [датаграммами](#)) другим хостам по [IP-сети](#) без

необходимости предварительного сообщения для установки специальных каналов передачи или путей данных. Протокол был разработан Дэвидом П. Ридом в 1980 году и официально определён в [RFC 768](#).

UDP использует простую модель передачи, без неявных «рукопожатий» для обеспечения надёжности, упорядочивания или целостности данных. Таким образом, UDP предоставляет ненадёжный сервис, и датаграммы могут прийти не по порядку, дублироваться или вовсе исчезнуть без следа. UDP подразумевает, что проверка ошибок и исправление либо не нужны, либо должны исполняться в приложении. Чувствительные ко времени приложения часто используют UDP, так как предпочтительнее сбросить пакеты, чем ждать задержавшиеся пакеты, что может оказаться невозможным в [системах реального времени](#). При необходимости исправления ошибок на сетевом уровне интерфейса приложение может задействовать [TCP](#) или [SCTP](#), разработанные для этой цели.

Природа UDP как протокола без сохранения состояния также полезна для серверов, отвечающих на небольшие запросы от огромного числа клиентов, например [DNS](#) и [потоковые мультимедийные приложения](#) вроде [IPTV](#), [Voice over IP](#), [протоколы туннелирования IP](#) и многие [онлайн-игры](#).

UDP — минимальный ориентированный на обработку сообщений протокол [транспортного уровня](#), задокументированный в [RFC 768](#).

UDP не предоставляет никаких гарантий доставки сообщения для вышестоящего протокола и не сохраняет состояния отправленных сообщений. По этой причине UDP иногда называют Unreliable Datagram Protocol (англ. — Ненадёжный протокол датаграмм).

UDP обеспечивает [многоканальную передачу](#) (с помощью номеров портов) и проверку целостности (с помощью [контрольных сумм](#)) заголовка и существенных данных. Надёжная передача в случае необходимости должна реализовываться пользовательским приложением.

Биты	0 - 15	16 - 31
-------------	--------	---------

0-31	Порт отправителя (Source port)	Порт получателя (Destination port)
-------------	--------------------------------	------------------------------------

32-63	Длина датаграммы (Length)	Контрольная сумма (Checksum)
--------------	---------------------------	------------------------------

64...	Данные (Data)
--------------	---------------

Заголовок UDP состоит из четырёх полей, каждое по 2 байта (16 бит). Два из них необязательны к использованию в IPv4 (розовые ячейки в таблице), в то время как в IPv6 необязателен только порт отправителя.

23.Протокол TCP. Потоковая передача данных

TCP ([англ.](#) *transmission control protocol*) — протокол управления передачей) — один из основных [протоколов передачи данных](#) интернета, предназначенный для управления [передачей данных](#). Сети и подсети, в которых совместно используются протоколы TCP и [IP](#) называются сетями [TCP/IP](#).

В [стеке протоколов IP](#) TCP выполняет функции протокола [транспортного уровня модели OSI](#).

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

Реализации TCP обычно встроены в [ядра ОС](#). Существуют реализации TCP, работающие в [пространстве пользователя](#).

Когда осуществляется передача от компьютера к компьютеру через Интернет, TCP работает на верхнем уровне между двумя конечными системами, например, [браузером](#) и веб-сервером. TCP осуществляет надежную передачу потока байтов от одной программы на некотором компьютере к другой программе на другом компьютере (например, программы для электронной почты, для обмена файлами). TCP контролирует длину сообщения, скорость обмена сообщениями, сетевой трафик.

Установка соединения

Процесс начала сеанса TCP (также называемый «рукопожатие» ([англ. handshake](#))), состоит из трёх шагов.

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

- Сервер получает сегмент, запоминает номер последовательности и пытается создать сокет (буферы и управляющие структуры памяти) для обслуживания нового клиента.
 - В случае успеха сервер посыпает клиенту сегмент с номером последовательности и флагами SYN и ACK, и переходит в состояние SYN-RECEIVED.
 - В случае неудачи сервер посыпает клиенту сегмент с флагом RST.

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

- Если он одновременно получает и флаг ACK (что обычно и происходит), то он переходит в состояние ESTABLISHED.
- Если клиент получает сегмент с флагом RST, то он прекращает попытки соединиться.
- Если клиент не получает ответа в течение 10 секунд, то он повторяет процесс соединения заново.

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

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

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

Передача данных

См. также: [Алгоритм Нейгla](#) и [Медленный старт](#)

При обмене данными приемник использует номер последовательности, содержащийся в получаемых сегментах, для восстановления их исходного порядка. Приемник уведомляет передающую сторону о номере последовательности, до которой он успешно получил данные, включая его в поле «номер подтверждения». Все получаемые данные, относящиеся к промежутку подтвержденных последовательностей, игнорируются. Если полученный сегмент содержит номер последовательности больший, чем ожидаемый, то данные из сегмента буферизируются, но номер подтвержденной последовательности не изменяется. Если впоследствии будет принят сегмент, относящийся к ожидаемому номеру последовательности, то порядок данных будет автоматически восстановлен исходя из номеров последовательностей в сегментах.

Для того, чтобы передающая сторона не отправляла данные интенсивнее, чем их может обработать приемник, TCP содержит средства управления потоком. Для этого используется поле «окно». В сегментах, направляемых от приемника передающей стороне, в поле «окно» указывается текущий размер приемного буфера. Передающая сторона сохраняет размер окна и отправляет данных не более, чем указал приемник. Если приемник указал нулевой размер окна, то передача данных в направлении этого узла не происходит, пока приемник не сообщит о большем размере окна.

В некоторых случаях передающее приложение может явно потребовать протолкнуть данные до некоторой последовательности принимающему приложению, не буферизируя их. Для этого используется флаг PSH. Если в полученном сегменте обнаруживается флаг PSH, то реализация TCP отдает все буферизированные на текущий момент данные принимающему приложению. «Проталкивание» используется, например, в интерактивных приложениях. В сетевых терминалах нет смысла ожидать ввода пользователя после того, как он закончил набирать команду. Поэтому последний сегмент, содержащий команду, обязан содержать флаг PSH, чтобы приложение на принимающей стороне смогло начать её выполнение.

Завершение соединения

Завершение соединения можно рассмотреть в три этапа:

1. Посылка серверу от клиента флага FIN на завершение соединения.
2. Сервер посыпает клиенту флаги ответа ACK , FIN, что соединение закрыто.
3. После получения этих флагов клиент закрывает соединение и в подтверждение отправляет серверу ACK , что соединение закрыто.

24. Глобальные сети

Глобальная сеть — любая [сеть связи](#), которая охватывает всю Землю. Термин, используемый в данной статье, относится в более узком смысле к двунаправленным сетям связи, а также базе технологий сетей. Ранние сети, такие как международные почтовые отправления, и односторонние сети связи, такие как радио и телевидение, не рассматриваются. Первая глобальная сеть была создана с помощью электрического телеграфа и достигла глобального размаха в 1899 году. Телефонные сети были вторыми и достигли глобального статуса в 1950-х годах. Совсем недавно взаимосвязанные IP-сети (в основном Интернет, по оценкам, 360 миллионов пользователей по всему миру в 2009 году), а также мобильные GSM-сети (более 3 миллиардов пользователей по всему миру в 2009 году) образовали крупнейшие глобальные сети из всех.

Спутниковые глобальные сети

Спутники связи — важная часть глобальных сетей. Имеются определенные глобальные группировки из низкоорбитальных спутников, такие как Iridium, Globalstar и Orbcomm, которые состоят из множества аналогичных спутников, выходят на орбиту расположенную с равными интервалами позициях и формируют mesh-сеть, иногда отправляя и получая информацию непосредственно между собой. Спутниковый доступ в Интернет стал возможным благодаря технологии VSAT.

Мобильные беспроводные сети

Считается^[кем?], что 80 % глобального рынка мобильной связи использует стандарт GSM^[источник не указан 62 дня], существующий больше чем в 212 странах и территориях. Его повсеместность делает международный роуминг очень распространенным между операторами мобильной телефонии, позволяя подписчикам использовать их телефоны во многих частях мира. Чтобы достигнуть этого, данные сети должны быть соединены посредством пиринга, и поэтому сеть GSM — действительно глобальная.

Межсетевое взаимодействие

Телеграф и сети связи telex были постепенно сокращены, таким образом, взаимодействие среди существующих глобальных сетей возникает в нескольких точках, например, такой как между речевой телефонией и сетями цифровых данных, и между этими и спутниковыми сетями. На данный момент множество приложений работают в нескольких сетях, таких как VoIP (речь по IP). Сети мобильной связи (речь и данные) также тесно пересекаются, потому что у сотовых телефонов XXI века есть возможность передачи речи и данных (интернет-навигация и посылка сообщений по электронной почте). Цифровые глобальные сети требуют огромной пропускной способности в основных магистралях. Это в настоящее время достигнуто [оптиковолоконными](#) кабелями.