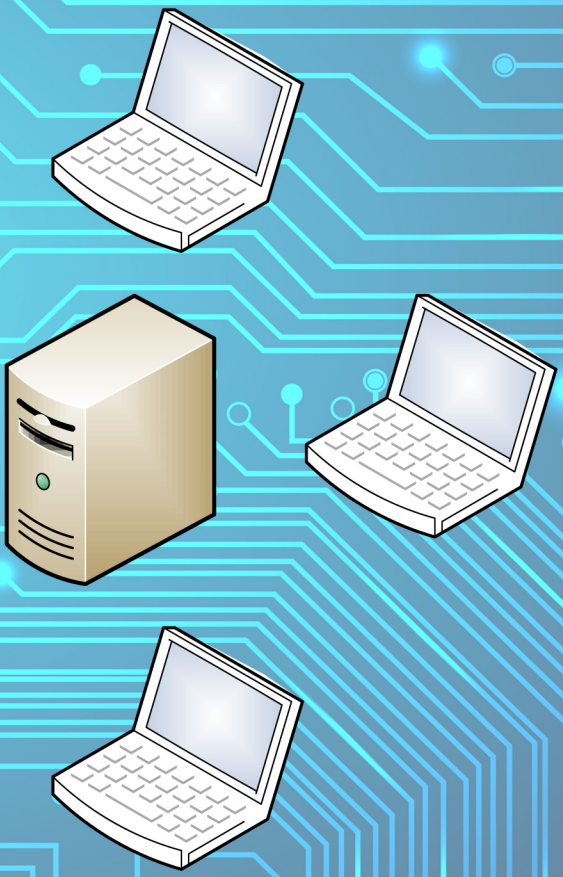
Е.В. Петрунина, О.Н. Савельева, Т.В. Гончарук

Компьютерные сети

Учебное пособие



Москва
2017

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Московский государственный
гуманитарно-экономический университет

Е.В. Петрунина, О.Н. Савельева
Т.В. Гончарук

КОМПЬЮТЕРНЫЕ СЕТИ

Учебное пособие

Москва
2017

УДК 681.324
ББК 32.973.202
П 31

Рецензенты:
М.В. Жиров, доктор техн. наук, профессор кафедры прикладной математики и информатики МГГЭУ;
С.А. Красников, доктор техн. наук, зав. кафедрой «Информационные технологии» МГУТУ им. К.Г. Разумовского.

Е.В. Петрунина, О.Н. Савельева, Т.В. Гончарук
П 31 Компьютерные сети: *учебное пособие*. – М.: МГГЭУ, 2017. – 114 с.

В учебном пособии рассмотрены основные принципы функционирования компьютерных сетей. Проанализирована модель системы передачи информации, дано описание различных линий связи и аппаратуры каналов связи. Приведена классификация компьютерных сетей, а также эталонная модель взаимодействия открытых систем. Даны основы теории кодирования и примеры построения двух корректирующих кодов. Большое внимание уделено стеку протоколов TCP/IP.

Для студентов младших курсов, обучающихся по направлениям подготовки «Прикладная математика и информатика», «Прикладная информатика» и «Информатика и вычислительная техника»

ISBN 978-5-9799-0097-1

Содержание

ВВЕДЕНИЕ

Учебное пособие представляет собой введение в сетевую тематику и дает базовые знания по организации и функционированию сетей. В пособии приведены общие понятия компьютерных сетей, их структуры, принципы функционирования. Проанализирована модель системы передачи информации, дано описание различных линий связи и аппаратуры каналов связи. Передача данных в сети рассматривается на базе эталонной базовой модели, разработанной Международной организацией по стандартам взаимодействия открытых сетей. Описываются правила и процедуры передачи данных между информационными системами.

В пособии приведена классификация компьютерных сетей, а также даны основы теории кодирования и примеры построения корректирующих кодов. Большое внимание уделено стеку протоколов TCP/IP.

1. ПЕРЕДАЧА ИНФОРМАЦИИ

1.1. Модель системы передачи информации

Информация всегда может быть представлена в виде некоторого сообщения, которое передается определенной физической средой. Под передачей информации можно понимать процесс ее распространения от источника к потребителю.

Рассмотрим простой пример. Общаются два человека: один говорит, а другой слушает его. В данном примере источник и передатчик информации — это одно и то же лицо, т.е. говорящий человек. Аналогичным образом дело обстоит и в отношении слушателя: он является одновременно приемником и потребителем полученной информации.

Однако, когда мы имеем дело с передачей информации по каналам связи, возникшую информацию необходимо преобразовать, чтобы подготовить ее к передаче.

Таким образом, в общем случае модель системы передачи информации имеет вид, представленный на *рис. 1.1*. Источник информации осуществляет кодирование информационного сообщения путем устранения присутствующей в нем избыточности и преобразует его в первичный электрический сигнал, формируя сообщение *A* на входе системы передачи.

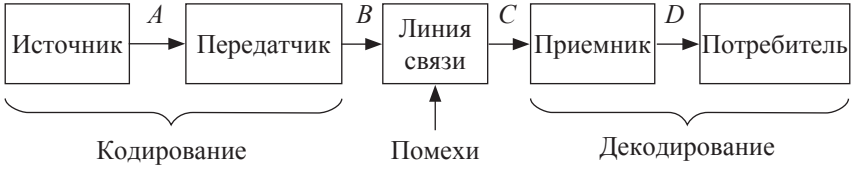


Рис. 1.1. Модель системы передачи информации

Например, буквы алфавита любого языка передаются с помощью двух символов — 0 и 1.

Передатчик готовит поступившее сообщение к передаче по определенной линии связи, преобразуя его в сигнал *B* на ее входе. В реальных условиях передачи присутствуют помехи, искажающие передаваемый сигнал, поэтому необходимо осуществлять помехоустойчивое кодирование. Данное кодирование заключается во введении избыточности в кодовую комбинацию первичного сигнала.

Как правило, первичный низкочастотный электрический сигнал *A* обычно преобразуется в передатчике во вторичный высокочастотный сигнал *B*, который и передается по линии связи.

Сигнал *C* на выходе линии связи, поступающий на вход приемника, из-за помех может отличаться от переданного сигнала. Задача приемника состоит в формировании сообщения *D* на выходе системы передачи, т.е. в обработке принятого сигнала, его декодировании и восстановлении по нему переданного сообщения, соответствующего исходному информационному сообщению *A*, которое и поступает потребителю.

Потребитель информации, получив сообщение, т.е. кодовую комбинацию символов первичного сигнала, сначала преобразует его, осуществляя декодирование, после чего оно превращается в соответствующую информацию, которая и была предназначена для получателя.

Рассмотренная обобщенная модель системы передачи информации может различаться в деталях при организации разных систем связи. Например, в телеграфии при передаче текстовых сообщений упомянутые сообщения *A* и *D* на входе и выходе системы передачи записываются с помощью буквенного алфавита на одном и том же языке, а различия между ними могут возникнуть только в результате искажений в процессе передачи под воздействием помех. Сигналы же *B* и *C* на входе и выходе линии связи (при этом также надо учитывать наличие помех) являются последовательностями элементарных электрических сигналов — чаще всего посылок тока и пауз. Следовательно, операции кодирования и декодирования в этом случае заключаются в преобразовании буквенного сообщения *A* в последовательность элементарных сигналов *B* и в обратном переходе от принятой последовательности сигналов *C* к буквенному сообщению *D*.

В телефонии сообщение *A*, имеющее характер звука, представляет собой определенные колебания давления, которые на этапе кодирования преобразуются в колебания электрического тока *B*, а при декодировании

осуществляется обратное преобразование принятых колебаний тока *C* в звук *D*.

Рассмотрим приведенную модель системы передачи информации применительно к компьютеру. Сообщение на входе *A* является определенной последовательностью чисел, которая на этапе кодирования преобразуется в соответствующую последовательность электрических сигналов *B*, непосредственно вводимых в машину. Декодирование состоит в преобразовании поступивших в компьютер сигналов *C*, которые являются суммой принятых сигналов и искажений при вводе. В результате получается новое сообщение *D*, принципиально отличающееся от сообщения *A*. Соответственно, основной целью рассмотренной линии связи является преобразование *A* в *D*.

В реальных условиях полная информационная модель передачи информации включает в себя передатчик, кодирующее устройство, кодер канала, канал связи, декодер канала, декодирующее устройство и приемник.

Передатчик преобразует поступающее от источника сообщение в сигнал для передачи по каналу, приемник осуществляет обратное преобразование сигнала в сообщение для потребителя. Для согласования характеристик передатчика и приемника с характеристиками канала связи вводятся соответствующие устройства.

Кодирующее устройство устраняет избыточность входной информации, что уменьшает среднее число символов в сообщениях.

Кодер канала обеспечивает достоверность передачи при наличии помех путем введения избыточности. В итоге имеем устранение избыточности с последующим перекодированием помехоустойчивым кодом. При малой избыточности и отсутствии помех в канале введение обоих устройств нецелесообразно, однако возможны и другие реальные ситуации, которые обосновывают их наличие.

Декодирующее устройство и декодер канала осуществляют обратные преобразования.

Итак, протекание процесса передачи и приема или распространения информации обеспечивают источник, передатчик, физическая среда (канал, линия связи), приемник и потребитель.

1.2. Каналы и линии связи

Под каналом связи будем понимать технические средства, которые обеспечивают распространение электрических сигналов от передатчика к при-

емнику. Линия связи — это физическая среда, обеспечивающая передачу информации. По своей природе линии делятся на механические, акустические, оптические и электрические (проводные и беспроводные).

Математическая модель описания реальных линий передачи с учетом их статистических свойств имеет вход и выход, служит для преобразования входных сообщений в выходные и характеризуется случайным процессом (последовательностью) на входе и распределением вероятностей преобразования входных сигналов в выходные, отражающим характер возможных искажений при передаче информации.

В настоящее время в вычислительных системах в качестве передающей среды используются выделенные телефонные линии, радиоканалы, каналы спутниковой связи, специальные каналы передачи цифровой информации, а также витая пара проводов, коаксиальный и оптоволоконный кабели. Тип канала определяет характер и величину помех, которые неизбежно возникают при передаче информации. Рассмотрим его основные характеристики.

Характеристики каналов связи

Скорость передачи V определяется количеством информации I, переданной за время T, и имеет размерность бит/с, т.е.

$$V = \frac{I}{T}.$$

Предельное значение скорости передачи называется пропускной способностью C или емкостью канала связи и имеет размерность бит/с или бод, т.е.

$$C = \lim_{T \to \infty} \frac{I}{T}.$$

Максимально возможная пропускная способность идеального канала связи определяется формулой Шеннона:

$$CF = \log_2 \left(1 + \left| \frac{W_c}{W_ш} \right| \right),$$

где F — полоса пропускания канала (Гц); $W_с$ и $W_ш$ — мощности сигнала и шума соответственно (Вт).

Необходимо отметить, что у реального канала связи всегда меньшая полоса пропускания из-за ряда неучтенных ограничений. Анализируя данное выражение, можно сделать следующие выводы. При высоком уровне шумов $(W_ш\ W_c)$ максимальная пропускная способность С близка к нулю. С растет быстрее с ростом F, чем с увеличением отношения $W_c/W_ш$. Наконец, для обеспечения C = const при сужении полосы пропускания F необходимо увеличивать мощность сигнала W_c, и наоборот.

Рассмотрим канальные характеристики.

Полоса пропускания канала представляет собой диапазон частот, для которых отношение амплитуды сигнала на выходе канала к амплитуде входного сигнала больше некоторой заданной величины, например 0,5. Таким образом, данная характеристика определяет диапазон частот, при которых сигнал передается без существенных искажений.

Затухание есть относительное уменьшение амплитуды или мощности передаваемого по каналу сигнала определенной частоты. Обычно эта характеристика измеряется в децибелах (дБ).

Амплитудно-частотная характеристика демонстрирует изменения амплитуд выходного сигнала по сравнению с амплитудами входного сигнала для всех частот полосы пропускания канала.

Пропускная способность канала связи существенно зависит от способов логического и физического кодирования передаваемой информации. При логическом кодировании исходная последовательность бит информации заменяется новой, более длинной последовательностью бит. Это приводит к уменьшению пропускной способности, однако вводимая избыточность лежит в основе построения корректирующих кодов (о чем речь пойдет ниже), которые позволяют обнаруживать и даже исправлять ошибки, возникающие при передаче информации. При физическом кодировании предназначенная для передачи дискретная информация представляется в виде сигналов того или иного типа, поступающих в канал связи. Ясно, что выбранный способ такого кодирования влияет на спектр передаваемого сигнала и в конечном итоге на пропускную способность канала.

Чтобы канал связи при передаче информации использовался наиболее эффективно, необходимо согласовывать скорость передачи с пропускной способностью, т.е. обеспечить выполнение условия

$$V \le C.$$

При этом желательно, чтобы скорость передачи была как можно ближе к пропускной способности. Если же указанное условие не выполняется, т.е. скорость поступления информации на вход канала связи больше его пропускной способности, то в этом случае по каналу будет передана не вся информация.

Такое согласование осуществляется путем соответствующего кодирования информационных сообщений, которое называется также статистическим кодированием. К. Шеннон связал способ кодирования сообщений со скоростью их передачи по каналам и вероятностью возникновения при этом искажений. Он доказал, что при отсутствии шумов всегда можно закодировать сообщения так, чтобы информация передавалась самыми короткими кодовыми словами. При наличии же помех можно передавать информацию без потерь. Для таких каналов, если скорость передачи не больше пропускной способности, всегда существует способ кодирования, при реализации которого информация будет передаваться со сколь угодно малой вероятностью ошибок.

Таким образом, пропускная способность канала связи — это верхнее значение скорости передачи информации по всем возможным распределениям сообщений на входе канала. Распределение вероятностей сообщений на выходе канала при известном сообщении на его входе является фиксированной характеристикой канала. Следовательно, для канала с заданной пропускной способностью всегда можно подобрать распределение сообщений на входе, для которого скорость передачи информации будет сколь угодно мало отличаться от значения его пропускной способности, при этом вероятность ошибки хотя бы в одном символе в течение любого фиксированного отрезка времени можно сделать сколь угодно малой.

Достоверность передачи информации определяется вероятностью ошибочного приема информационного символа — Рош. Другими словами, эта характеристика есть отношение числа ошибочно принятых знаков к общему количеству переданных знаков и обычно составляет для каналов величину порядка 10^{-6} (в оптоволоконных линиях связи она равна 10^{-9}). Таким образом, если Рош = 10^{-6}, то в среднем из миллиона переданных бит информации один бит окажется ошибочным.

Надежность передачи определяется средним временем безотказной работы и для вычислительных сетей обычно составляет как минимум несколько тысяч часов.

1.3. Типы линий связи

Линии связи классифицируются по типу физической среды, которая осуществляет передачу сигнала от передатчика к приемнику.

В низко- и среднескоростных каналах связи в качестве передающей среды, как правило, используются группы параллельных или скрученных проводов, называемых витой парой. Скручивание проводов (соответствующая характеристика называется шагом скручивания) уменьшает влияние внешних электромагнитных полей на процесс передачи информации.

В широкополосных каналах применяются коаксиальные и оптоволоконные кабели, радио-волноводы, а также беспроводные радиоканалы связи.

Наиболее часто в компьютерных сетях применяются кабельные соединения, выступающие в качестве среды электрических или оптических сигналов между компьютерами и другими сетевыми устройствами.

Кабель — это изделие, состоящее из проводников, слоев экрана и изоляции.

Важнейшие характеристики:

- *Коэффициент затухания*, дБ/км. Зависит от свойств материалов проводников и изоляционного материала. Наилучшими свойствами (малым сопротивлением) обладают медь и серебро. Коэффициент затухания зависит также от геометрических размеров проводников.
- *Скорость распространения*, км/мс. С ростом частоты скорость распространения увеличивается, приближаясь к скорости света в вакууме 300 км/мс. Данный параметр зависит также от свойств диэлектрика, применяемого в кабеле.
- *Перекрестные наводки на ближнем конце* (Near End Cross Talk, NEXT).
- *Волновое сопротивление (импеданс)* Ом — сопротивление, которое встречает электромагнитная волна при распространении вдоль однородной линии без отражения, т.е. при условии, что на процесс передачи не влияют несогласованности на концах линии. Волновое сопротивление симметричного кабеля зависит от удельных значений емкости и индуктивности кабеля.
- *Активное сопротивление* — это сопротивление постоянному току в электрической цепи. В отличие от импеданса активное сопротивление не зависит от частоты и возрастает с увеличением длины кабеля.

- *Емкость* — это свойство металлических проводников накапливать энергию. Два электрических проводника в кабеле, разделенные диэлектриком, представляют собой конденсатор, способный накапливать заряд. Емкость является нежелательной величиной, поэтому следует стремиться к тому, чтобы она была как можно меньше (иногда применяют термин «паразитная емкость»). Высокое значение емкости в кабеле приводит к искажению сигнала и ограничивает полосу пропускания линии.
- *Электрический шум* — это нежелательное переменное напряжение в проводнике. Электрический шум бывает двух типов: фоновый и импульсный. Электрический шум можно также разделить на низко-, средне- и высокочастотный. Источниками фонового электрического шума в диапазоне до 150 кГц являются линии электропередачи, телефоны и лампы дневного света; в диапазоне от 150 кГц до 20 МГц — компьютеры, принтеры, ксероксы; в диапазоне от 20 МГц до 1 ГГц — телевизионные и радиопередатчики, микроволновые печи. Основными источниками импульсного электрического шума являются моторы, переключатели и сварочные агрегаты. Электрический шум измеряется в милливольтах.
- *Диаметр или площадь сечения проводника*. В линиях связи используются следующие типы кабеля:
  — витая пара (twistedpair);
  — коаксиальный кабель (coaxialcable);
  — оптоволоконный кабель (fideroptic).

Витая пара, состоящая из двух изолированных медных проводников, может быть неэкранированной или экранированной. В первом случае такой кабель имеет самую слабую помехозащищенность и низкую скорость передачи. Линии связи обычно не длиннее ста метров, так как затухание сигнала при его прохождении по кабелю слишком велико.

В случае экранированной витой пары каждая из них помещается в металлическую оплетку-экран, размещаемую внутри пластмассовой защитной оболочки. При этом существенно снижаются излучения кабеля, воздействия внешних электромагнитных полей и взаимовлияния пар медных проводов, что достигается, однако, за счет значительного увеличения стоимости. Кроме того, при использовании таких линий необходимы специальные экранированные разъемы, поэтому они применяются гораздо реже.

Коаксиальный кабель имеет внутренний центральный проводник, в качестве которого применяется медная жила. Внешний проводник, являющийся внешним экраном, отделен от центрального проводника слоем изоляции, и все это размещено внутри защитной оболочки. Промышленностью выпускаются два типа таких кабелей: тонкий (более современный, гибкий, дешевый, диаметром около 0,5 см) и толстый (диаметром около 1 см). Коаксиальный кабель по сравнению с витой парой обладает лучшими параметрами: допустимая скорость передачи информации составляет 50 Мбит/с на расстояние до километра. Кроме того, к такому кабелю значительно труднее подключиться, если пытаться осуществить несанкционированное прослушивание компьютерной сети.

Оптоволоконный кабель по своей структуре похож на коаксиальный кабель: он содержит центральное оптическое стеклянное волокно, являющееся проводником света, которое заключено в стеклянное же покрытие, и все это размещается внутри защитной оболочки. Стеклянное покрытие обладает меньшим показателем преломления, чем центральное волокно, поэтому световые лучи, распространяясь по сердцевине и отражаясь от покрытия, не покидают центральный проводник. В зависимости от диаметра последнего и распределения показателя преломления различают три режима распространения лучей света по сердцевине, определяемые термином «мода».

В одномодовом кабеле центральное волокно имеет очень малый диаметр, соизмеримый с длиной волны света, поэтому световые лучи распространяются вдоль оптической оси, не отражаясь от внешнего по отношению к нему стеклянного покрытия. В качестве источника излучения света в таких кабелях используются полупроводниковые лазеры. Многомодовые кабели бывают двух типов — со ступенчатым и плавным изменением показателя преломления. В них используются более технологичные сердечники, диаметр которых примерно на порядок больше, поэтому в центральном проводнике есть несколько путей распространения лучей света, под разными углами отражающихся от стеклянного покрытия. В качестве источника излучения света в таких кабелях используются более дешевые светодиодные излучатели. Полоса пропускания многомодовых кабелей гораздо уже за счет потерь света при отражениях. Следует отметить, что электрические помехи не оказывают влияния на передачу данных по оптоволоконному кабелю, которая ведется со скоростями до 100 Мбит/с.

К беспроводным линиям связи, являющимся открытыми средами передачи информации, относятся земная и водная поверхности, атмосфера и

космическое пространство. Распространение радиоволн, представляющих собой электромагнитные колебания, т.е. совокупность переменных электрических и магнитных полей, в таких средах свободно осуществляется в разных направлениях. С увеличением расстояния уровень электромагнитного сигнала, излучаемого источником, постоянно уменьшается; к тому же на него воздействуют естественные помехи.

На распространение радиоволн сильно влияют поверхность Земли и части ее атмосферы — тропосфера и ионосфера, нижняя и верхняя области соответственно. Траектория приземных радиоволн, распространяющихся вдоль земной поверхности и огибающих ее значительные неровности, повторяет профили граничных участков между поверхностью и атмосферой, поскольку эти две области имеют весьма разные электрические параметры. Пространственные волны распространяются в земной атмосфере и за ее пределами. Тропосфера (атмосферный слой высотой до 15 км) искривляет траекторию таких радиоволн и частично рассеивает их, причем электрические характеристики тропосферы сильно зависят от метеоусловий. Ионосфера (атмосферный слой высотой от 60 км до 20 тыс. км) оказывает на пространственные радиоволны такое же воздействие, а на ее электрические параметры существенно влияют время суток и времена года. Область за пределами ионосферы условно можно считать вакуумом, где радиоволны распространяются практически прямолинейно.

В зависимости от особенностей распространения в различных пространственных средах спектр радиоволн распределен на 8 диапазонов, представленных в *табл. 1.1*.

Данное распределение, в основу которого положен десятичный принцип классификации радиоволн, произведено в соответствии с рекомендациями Международного консультативного комитета по радио.

Высокочастотный (ВЧ) или коротковолновый (КВ) диапазон по международным соглашениям отводится специальным службам: мобильная связь, радиовещание, радионавигация, космическая связь и т.д. Волны этого диапазона распространяются (вдоль земной поверхности или отражаясь от ионосферы) на большие расстояния при меньшей мощности излучения, хотя они в значительной мере подвержены влиянию помех.

Ультракоротковолновые (УКВ) или радиорелейные системы передачи до недавнего времени пересылали значительную часть телевизионного и телефонного трафиков, однако сейчас дальнюю связь стали обеспечивать в основном оптоволоконные системы.

Таблица 1.1

Характеристики диапазонов радиоволн

Наименование волн	Условное обозначение	Диапазон частот	Границы диапазона
Мириаметровые	ОНЧ	3…30 Гц	10…100 км
Километровые	НЧ	30…300 Гц	1…10 км
Гектометровые	СЧ	300…3000 Гц	100…1000 м
Декаметровые	ВЧ, КВ	3…30 МГц	10…100 м
Метровые	ОВЧ	30…300 МГц	1…10 м
Дециметровые	УВЧ, УКВ	300…3000 МГц	10…100 см
Сантиметровые	СВЧ	3…30 ГГц	1…10 см
Миллиметровые	КВЧ	30…300 ГГц	1…10 мм

Сверхвысокочастотные (СВЧ) сигналы могут распространяться в свободном пространстве. Чтобы сигнал перемещался в нужном направлении, а внешние помехи и его потери при этом были минимальными, используются волноводы. Такие устройства представляют собой трубки со стальной оболочкой, в которые под давлением закачивается азот или сухой воздух. Это делается для уменьшения влажности, так как влажная среда заметно увеличивает затухание сигнала в СВЧ-диапазоне. Однако в настоящее время для передачи таких сигналов все больше применяются оптоволоконные кабели.

Спутниковые системы передачи пересылают сигналы нескольких типов: телевизионные, телефонные и высокоскоростные. Спутник является ретранслятором, принимая сигналы с наземных станций, усиливая их и отсылая обратно всем станциям в зоне его обслуживания. Коммерческий спутник вращается вокруг Земли с постоянной скоростью на высоте примерно 36 тыс. км на геостационарной орбите, благодаря чему он постоянно находится над одной и той же точкой поверхности. Это позволяет трем равномерно размещенным над экватором спутникам покрывать почти всю земную поверхность от 60° северной широты до 60° южной широты. Низкоорбитальные спутники, эксплуатация которых началась лет 10 назад, ретранслируют сигналы меньшей мощности и используются для обеспечения пейджинговой и международной сотовой связи.

Мобильная сотовая радиосвязь получила развитие в последние 15 лет. Площадь, например крупного города с пригородами, разбивается на ячей-

ки, называемые сотами, каждая из которых обслуживается своей базовой станцией. При перемещении абонента из одной соты в другую его мобильный телефон автоматически переключается на новую базовую станцию. Когда абонент звонит, его мобильный телефон занимает свободный канал в той соте, в которой он находится. Нужное соединение осуществляется через базовую станцию и центр коммутации, связывающий этого абонента с другим мобильным телефонным аппаратом через радиоканал и другую базовую станцию, в зоне обслуживания которой и находится вызываемый абонент. Кроме того, возможно соединение и через телефонную сеть общего пользования со стационарным телефонным аппаратом.

В зависимости от **типа передаваемого сигнала** линии связи подразделяются на аналоговые и цифровые.

Аналоговые линии связи предназначены для передачи аналоговых (непрерывных) сигналов, имеющих непрерывный диапазон значений, причем несколько низкоскоростных абонентских линий можно объединить в одном канале. Традиционная область применения таких линий — телефонные сети.

Цифровые линии связи используются для передачи цифровых (дискретных), сигналов, имеющих конечное число состояний. Кроме обмена компьютерными данными, по таким линиям могут передаваться оцифрованные звук и изображение.

С точки зрения **режимов передачи данных**, линии связи делятся на асинхронные и синхронные.

Асинхронный режим предполагает независимую передачу символов, причем каждый из них передается в свободном темпе и сопровождается служебными сигналами, указывающими на его начало и конец.

Синхронный режим реализует передачу информации блоками символов. Каждый блок передается непрерывно в принудительном темпе и сопровождается служебными кодовыми комбинациями, обеспечивающими синхронизацию передающего и принимающего устройств. Первый вариант передачи обладает существенной избыточностью информации из-за большого числа служебных сигналов. Второй вариант передачи более быстр, так как избыточность информации меньше, но аппаратура для его реализации сложнее.

По направлению передачи данных линии связи подразделяются на три типа:

— симплексные;

— полудуплексные;
— дуплексные.

Симплексные линии обеспечивают передачу информации только в одном направлении.

Полудуплексные линии передают данные в обоих направлениях, но в разные промежутки времени.

Дуплексные линии позволяют одновременно передавать информацию в обоих направлениях.

Наконец, с точки зрения **режимов взаимодействия абонентов** при установлении соединения между ними можно выделить две группы линий связи. Оперативный режим взаимодействия on-line, или «на линии», реализуется в том случае, когда обмен информацией между отправителем и получателем осуществляется без заметных для них временны́х задержек. Если же такие задержки при обмене информацией весьма велики, то реализуется режим взаимодействия off-line или «вне линии». Классификация рассмотренных типов линий связи представлена на *рис. 1.2*.

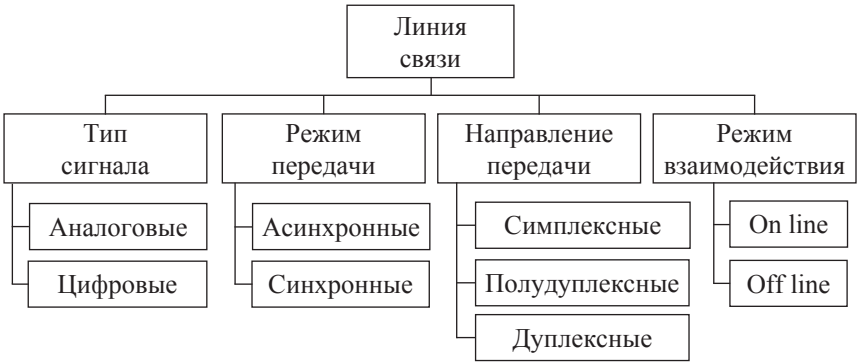


Рис. 1.2. Типы линий связи

1.4. Аппаратура каналов связи

Основными устройствами, участвующими в процессе распространения информации, в настоящее время, пожалуй, можно считать модемы (в том числе оптические). Кроме них передачу данных обеспечивают сетевые адаптеры, устройства подключения к цифровым каналам, а также проме-

жуточная аппаратура, в состав которой входят мультиплексоры, демультиплексоры, коммутаторы и повторители.

Модем осуществляет преобразование двоичных информационных сигналов для их передачи по аналоговым линиям связи (модуляцию) и обратное преобразование (демодуляцию) — при приеме информации; отсюда и название этих устройств. Дискретный канал связи, обеспечивающий передачу данных между двумя компьютерами, формируется так, как показано на *рис. 1.3*.

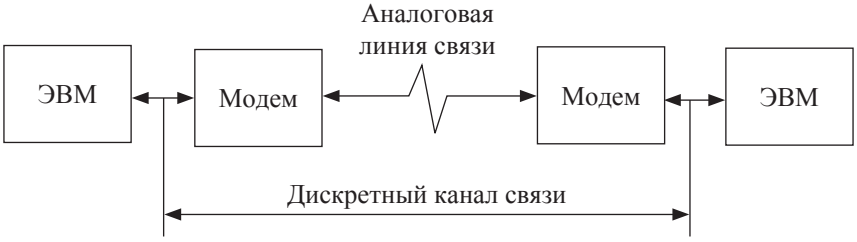


Рис. 1.3. Дискретный канал связи

Сетевой адаптер реализует функции сопряжения компьютера с каналом связи. Одноканальный адаптер связывает компьютер с одним каналом.

Мультиплексор передачи данных является многоканальным устройством сопряжения компьютера с несколькими каналами связи. Другими словами, он объединяет сообщения, поступившие от разных источников, в одном выходном канале, т.е. уплотняет принятые сообщения. Мультиплексоры использовались на начальных этапах построения вычислительных сетей, однако с укрупнением сетей и усложнением их структуры эти функции стали обеспечивать специальные процессоры.

Демультиплексор выполняет функции, обратные мультиплексированию сообщений.

Коммутатор наряду с мультиплексором и демультиплексором обеспечивает формирование составного канала связи между двумя абонентами в вычислительных сетях сложной конфигурации. Это комбинационная схема, коммутирующая любой вход со всеми выходами, причем различают два типа таких схем. Пространственный коммутатор, является матрицей $n \times n$ с n горизонтальными (входными) и n вертикальными (выходными) шинами.

В узлах матрицы располагаются коммутирующие или переключательные элементы, благодаря чему и обеспечивается соединение каждого входа хотя бы с одним выходом, причем только один элемент может быть открыт в каждой из выходных шин. Однако недостатком таких коммутаторов является большое количество переключательных элементов, равное n^2, поэтому используются также многоступенчатые коммутаторы с меньшим числом таких элементов, но более сложной структуры и с большей задержкой сигнала. Временной коммутатор содержит буферную память, в ячейки которой данные заносятся в результате последовательного опроса входов, а коммутация реализуется считыванием данных из требуемых ячеек на соответствующие выходы.

Повторитель применяется для промежуточного усиления передаваемых сигналов, когда протяженность линии связи больше, чем расстояние, на которое без искажений может передавать сигнал данная физическая среда. Рассмотренная промежуточная аппаратура используется как в аналоговых, так и в цифровых линиях для создания высокоскоростных составных каналов связи между абонентами в информационных сетях.

Контрольные вопросы

1. Какие элементы обеспечивают распространение информации?
2. Чем канал отличается от линии связи?
3. Что такое емкость канала связи?
4. Чем бод отличается от бит/с?
5. Чем определяется максимально возможная пропускная способность идеального канала связи?
6. Чем достоверность передачи информации отличается от ее надежности?
7. Как классифицируются линии связи?
8. Какие функции выполняет модем?
9. Что такое коммуникации и какие они бывают?
10. Что такое компьютерная коммуникационная среда?
11. Каковы основные характеристики кабелей?

2. КОМПЬЮТЕРНЫЕ СЕТИ

Под компьютерной сетью будем понимать такую коммуникационную среду, в которой продуктом обработки, хранения, передачи и использования является информация. Основным «рабочим органом» таких сетей с полным основанием можно считать компьютер, в связи с чем правомерно говорить о компьютерных коммуникационных сетях.

Термин «коммуникация» происходит от латинского слова *communicatio*, которое можно перевести как «передача», «связь», «сообщение», и подразумевает в общем случае не только процесс обмена любой информацией или энергией, но и средства перемещения реальных объектов. На *рис. 2.1* приведена классификация коммуникаций.

Как следует из рисунка, существует пять типов информационных коммуникаций. Аудио- и видеокоммуникации подразумевают передачу традиционными методами звука и изображения соответственно. Аудиторные

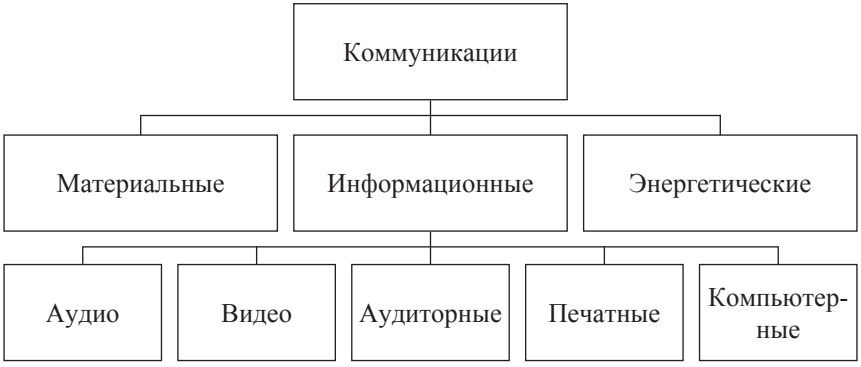


Рис. 2.1. Классификация коммуникаций

коммуникации — это лектории, театры, церкви и т.п. Печатные коммуникации связаны с распространением печатной продукции. Компьютерные коммуникации к настоящему времени превратились в универсальное средство общения. Следует отметить, что дистанционную передачу информации, в том числе и с помощью компьютеров, называют также телекоммуникациями (приставка «теле-» означает «далеко»). Предметом нашего рассмотрения будет именно этот тип коммуникаций.

2.1. Типы компьютерных сетей

Компьютерная коммуникационная среда — это совокупность правил и средств обмена информацией между людьми с помощью компьютеров. Компьютерная (или вычислительная) сеть представляет собой систему взаимосвязанных компьютеров, предназначенных для передачи, хранения и обработки информации. При наличии такой сети становятся доступными данные, хранящиеся в центральном компьютере, а также его ресурсы. Компонентами вычислительной сети кроме компьютеров могут быть и разнообразные периферийные устройства, играющие роль источников и получателей информации, передаваемой по сети.

Приведем определения основных сетевых терминов, которые будем использовать в дальнейшем изложении. Сервер является главным компьютером сети, который предоставляет доступ к общей базе данных, а также обеспечивает взаимодействие пользователей и совместное использование ими устройств ввода-вывода. Клиентом сети является ЭВМ, имеющая доступ к ресурсам сервера. Чтобы обеспечить эту возможность, каждая такая машина должна быть занесена в список клиентов сервера, в результате чего ей выделяются регистрационное имя и пароль. Узлом или абонентом сети является ее объект, обеспечивающий генерацию или потребление информации. Протокол связи или обмена информацией устанавливает единые правила представления, передачи и восприятия данных для обеспечения процесса информационного обмена. Пакетом будем называть информационную последовательность байтов, предназначенную для передачи по сети.

Классификация компьютерных сетей, проведенная по нескольким признакам, представлена на *рис. 2.2*.

В зависимости от территориального расположения узлов сети, т.е. расстояния между ними, сети можно разделить на локальные (местные, корпоративные), региональные (территориальные) и глобальные сети.

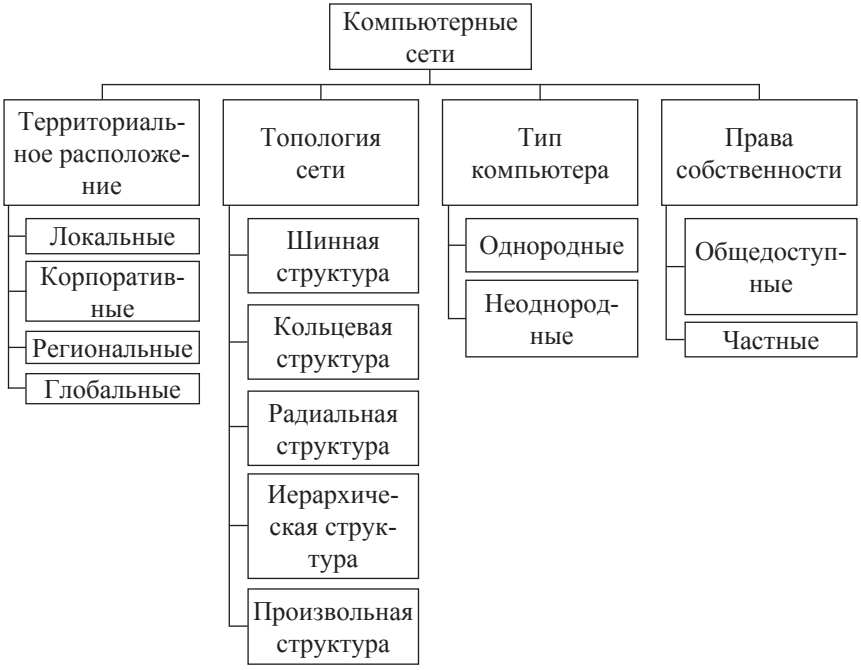


Рис. 2.2. Классификация компьютерных сетей

Локальная вычислительная сеть (ЛВС, по-английски LAN — Local Area Network) объединяет абонентов, расположенных на относительно небольшом удалении друг от друга (обычно до километра). Корпоративная сеть объединяет связанные между собой ЛВС, размещенные в одном или нескольких рядом расположенных зданиях; обычно такая сеть организуется в пределах одного предприятия или учреждения. Региональная сеть (MAN — Metropolitan Area Network) связывает абонентов, расположенных в десятках километров друг от друга, обычно в пределах большого города или определенной территории. Глобальная сеть (WAN — Wide Area Network) объединяет абонентов разных стран и континентов. Наиболее известная сеть данного класса — Интернет — является телекоммуникационной информационной глобальной мегасетью, охватывающей свыше ста стран. Подобные сети позволят объединить информационные ресурсы человечества и обеспечить доступ к ним.

Топология сети определяет геометрическую схему расположения узлов сети и связей между ними. С точки зрения их топологии различают сети шинной (магистральной), кольцевой, звездообразной, иерархической и произвольной структуры; первые три из них наиболее характерны для ЛВС.

Шинная структура, представленная на *рис. 2.3*, является, пожалуй, самой простой. Данные от любого узла сети распространяются в обе стороны

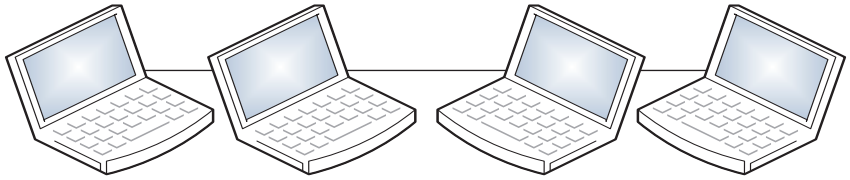


Рис. 2.3. Шинная структура ЛВС

по шине, в качестве которой используется коаксиальный кабель, и становятся доступными всем абонентам. Однако принимает сообщение только тот, кому оно адресовано. Такую сеть легко наращивать, она нечувствительна к неисправностям отдельных узлов, обладает высоким быстродействием не-смотря на сравнительно малую протяженность.

Кольцевая структура, представленная на *рис. 2.4*, обеспечивает соединение всех узлов сети замкнутой кривой, причем выход одного узла соединен со входом следующего. Информация передается по кольцу и поочередно становится доступной всем абонентам, но распознает и получает ее только тот, кому она адресована. Такая структура эффективна для сетей, располагающихся на небольшом пространстве, однако ее быстродействие и надежность работы ниже. Ведь потеря работоспособности любого узла разрывает передающее кольцо, поэтому необходим комплекс защитных мер для сохранения передающей линии.

Звездообразная структура, показанная на *рис. 2.5*, предполагает наличие центрального узла, от которого линии связи идут к каждому из остальных узлов. Таким образом, все периферийные узлы связываются друг с другом именно через центральный узел, работоспособность которого и определяет надежность сети. Такая структура упрощает взаимодействие

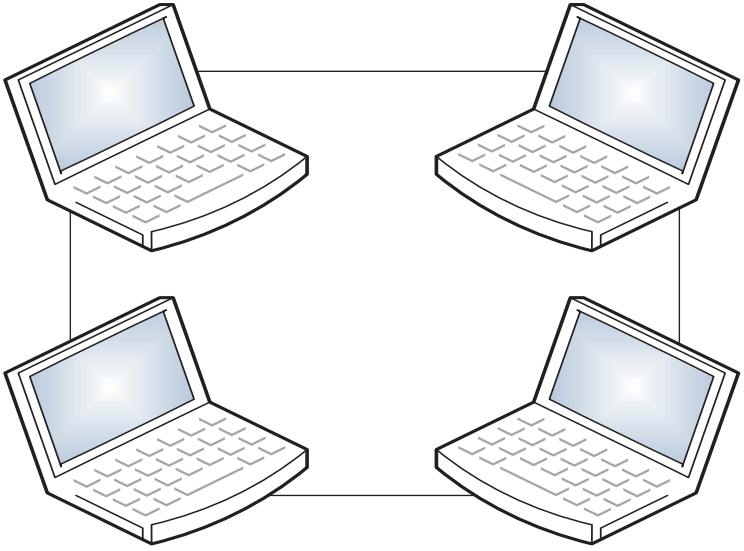


Рис. 2.3. Шинная структура ЛВС

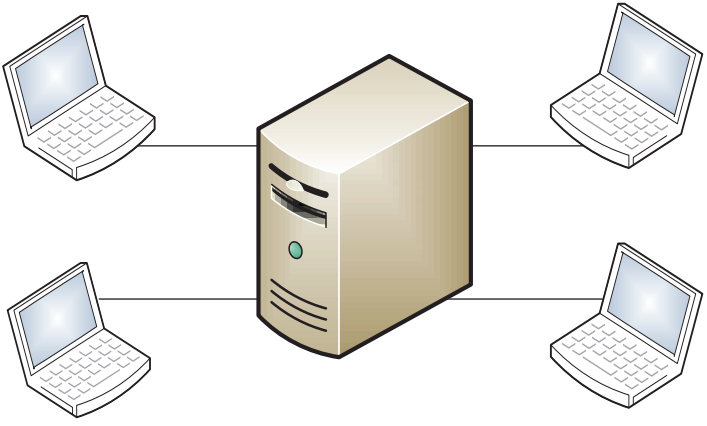


Рис. 2.5. Звездообразная структура ЛВС

периферийных узлов, что позволяет использовать более простые и дешевые сетевые адаптеры.

Иерархическая структура, представленная на *рис. 2.6*, в некотором смысле может считаться разновидностью звездообразной структуры, поскольку только через узел верхнего уровня осуществляется связь между узлами следующего уровня иерархии. Это утверждение справедливо в отношении всех узлов такой структуры.

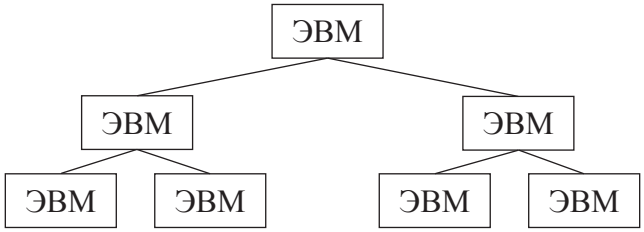


Рис. 2.6. Иерархическая структура сети

Рассмотренные топологии применяются при реализации сетей с относительно небольшим числом компьютеров. Более крупные сети могут быть произвольной структуры, отдельные фрагменты которой представляют собой приведенные базовые топологии или их сочетания. В качестве примера на *рис. 2.7* представлена сеть с топологией «звезда — шина».

В зависимости от типа используемых в сети компьютеров различают однородные и неоднородные сети. Очевидно, что первые из них содержат только однотипные ЭВМ, а вторые (как правило, значительно более крупные) — разнотипные компьютеры.

С точки зрения возможностей доступа к ресурсам сети или прав собственности на них выделяют общедоступные и частные сети. Необходимо отметить, что представленная на *рис. 2.2* схема классификации компьютерных сетей не является полной. Признаками различия сетей служат также применяемые в них протоколы обмена и способы коммутации данных. Однако эти более сложные понятия, связанные с функционированием компьютерных сетей, будут рассмотрены ниже.

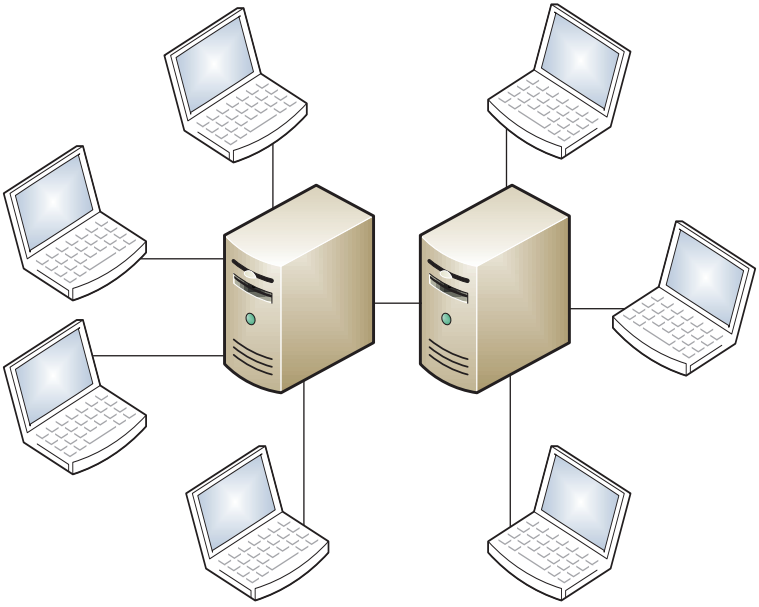


Рис. 2.7. Структура сети «звезда — шина»

2.2. Функционирование компьютерных сетей

При работе в сети абонент располагает разнообразными возможностями. Он может в диалоговом режиме обмениваться сообщениями с другими абонентами. Имея на своем компьютере электронный почтовый ящик, он может пользоваться услугами электронной почты. Обладая соответствующими правами доступа, абонент может работать с удаленной базой данных или удаленным файлом. Наконец, установив логическую связь с процессом, протекающим на другой ЭВМ, пользователь также может провести с ним сеанс связи.

В то же время компьютерные сети и программные продукты к ним производят многие фирмы, поэтому возникла задача объединения различных сетей. Для ее решения была принята концепция открытой системы и для таких систем была разработана эталонная модель обмена информацией. Под

открытой системой понимается не замкнутая на себя система, способная взаимодействовать по принятым правилам с другими системами. Эталонная модель обмена представляет общие рекомендации, которые должны быть реализованы производителями как сетевой аппаратуры, так и сетевых программных средств, чтобы проблема совместимости компьютерных сетей была разрешена. Однако следует отметить, что предложенная модель рассматривает только общие функции, а не конкретные решения, обладает большой избыточностью. Поэтому далеко не все конкретные сети абсолютно точно ей соответствуют.

Международная организация по стандартам (International Standards Organization, ISO), созданная в 1946 году, разработала универсальную модель взаимодействия открытых систем (Open System Interconnection, OSI), или модель ISO/OSI. Эталонная модель взаимосвязи открытых систем (семиуровневая модель OSI) введена в 1977 г.

Уровни эталонной модели OSI представляют из себя вертикальную структуру, где все сетевые функции разделены между семью уровнями. Следует особо отметить, что каждому такому уровню соответствует строго описанные операции, оборудование и протоколы.

Взаимодействие между уровнями организовано следующим образом:

- по вертикали — внутри отдельно взятой ЭВМ и только с соседними уровнями;
- по горизонтали — организовано логическое взаимодействие — с таким же уровнем другого компьютера на другом конце канала связи (то есть сетевой уровень на одном компьютере взаимодействует с сетевым уровнем на другом компьютере).

Согласно этой модели, архитектура которой представлена на *рис. 2.8*, все сетевые функции разделены на семь уровней. Нижние уровни выполняют более простые функции и предоставляют услуги вышестоящим, которые управляют ими и реализуют более сложные функции. Каждый уровень должен взаимодействовать только с двумя уровнями, расположенными выше и ниже его (с одним верхним или одним нижним — крайние уровни). При работе в сети каждым абонентом реализуются все функции. Непосредственная связь между ними устанавливается только на нижнем уровне, а остальные одноименные уровни абонентов имеют лишь виртуальную связь. При организации обмена информацией она проходит все уровни сверху вниз у передатчика и в обратном направлении (снизу вверх) — у приемника (см. *рис. 2.9*).

Рис. 2.8. Эталонная модель архитектуры открытой системы

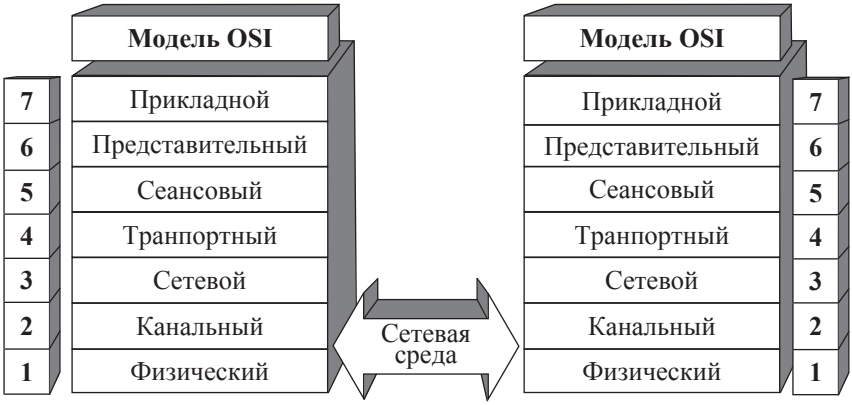


Рис. 2.9. Взаимодействие компьютеров

Первый уровень — физический — обеспечивает выполнение всех действий в канале связи. Именно здесь информация представляется в виде передаваемых сигналов, т.е. осуществляется управление аппаратурой передачи данных для реализации функции передачи через конкретную физическую среду.

Второй уровень — канальный — осуществляет передачу по информационному каналу, под которым понимается логический канал, формируемый между двумя компьютерами, соединенными физическим каналом связи. Этот уровень осуществляет формирование и передачу кадров — пакетов канального уровня, а также обнаруживает и исправляет ошибки, которые возникают при передаче информации на физическом уровне. В локальных компьютерных сетях канальный уровень разделен на два подуровня: первый управляет доступом к каналу передачи, второй — логическим каналом.

Третий уровень — сетевой — обеспечивает межсетевое взаимодействие и маршрутизацию пакетов, т.е. определяет и реализует маршруты, по которым они передаются через промежуточные компоненты сети. Фактически маршрутизация сводится к формированию логического канала между сетевыми объектами, при наличии которого возможно их взаимодействие.

Четвертый уровень — транспортный — осуществляет связь между оконечными пунктами сетевого обмена информацией. На этом уровне производится мультиплексирование и демультиплексирование пакетов, которые на транспортном уровне называются сегментами. Логические каналы, образуемые на данном этапе, по которым передаются информационные пакеты, называются транспортными каналами.

Пятый уровень — сеансовый — организует сеансы связи, ведущиеся в диалоговом режиме между абонентами сети. При этом определяется тип устанавливаемой связи (дуплексная или полудуплексная), а также режим обмена запросами и ответами между участниками диалога.

Шестой уровень — представительный — определяет формат представления передаваемых данных, принятый в используемой системе, причем эти данные при необходимости переводятся из одного кода в другой.

Седьмой уровень — прикладной — осуществляет управление прикладными процессами, которые будут реализованы в сетевом окружении. При этом данные, которыми будут обмениваться процессы, формируются в блоки, и также обеспечивается сервис прикладных пользовательских программ.

Данные, поступающие в сеть от внешнего процесса и предназначенные для передачи, подвергаются обработке на всех уровнях рассмотренной модели. При этом каждый уровень, кроме физического, добавляет к ним свой заголовок, содержащий служебную информацию, необходимую для правильной адресации пришедших сообщений и выполнения контрольных функций (*рис. 2.10*).

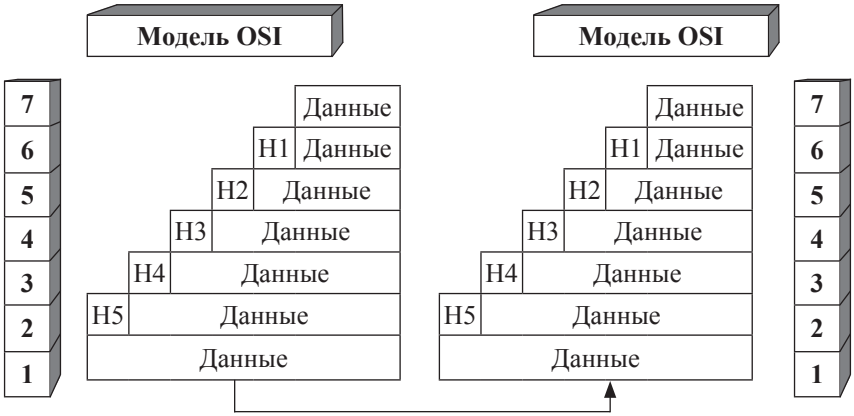


Рис. 2.10. Процесс инкапсуляции

Например, сообщение, поступившее на транспортный уровень, делится на сегменты со своими заголовками, которые передаются на следующий сетевой уровень. Здесь эти сегменты делятся на пакеты уже со своими заголовками, а затем на канальном уровне может осуществляться разбиение пакетов на кадры. Преобразованное таким образом сообщение со всеми заголовками передается в компьютерную сеть, абоненты которой являются его адресатами.

В приемном узле сети производится обратная последовательность действий, в результате чего исходное сообщение восстанавливается. Уровни эталонной модели последовательно считывают и удаляют соответствующие заголовки, причем нижние уровни игнорируют заголовки верхних уровней. В итоге, поднявшись до верхнего прикладного уровня, сообщение поступает тому процессу, которому оно и было предназначено.

В стандартах ISO для обозначения той или иной порции данных, с которыми работают протоколы разных уровней модели OSI, используется

общее название — протокольный блок данных (Protocol Data Unit, PDU). Для обозначения блоков данных определенных уровней часто используются специальные названия: кадр (frame), пакет (packet), сегмент (segment).

Функции, описываемые на первом физическом уровне, реализуются аппаратными средствами — адаптерами, мультиплексорами и т.д. Функции остальных уровней реализуются программными модулями — драйверами.

Взаимодействие между одноименными уровнями модели осуществляется абонентами сети по установленным соглашениям, совокупность которых является протоколом.

Такие протоколы фактически определяют дисциплину обслуживания абонентов при их взаимодействии и реализуются программным путем в виде драйверов компьютерных сетей.

Существует несколько основных стеков коммуникационных протоколов, разработанных различными фирмами и международными организациями. Под стеком здесь понимается набор иерархически организованных протоколов, достаточный для взаимодействия абонентов на всех уровнях эталонной модели. Приведем некоторые наиболее распространенные в настоящее время протоколы.

На физическом уровне используется протокол X.21. На канальном уровне поддерживаются протоколы Ethernet, Token Ring, FDDI, X.25 и др. Из протоколов сетевого уровня следует выделить протокол IP (Internet Protocol), а транспортного уровня — протокол TCP (Transmission Control Protocol — протокол управления передачей). Среди протоколов верхних уровней иерархии отметим протоколы Telnet, WWW, SMTP (Simple Mail Transfer Protocol — простой протокол пересылки почты), X.400 (электронная почта), X.500 (справочная служба), FTAM (File Transfer, Access and Management — передача файлов, доступ к ним и управление файлами). Наконец, существует группа стандартов IEEE.802.1—802.12, часть которых разработана для локальных вычислительных сетей и касается канального уровня эталонной модели, а точнее — непосредственно подуровня управления доступом к каналу передачи.

2.3. Физический уровень (Physical Layer)

Физический уровень предназначен для сопряжения с физическими средствами соединения.

Физические средства соединения — это совокупность физической среды, аппаратных и программных средств, обеспечивающая передачу сигналов между системами.

Физическая среда — это материальная субстанция, через которую осуществляется передача сигналов. Физическая среда является основой, на которой строятся физические средства соединения. В качестве физической среды широко используются эфир, металлы, оптическое стекло и кварц.

Физический уровень подразделяется на два подуровня:

- подуровень стыковки со средой;
- подуровень преобразования передачи.

Подуровень стыковки со средой обеспечивает сопряжение потока данных с используемым физическим каналом связи.

Подуровень преобразования передачи осуществляет преобразования, связанные с применяемыми протоколами.

Физический уровень обеспечивает физический интерфейс с каналом передачи данных, а также описывает процедуры передачи сигналов в канал и получения их из канала. На этом уровне определяются электрические, механические, функциональные и процедурные параметры для физической связи в системах. Физический уровень получает пакеты данных от вышележащего канального уровня и преобразует их в оптические или электрические сигналы, соответствующие 0 и 1 бинарного потока. Эти сигналы посылаются через среду передачи на приемный узел. Механические и электрические/оптические свойства среды передачи определяются на физическом уровне и включают:

- тип кабелей и разъемов;
- разводку контактов в разъемах;
- схему кодирования сигналов для значений 0 и 1.

Физический уровень выполняет следующие функции:

1) Установление и разъединение физических соединений;
2) Передача сигналов в последовательном коде и прием;
3) Прослушивание (в нужных случаях) каналов;
4) Идентификация каналов;
5) Оповещение о появлении неисправностей и отказов.

Оповещение о появлении неисправностей и отказов связано с тем, что на физическом уровне происходит обнаружение определенного класса событий, мешающих нормальной работе сети (столкновение кадров, посланных сразу несколькими системами, обрыв канала, отключение пита-

ния, потеря механического контакта и т.д.). Виды сервиса, предоставляемого канальному уровню, определяются протоколами физического уровня. В случаях, когда к одному каналу подключается группа систем, производится прослушивание канала, которое позволяет определить, свободен ли он для передачи.

Функции физического уровня реализуются во всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером. Повторители являются единственным типом оборудования, которое работает только на физическом уровне.

В глобальных сетях на физическом уровне могут использоваться модемы и интерфейс RS-232C. В локальных сетях для преобразования данных применяют сетевые адаптеры, обеспечивающие скоростную передачу данных в цифровой форме. Пример протокола физического уровня — это интерфейс RS-232C/CCITT V.2, который является наиболее широко распространенной стандартной последовательной связью между компьютерами и периферийными устройствами.

Физический уровень может обеспечивать как асинхронную (последовательную) так и синхронную (параллельную) передачу. На физическом уровне определяется схема кодирования для представления двоичных значений с целью их передачи по каналу связи.

Примером протокола физического уровня может служить спецификация 10 Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для представления данных на кабеле и другие характеристики среды и электрических сигналов.

2.4. Канальный уровень (Data Link)

Канальный уровень определяет доступ к среде и управление передачей посредством процедуры передачи данных по каналу.

Единицей информации канального уровня являются кадры (frame). *Кадры* — это логически организованная структура, в которую можно помещать данные. Задача канального уровня передавать кадры от сетевого уровня к физическому уровню.

Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит, в начало и конец каж-

дого кадра, чтобы отметить его, а также вычисляет контрольную сумму, суммируя все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка.

Задача канального уровня — преобразование пакетов, поступающих с сетевого уровня, подготовка их к передаче, преобразование в кадр соответствующего размера. Этот уровень также обязан обнаруживать ошибки передачи. На канальном уровне определяются правила использования физического уровня узлами сети.

Канальный уровень обеспечивает создание, передачу и прием кадров данных. Этот уровень обслуживает запросы сетевого уровня и использует сервис физического уровня для приема и передачи пакетов (*рис. 2.11*).

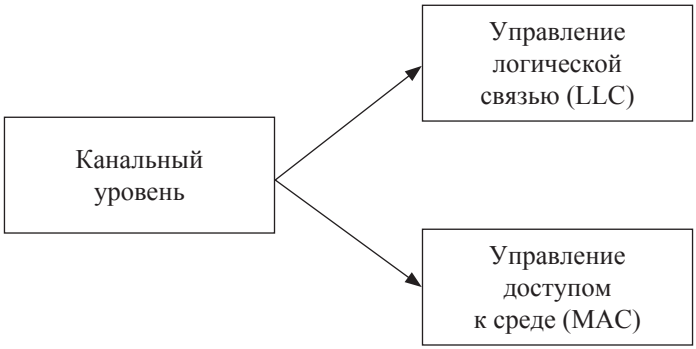


Рис. 2.11. Канальный уровень (Data Link)

При больших размерах передаваемых блоков данных канальный уровень делит их на кадры и передает кадры в виде последовательностей. При получении кадров уровень формирует из них переданные блоки данных. Размер блока данных зависит от способа передачи, качества канала, по которому он передается (*рис. 2.12*).

Канальный уровень может выполнять несколько функций.

1. Организация (установление, управление, расторжение) канальных соединений и идентификация их портов.

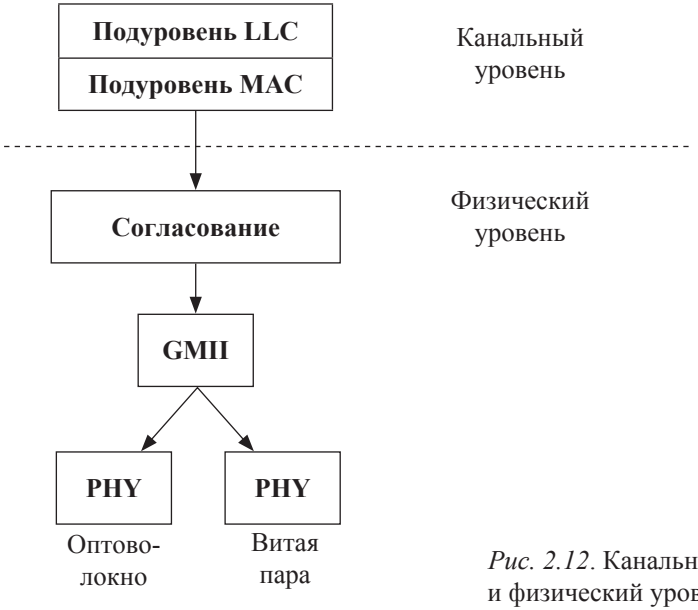


Рис. 2.12. Канальный и физический уровни

2. Организация и передача кадров.
3. Обнаружение и исправление ошибок.
4. Управление потоками данных.
5. Обеспечение прозрачности логических каналов (передачи по ним данных, закодированных любым способом).

Для ЛВС канальный уровень разбивается на два подуровня (*рис. 2.13*):

- LLC (LogicalLinkControl) — отвечает за установление канала связи и за безошибочную посылку и прием сообщений данных;
- MAC (MediaAccessControl) — обеспечивает совместный доступ сетевых адаптеров к физическому уровню, определение границ кадров, распознавание адресов назначения (например, доступ к общей шине).

На канальном уровне обнаруживают ошибки в передаваемых данных.

Взаимодействие узлов локальных сетей происходит на основе протоколов канального уровня.

Международным институтом инженеров по электротехнике и радиоэлектронике *(Institute* of *Electrical and Electronics Engineers — IEEE*) разработано семейство стандартов 802.x, которое регламентирует функци-

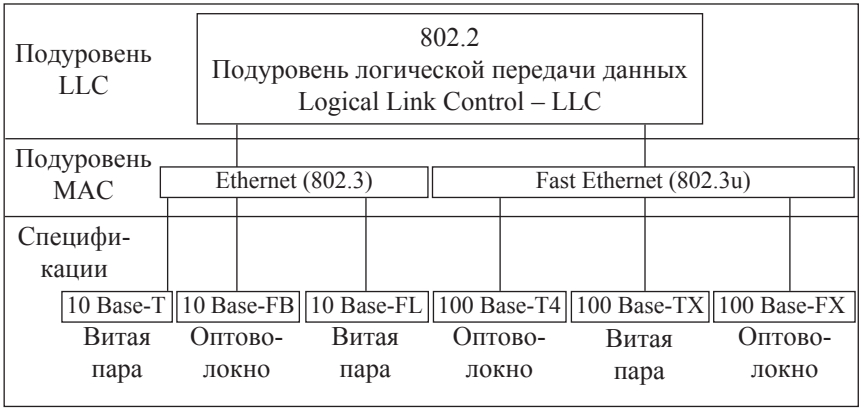


Рис. 2.13. Подуровни канального уровня

онирование канального и физического уровней семиуровневой модели ISO/OSI.

На подуровне LLC существует несколько процедур, которые позволяют устанавливать или не устанавливать связь перед передачей кадров, содержащих данные, восстанавливать или не восстанавливать кадры при их потере или обнаружении ошибок. Этот подуровень реализует связь с протоколами сетевого уровня.

Связь с сетевым уровнем и определение логических процедур передачи кадров по сети реализует протокол 802.2. Протокол 802.1 дает общие определения локальных вычислительных сетей, связь с моделью ISO/OSI. Существуют также модификации этого протокола.

Подуровень MAC определяет особенности доступа к физической среде при использовании различных технологий локальных сетей. Протоколы MAC-уровня ориентированы на совместное использование физической среды абонентами.

Разделяемая среда (shared media) применяется в таких широко распространенных в локальных сетях технологиях, как Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring, FDDI. Использование разделяемой между пользователями среды улучшает загрузку канала связи, удешевляет сеть, но ограничивает скорость передачи данных между двумя узлами.

Каждой технологии MAC-уровня соответствует несколько вариантов (спецификаций) протоколов физического уровня.

Спецификация технологии MAC-уровня определяет среду физического уровня и основные параметры передачи данных (*скорость передачи, вид среды, узкополосная или широкополосная*).

Так, протоколу 802.3, описывающему технологию Ethernet, соответствуют спецификации физического уровня: 10 Base-T, 10 Base-FB, 10 Base-FL. Число 10 показывает, что скорость передачи данных составляет 10 Мбит/с, Base — система узкополосная. Спецификация 10 Base-T предусматривает построение локальной сети на основе использования неэкранированной витой пары UTP не ниже 3-й категории и концентратора. Спецификации 10 Base-FB, 10 Base-FL используют волоконно-оптические кабели. Более ранние спецификации 10 Base-5 и 10 Base-2 предусматривали использование «толстого» или «тонкого» коаксиального кабеля.

Формат кадра *LLC*. Этот формат представлен на *рис. 2.14*.

Флаг	DSAP	SSAP	Control	Data	Флаг
01111110	1 байт	1 байт	1–2 байта	46–1497 байт	01111110

Рис. 2.14. Формат кадра *LLC*

Флаги определяют границы кадра *LLC*.

В поле данных (Data) размещаются пакеты сетевых протоколов.

Поле адреса точки входа службы назначения (DSAP — Destination Service Access Point) и адреса точки входа службы источника (SSAP — Source Service Access Point) длиной по 1 байту адресуют службу верхнего уровня, которая передает и принимает пакет данных.

Например, служба IP имеет значение SAP, равное 0x6. Обычно это одинаковые адреса. Адреса DSAP и SSAP могут различаться только в том случае, если служба имеет несколько адресов точек входа. Таким образом, адреса DSAP и SSAP не являются адресами узла назначения и узла источника.

Поле управления (Control) имеет длину 1 или 2 байта в зависимости от того, какой тип кадра передается: информационный (Information), управляющий (Supervisory), ненумерованный (Unnumbered). У первых двух *длина* поля Control составляет 2 байта, у ненумерованного — 1 байт.

Тип кадра определяется процедурой управления логическим каналом LLC. Стандартом 802.2 предусмотрено 3 типа таких процедур:

- LLC1 — процедура без установления соединения и подтверждения;

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

Процедура LLC1 применяется при дейтаграммном режиме передачи данных. Для передачи данных используются ненумерованные кадры. Восстановление принятых с ошибками данных производят протоколы верхних уровней, например, протокол транспортного уровня. В дейтаграммном режиме функционирует, например, протокол IP.

Процедура LLC2 перед началом передачи данных устанавливает соединение, послав соответствующий запрос и получив подтверждение, после чего передаются данные. Процедура позволяет восстанавливать потерянные и исправлять ошибочные данные, используя режим скользящего окна. Для этих целей она использует все три типа кадров (информационные, управляющие, ненумерованные). Данная процедура более сложная и менее быстродействующая по сравнению с LLC1, поэтому она применяется в локальных сетях значительно реже, чем LLC1, например, протоколом NetBIOS/NetBEUI.

Широкое применение процедура, подобная LLC2, получила в глобальных сетях для надежной передачи данных по ненадежным линиям связи. Например, она используется в протоколе LAP-B сетей X.25, в протоколе LAP-D сетей ISDN, в протоколе LAP-M сетей с модемами, частично — в протоколе LAP-F сетей Frame Relay.

Процедура LLC3 задействуется в системах управления технологическими процессами, когда необходимо высокое быстродействие и знание того, дошла ли управляющая информация до объекта.

Наиболее широкое распространение в локальных сетях получила процедура LLC1, в которой используются только ненумерованные типы кадров.

На передающей стороне кадр LLC-уровня передается на MAC-уровень, где инкапсулируется в кадр соответствующей технологии данного уровня. При этом флаги кадра LLC отбрасываются.

2.5. Сетевой уровень (Network Layer)

Сетевой уровень обеспечивает прокладку каналов, соединяющих абонентские и административные системы через коммуникационную сеть, выбор маршрута наиболее быстрого и надежного пути.

Сетевой уровень устанавливает связь в вычислительной сети между двумя системами и обеспечивает прокладку виртуальных каналов между ними.

Виртуальный или логический канал — это такое функционирование компонентов сети, которое создает взаимодействующим компонентам иллюзию прокладки между ними нужного тракта. Кроме этого, сетевой уровень сообщает транспортному уровню о появляющихся ошибках. Сообщения сетевого уровня принято называть *пакетами* (packet). В них помещаются фрагменты данных. Сетевой уровень отвечает за их адресацию и доставку.

Маршрутизация (routing) — процесс определения в коммуникационной сети наилучшего пути, по которому пакет может достигнуть адресата, точнее, набор правил, определяющих маршрут следования информации в сетях связи. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени. Некоторые алгоритмы маршрутизации пытаются приспособиться к изменению нагрузки, в то время как другие принимают решения на основе средних показателей за длительное время. Выбор маршрута может осуществляться и по другим критериям, например, надежности передачи.

Протокол канального уровня обеспечивает доставку данных между любыми узлами только в сети с соответствующей *типовой топологией*. Таким образом, канальный уровень регулирует доставку данных внутри сети, осуществление передачи данных между узлами производится на сетевом уровне. При организации доставки пакетов на сетевом уровне используется понятие *номер сети.* В этом случае *адрес* получателя состоит из *номера сети* и *номера компьютера* в этой сети.

Сети соединяются между собой специальными устройствами, называемыми маршрутизаторами (*рис. 2.15*).

Маршрутизатор — устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. Для того чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач (hops) между сетями, каждый раз выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, по которым проходит пакет.

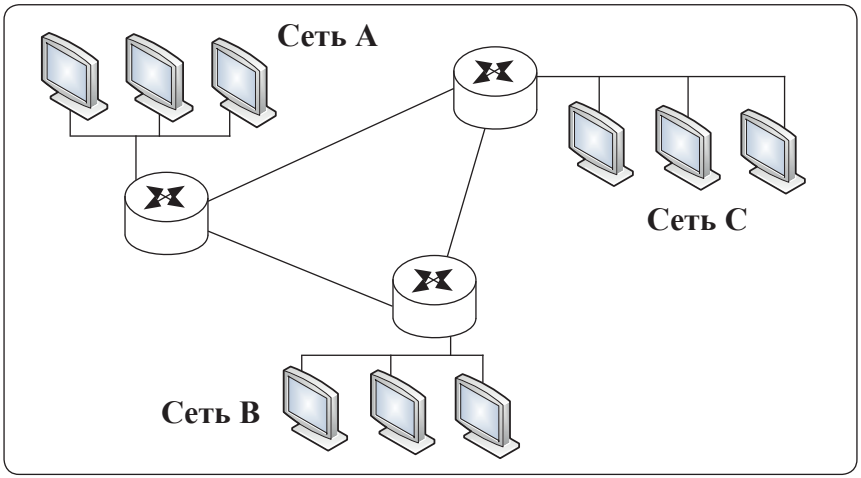


Рис. 2.15. Маршрутизация сети

Сетевой уровень отвечает за деление пользователей на группы и маршрутизацию пакетов на основе преобразования MAC-адресов в сетевые адреса. Сетевой уровень обеспечивает также прозрачную передачу пакетов на транспортный уровень.

Сетевой уровень выполняет ряд функций.

1. Создание сетевых соединений и идентификация их портов.
2. Обнаружение и исправление ошибок, возникающих при передаче через коммуникационную сеть.
3. Управление потоками пакетов.
4. Организация (упорядочение) последовательностей пакетов.
5. Маршрутизация и коммутация.
6. Сегментирование и объединение пакетов.

На сетевом уровне определяется два вида протоколов. Первый вид относится к определению *правил передачи пакетов* с данными конечных узлов от узла к маршрутизатору и между маршрутизаторами. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых *протоколами обмена маршрутной информацией*. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений.

Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов.

Виды протоколов сетевого уровня:

- сетевые протоколы (продвижение пакетов через сеть: IP, ICMP);
- протоколы маршрутизации: RIP, OSPF;
- протоколы разрешения адресов (ARP).

2.6. Транспортный уровень (Transport Layer)

Транспортный уровень предназначен для передачи пакетов через коммуникационную сеть. На транспортном уровне пакеты разбиваются на блоки (*рис. 2.16*).

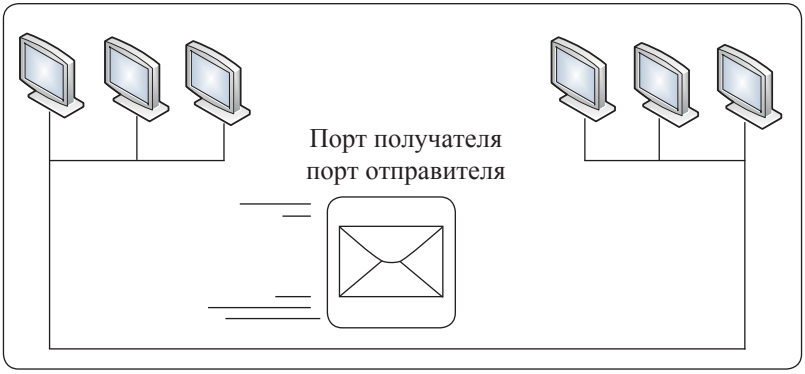


Рис. 2.16. Взаимодействие на транспортном уровне

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Работа транспортного уровня заключается в том, чтобы обеспечить приложениям или верхним уровням модели (прикладному и сеансовому) передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления

прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное — способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

Транспортный уровень определяет адресацию физических устройств (систем, их частей) в сети. Этот уровень гарантирует доставку блоков информации адресатам и управляет этой доставкой. Его главной задачей является обеспечение эффективных, удобных и надежных форм передачи информации между системами. Когда в процессе обработки находится более одного пакета, транспортный уровень контролирует очередность прохождения пакетов. Если проходит дубликат принятого ранее сообщения, то данный уровень опознает это и игнорирует сообщение.

В функции транспортного уровня входят:

1) управление передачей по сети и обеспечение целостности блоков данных;
2) обнаружение ошибок, частичная их ликвидация и сообщение о неисправленных ошибках;
3) восстановление передачи после отказов и неисправностей;
4) укрупнение или разделение блоков данных;
5) предоставление приоритетов при передаче блоков (нормальная или срочная);
6) подтверждение передачи;
7) ликвидация блоков при тупиковых ситуациях в сети.

Начиная с транспортного уровня, все вышележащие протоколы реализуются программными средствами, обычно включаемыми в состав сетевой операционной системы.

Наиболее распространенные протоколы транспортного уровня включают в себя:

- TCP (Transmission Control Protocol) — протокол управления передачей стека TCP/IP;
- UDP (User Datagram Protocol) — пользовательский протокол дейтаграмм стека TCP/IP;
- NCP (NetWare Core Protocol) — базовый протокол сетей NetWare;
- SPX (Sequenced Packet eXchange) — упорядоченный обмен пакетами стека Novell;
- TP4 (Transmission Protocol) — протокол передачи класса 4.

2.7. Сеансовый уровень (Session Layer)

Сеансовый уровень — это уровень, определяющий процедуру проведения сеансов между пользователями или прикладными процессами.

Сеансовый уровень обеспечивает управление диалогом для того, чтобы фиксировать, какая из сторон является активной в настоящий момент, а также предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, вместо того чтобы начинать все сначала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется.

Сеансовый уровень управляет передачей информации между прикладными процессами, координирует прием, передачу и выдачу одного сеанса связи. Кроме того, сеансовый уровень содержит дополнительно функции управления паролями, управления диалогом, синхронизации и отмены связи в сеансе передачи после сбоя вследствие ошибок в нижерасположенных уровнях. Функции этого уровня состоят *в координации связи* между двумя прикладными программами, работающими на разных рабочих станциях. Это происходит в виде хорошо структурированного диалога. В число этих функций входит создание сеанса, управление передачей и приемом пакетов сообщений во время сеанса и завершение сеанса (*рис. 2.17*).

На сеансовом уровне определяется, какой будет передача между двумя прикладными процессами:

- *полудуплексной* (процессы будут передавать и принимать данные по очереди);
- *дуплексной* (процессы будут передавать данные и принимать их одновременно).

В полудуплексном режиме сеансовый уровень выдает тому процессу, который начинает передачу, *маркер данных*. Когда второму процессу приходит время отвечать, маркер данных передается ему. Сеансовый уровень разрешает передачу только той стороне, которая обладает маркером данных.

Сеансовый уровень обеспечивает выполнение функций.

1. Установление и завершение на сеансовом уровне соединения между взаимодействующими системами.
2. Выполнение нормального и срочного обмена данными между прикладными процессами.

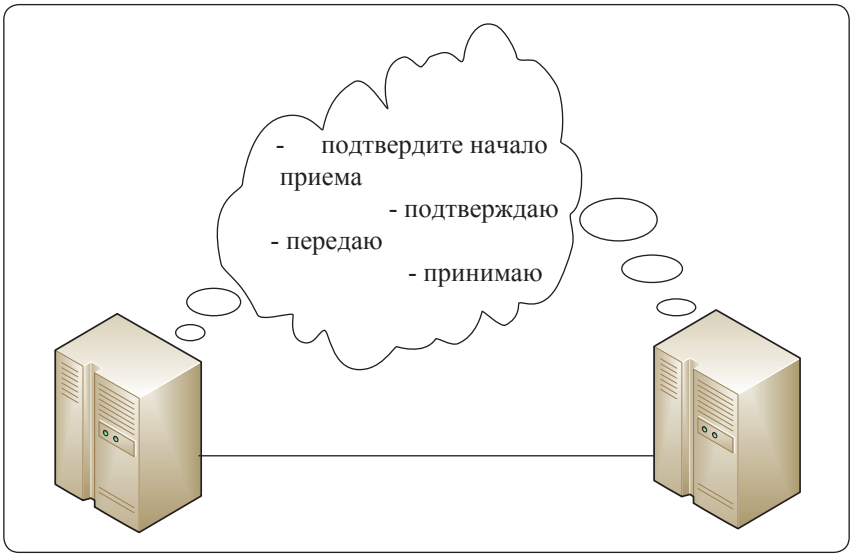


Рис. 2.17. Сеансовый уровень

3. Управление взаимодействием прикладных процессов.
4. Синхронизация сеансовых соединений.
5. Извещение прикладных процессов об исключительных ситуациях.
6. Установление в прикладном процессе меток, позволяющих после отказа либо ошибки восстановить его выполнение от ближайшей метки.
7. Прерывание в нужных случаях прикладного процесса и его корректное возобновление.
8. Прекращение сеанса без потери данных.
9. Передача особых сообщений о ходе проведения сеанса.

Сеансовый уровень отвечает за организацию сеансов обмена данными между оконечными машинами. Протоколы сеансового уровня обычно являются составной частью протоколов трех верхних уровней модели.

2.8. Уровень представления данных (Presentation Layer)

Уровень представления данных, или представительский уровень, представляет данные, передаваемые между прикладными процессами, в нужной форме.

Этот уровень обеспечивает то, что информация, передаваемая прикладным уровнем, будет понятна прикладному уровню в другой системе. В случаях необходимости уровень представления в момент передачи информации выполняет преобразование форматов данных в некоторый общий формат представления, а в момент приема соответственно выполняет обратное преобразование. Таким образом, прикладные уровни могут преодолеть, например, синтаксические различия в представлении данных. Такая ситуация может возникнуть в ЛВС с неоднотипными компьютерами (IBM PC и Macintosh), которым необходимо обмениваться данными. Так, в полях баз данных информация должна быть представлена в виде букв и цифр, а зачастую и в виде графического изображения. Обрабатывать же эти данные нужно, например, как числа с плавающей запятой.

В основу общего представления данных положена единая для всех уровней модели система ASN.1. Эта система служит для описания структуры файлов, а также позволяет решить проблему шифрования данных. На этом уровне может выполняться шифрование и дешифрование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных сервисов. Примером такого протокола является протокол Secure Socket Layer (SSL), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP. Этот уровень обеспечивает преобразование данных (кодирование, компрессия

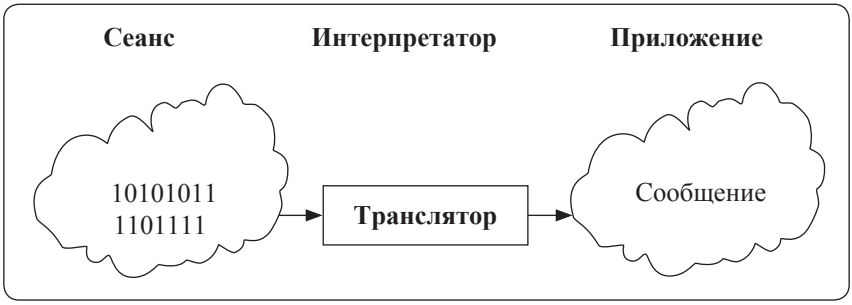


Рис. 2.18. Взаимодействие приложений

и т.п.) прикладного уровня в поток информации для транспортного уровня (*рис. 2.18*).

Представительный уровень выполняет несколько основных функций.

1. Генерация запросов на установление сеансов взаимодействия прикладных процессов.
2. Согласование представления данных между прикладными процессами.
3. Реализация форм представления данных.
4. Представление графического материала (чертежей, рисунков, схем).
5. Засекречивание данных.
6. Передача запросов на прекращение сеансов.

Протоколы уровня представления данных обычно являются составной частью протоколов трех верхних уровней модели.

2.9. Прикладной уровень (Application Layer)

Прикладной уровень обеспечивает прикладным процессам средства доступа к области взаимодействия, является верхним (седьмым) уровнем и непосредственно примыкает к прикладным процессам. В действительности прикладной уровень — это набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например с помощью протокола электронной почты [30]. Специальные элементы прикладного сервиса обеспечивают сервис для конкретных прикладных программ, таких как программы пересылки файлов и эмуляции терминалов. Если, например, программе необходимо переслать файлы, то обязательно будет использован протокол передачи, доступа и управления файлами FTAM (File Transfer, Access, and Management). В модели OSI прикладная программа, которой нужно выполнить конкретную задачу (например, обновить базу данных на компьютере), посылает конкретные данные в виде *дейтаграммы* на прикладной уровень. Одна из основных задач этого уровня — определить, как следует обрабатывать запрос прикладной программы, другими словами, какой вид должен принять данный запрос (*рис. 2.19*).

Единица данных, которой оперирует прикладной уровень, обычно называется сообщением (message).

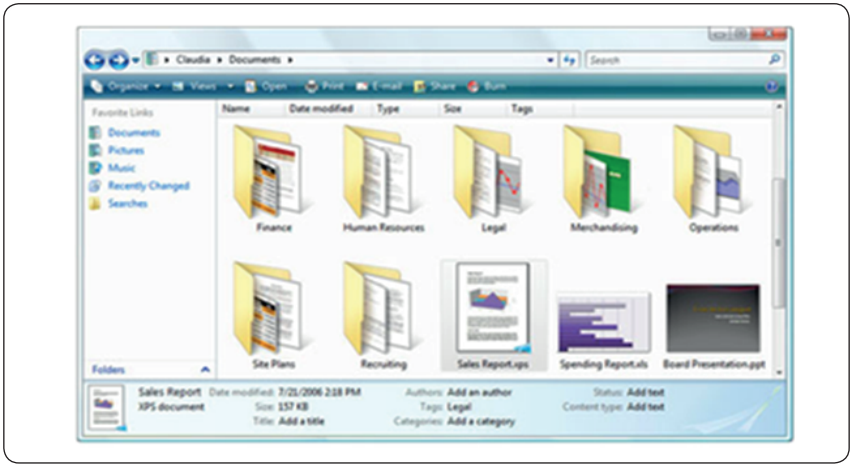


Рис. 2.19. Прикладной уровень

Прикладной уровень выполняет ряд функций:
Описание форм и методов взаимодействия прикладных процессов.

1. Выполнение различных видов работ:
   - передача файлов;
   - управление заданиями;
   - управление системой и т.д.
2. Идентификация пользователей по их паролям, адресам, электронным подписям.
3. Определение функционирующих абонентов и возможности доступа к новым прикладным процессам.
4. Определение достаточности имеющихся ресурсов.
5. Организация запросов на соединение с другими прикладными процессами.
6. Передача заявок представительскому уровню на необходимые методы описания информации.
7. Выбор процедур планируемого диалога процессов.
8. Управление данными, которыми обмениваются прикладные процессы и синхронизация взаимодействия прикладных процессов.
9. Определение качества обслуживания (время доставки блоков данных, допустимой частоты ошибок).

10. Соглашение об исправлении ошибок и определении достоверности данных.
11. Согласование ограничений, накладываемых на синтаксис (наборы символов, структура данных).

Указанные функции определяют виды сервиса, которые прикладной уровень предоставляет прикладным процессам. Кроме этого, прикладной уровень передает прикладным процессам сервис, предоставляемый физическим, канальным, сетевым, транспортным, сеансовым и представительским уровнями.

На *прикладном уровне* необходимо предоставить в распоряжение пользователей уже переработанную информацию. С этим может справиться системное и пользовательское программное обеспечение.

Прикладной уровень отвечает за доступ приложений в сеть. Задачами этого уровня является перенос файлов, обмен почтовыми сообщениями и управление сетью.

К числу наиболее распространенных протоколов верхних трех уровней относятся:

- FTP (File Transfer Protocol) — протокол передачи файлов;
- TFTP (Trivial File Transfer Protocol) — простейший протокол пересылки файлов;
- X.400 — электронная почта;
- Telnet — работа с удаленным терминалом;
- SMTP (Simple Mail Transfer Protocol) — простой протокол почтового обмена;
- CMIP (Common Management Information Protocol) — общий протокол управления информацией;
- SLIP (Serial Line IP) IP для последовательных линий. Протокол последовательной посимвольной передачи данных;
- SNMP (Simple Network Management Protocol) — простой протокол сетевого управления;
- FTAM (File Transfer, Access, and Management) — протокол передачи, доступа и управления файлами.

Сетезависимые и сетенезависимые уровни семиуровневой модели OSI

По своим функциональным возможностям семь уровней модели OSI можно отнести к одной из двух групп:

- группа, в которой уровни зависят от конкретной технической реализации компьютерной сети. Физический, канальный и сетевой уровни являются сетезависимыми, другими словами, эти уровни неразрывно связаны с конкретным используемым сетевым оборудованием;
- группа, в которой уровни в основном ориентированы на работу с приложениями. Сеансовый, представительный и прикладной уровни ориентированы на используемые приложения и практически не зависят от того, какое именно сетевое оборудование используется в компьютерной сети, то есть они являются сетенезависимыми.

2.10. Корректирующие коды

При передаче информации всегда возможно возникновение ошибок, прежде всего из-за помех в канале связи. Существуют различные способы обнаружения и исправления подобных ошибок передачи, когда, например, вместо посланной единицы принимается нуль или наоборот. Можно выделить три таких способа.

Очевидно, что самым простым вариантом является многократная передача информационных сообщений. За правильно принятый сигнал принимается тот, который повторился большее число раз среди неоднократно переданных. Разумеется, это ведет к резкому увеличению времени передачи (к уменьшению ее скорости).

Второй способ заключается в одновременной передаче одних и тех же сообщений по нескольким параллельным каналам связи. В этом случае должно быть не менее трех таких каналов с независимым возникновением ошибок в каждом из них, и решение о правильности передачи принимается на основе совпадения «два из трех». По своей сути этот способ аналогичен предыдущему.

Однако наиболее эффективным способом обнаружения ошибок является использование корректирующих кодов. Принципы построения некоторых из которых мы и рассмотрим.

Основы построения корректирующих кодов

Кодирование информации заключается в представлении чисел и слов соответствующими им комбинациями символов. Каждая такая комбинация называется кодовой комбинацией, а полная их совокупность образует код.

В равномерных кодах все кодовые комбинации содержат одинаковое число знаков (например, пять в коде Бодо). Примером неравномерного кода служит код Морзе.

С помощью n двоичных знаков можно получить 2^n различных комбинаций. Если все они задействованы для представления информации, то такой код называется простым. В нем всякая ошибка, состоящая в замене 0 на 1 или наоборот, превращает одну информационную комбинацию в другую и потому не может быть обнаружена. Например, при $n = 2$ имеем четыре комбинации: 00, 01, 10 и 11. Если ими кодируются числа 0, 1, 2 и 3, то в этом случае любая ошибка при передаче не будет обнаружена, так как всякая принятая двухразрядная кодовая комбинация представляет собой код какого-то одного из четырех реальных чисел. Добавим в каждую кодовую комбинацию по одному дополнительному двоичному разряду. Идея такой избыточности заключается в том, что теперь каждое из четырех чисел будем представлять не двумя, а тремя двоичными разрядами, т.е. в каждой кодовой комбинации один двоичный разряд окажется лишним. Пусть комбинации 001, 010, 100 и 111 соответствуют тем же исходным числам. Но ведь при $n = 3$ можно закодировать восемь чисел от нуля до семи включительно, а мы по-прежнему кодируем только четыре числа. Значит, остальные комбинации 000, 011, 101 и 110 при таком подходе являются запрещенными, поскольку не представляют никакие числа, поэтому их получение будет свидетельствовать о том, что при передаче информации произошла ошибка. Следовательно, надо вводить избыточность, чтобы получить код, обладающий корректирующими возможностями.

Методы автоматического схемного контроля основаны на применении избыточных или корректирующих кодов, причем такие коды называются корректирующими даже в том случае, если они только обнаруживают ошибки, но не исправляют их. В корректирующих кодах лишь часть всех возможных кодовых комбинаций используется для представления информации, а остальные комбинации являются запрещенными, и их появление свидетельствует о наличии ошибки.

В приведенном примере любая одиночная ошибка в двоичном знаке приводит к запрещенной комбинации, которая и будет обнаружена. Избыточность k кода определяется разностью

$$k = n - m,$$

где n — общее количество знаков в данном коде; $m = \log_2 N$ — количество информационных знаков; N — количество двоичных чисел, которые изображаются в коде n знаками.

В систематическом n-значном коде всегда содержится постоянное количество m информационных и k избыточных или контрольных знаков, причем последние занимают одни и те же позиции во всех кодовых комбинациях. Следует отметить, что m и k — целые числа. Так, в нашем примере $n = 3$, $m = \log_2 4 = 2$ и $k = 1$.

Относительная избыточность r кода определяется отношением $r = \frac{k}{m}$.

Корректирующая способность кода определяется вероятностью обнаружения или исправления с его помощью различных ошибок. Эта характеристика кода тесно связана с понятием минимального кодового расстояния, но прежде чем перейти к его рассмотрению, надо определиться еще с двумя понятиями.

Вес $w(a)$ кодовой комбинации а равен количеству содержащихся в ней двоичных единиц. Кодовое расстояние (или хэмминговое расстояние) между двумя кодовыми комбинациями характеризует степень их отличия и определяется количеством позиций, в которых элементы этих комбинаций (двоичные знаки) не совпадают. Следовательно, кодовое расстояние между комбинациями a и b равно весу некоторой третьей комбинации c, полученной поразрядным сложением по модулю 2 этих двух комбинаций, т.е.

$$W(c)w = a \oplus b = (\oplus)\sum_{i-1}^{n} a_i b_i.$$

Например, $a = 101110$, $b = 100011$. Тогда $c = a \oplus b = 001101$ и $w(c) = 3$ — кодовое расстояние между комбинациями a и b.

Минимальным кодовым расстоянием d кода называется минимальное расстояние между двумя любыми комбинациями в этом коде. Например, если есть хотя бы одна пара комбинаций кода, которые отличаются только в одной позиции, то для данного кода $d = 1$ такое значение имеет этот параметр у простого кода.

Для всех корректирующих кодов $d > 1$, и чем оно больше, тем выше корректирующая способность. Так, для уверенного обнаружения одиночной ошибки, т.е. ошибки в одном двоичном разряде, требуется код с $d \geq 2$. Ведь любые два сообщения, представленные в таком коде, отличаются не менее чем в двух позициях (для числовой информации — в двух разрядах). Следовательно, никакая одиночная ошибка не может превратить одну инфор-

мационную комбинацию в другую, она обязательно создаст запрещенную комбинацию, которая и будет обнаружена.

Исправить одиночную ошибку после ее обнаружения можно только при применении кода с $d \geq 3$. В этом случае имеем такую последовательность комбинаций: правильная комбинация, две запрещенные, правильная, две запрещенные и т.д. Любая одиночная ошибка в этом коде создает запрещенную комбинацию, отличающуюся от правильной в одной позиции. От всех других информационных комбинаций она отличается не менее чем в двух позициях. Этого оказывается достаточно для определения позиции ошибочного разряда и его исправления. Например, можно поочередно инвертировать каждый двоичный разряд, проверяя получающуюся после этого комбинацию. Если она остается запрещенной, то разряд восстанавливается. Очевидно, что при этом есть единственная возможность получить информационную комбинацию — инвертировать ошибочный разряд.

Приведенные рассуждения допускают простую геометрическую интерпретацию, представленную на *рис. 2.20*.

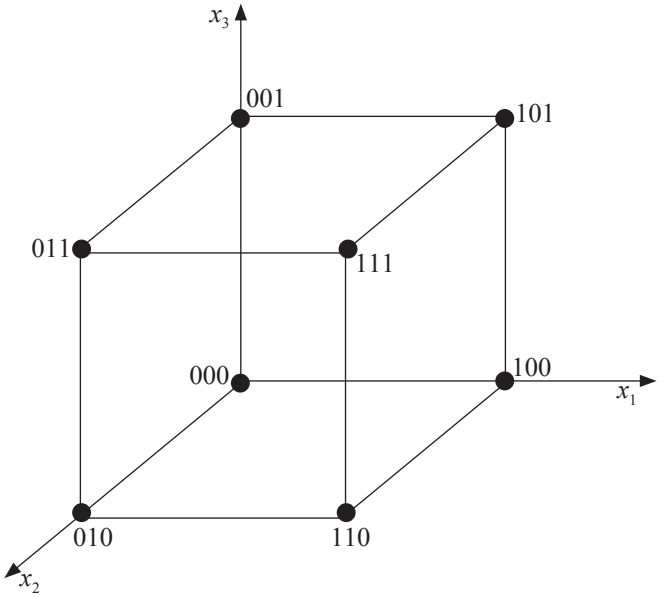


Рис. 2.20. Геометрическое представление кодовых комбинаций

Координаты вершин единичного куба являются трехразрядными кодовыми комбинациями, кодовое расстояние между которыми интерпретируется как сумма длин ребер между соответствующими вершинами. Когда эти кодовые комбинации отличаются друг от друга на длину одного ребра куба ($d = 1$), помеха переводит одну информационную комбинацию в другую, и в этом случае обнаружить ошибку нельзя. Ее можно обнаружить при отличии кодовых комбинаций на два ребра ($d = 2$), а обнаружить и исправить в случае, когда расстояние между кодовыми комбинациями составляет три ребра ($d = 3$).

Аналогичным образом можно показать, что для обнаружения групповых ошибок кратности t (для одиночных ошибок $t = 1$, для двойных $t = 2$ и т.д.) или меньше требуется код с $d = t + 1$, а для исправления таких ошибок — код с $d = 2t + 1$. Но может быть обнаружена и часть других ошибок.

В любом корректирующем коде с минимальным расстоянием d любая информационная комбинация превращается в ближайшую другую информационную комбинацию изменением определенных d разрядов. Если же изменить другие d или даже $d + 1$ разрядов, может получиться запрещенная комбинация, и тогда ошибка кратности $t > d - 1$ будет обнаружена. Правда, обнаруживается лишь часть таких ошибок, а приведенные соотношения подразумевают уверенное обнаружение или исправление t-кратных ошибок.

Одна из основных задач теории кодирования — разработка корректирующих кодов — решается в двух постановках: необходимо разработать код, имеющий максимальную корректирующую способность при заданной избыточности, или код, обеспечивающий заданную корректирующую способность при минимальной избыточности.

Необходимо отметить, что теория кодирования, как и теория передачи информации и многие задачи математической статистики, в частности статистики случайных процессов, является составной частью теории информации.

Теория информации — это раздел кибернетики, занимающийся математическим описанием и оценкой методов передачи, хранения, получения и классификации информации, представляет собой совокупность теорий, общими для которых являются методы теории вероятностей, что отражает присутствие в процессах случайных факторов, связанных с информацией.

Основы теории передачи информации разработал К. Шеннон, решивший проблему нахождения максимально достижимой скорости передачи информации при сколь угодно малой вероятности ошибок, связав ее с количеством

информации. Важное значение в теории кодирования имеют две теоремы, доказанные К. Шенноном. Первая из них утверждает, что для канала связи, не вносящего своих помех, сообщения с энтропией H можно закодировать так, чтобы среднее число элементов кода, приходящихся на один элемент кодируемого алфавита, было бы минимальным (т.е. можно построить код, который сообщениям длины n будет ставить в соответствие кодовые слова в двоичном коде, математическое ожидание длины которых равно nH, с точностью до величины, бесконечно малой по сравнению с n, когда $n \to \infty$).

Согласно второй теореме, относящейся к каналам с помехами, для таких каналов всегда существует способ кодирования, при котором сообщения будут передаваться со сколь угодно высокой достоверностью, если скорость передачи информации не превышает пропускной способности канала связи. Реализация этой возможности составляет содержание помехоустойчивого кодирования и связана с построением корректирующих кодов. В теории кодирования большое внимание уделяется поискам способов кодирования и декодирования, близких к оптимальным и достаточно простых при их аппаратурной реализации.

Код с проверкой на четность/нечетность

Код с проверкой на четность образуется добавлением к группе информационных двоичных разрядов, представляющих простой код, одного контрольного разряда. Его значение (0 или 1) выбирается так, чтобы общее количество единиц в слове всегда было четным (или нечетным, если формируется код с проверкой на нечетность). После любых действий над словом, в том числе и после его пересылки, производится подсчет единиц, которых должно быть четное/нечетное количество. Нарушение этого требования свидетельствует о том, что произошла ошибка. Собственно, идея построения такого кода уже была приведена в примере с кодированием четырех чисел. Теперь остановимся на этом подробнее.

Пусть в четырехзначном двоичном коде только восемь кодовых комбинаций из шестнадцати возможных представляют числа от нуля до семи, а остальные комбинации запрещены. Закодируем числа так:

0 — 0000, 4 — 1001,
1 — 0011, 5 — 1010,
2 — 0101, 6 — 1100,
3 — 0110, 7 — 1111.

Нетрудно заметить, что при таком представлении три старших разряда являются информационными, так как в двоичном виде кодируют исходные числа, а значение четвертого контрольного разряда формируется в соответствии с приведенным правилом, чтобы количество единиц в каждой комбинации было бы четным. Остальные восемь четырехразрядных комбинаций, не представленных здесь, являются запрещенными, поскольку ими числа не кодируются.

Таким образом, сформирован корректирующий код с проверкой на четность, у которого $d = 2$, поэтому с его помощью все одиночные ошибки будут обнаружены. Кроме того, обнаружены будут и все групповые ошибки нечетной кратности (в данном случае — тройные ошибки), так как четность числа единиц в кодовой комбинации при этом нарушается в любом случае.

В принципе, коды с проверкой на четность или нечетность равноценны. Целесообразно все же число единиц в слове делать нечетным. Тогда любая кодовая комбинация будет иметь хотя бы одну единицу (при изображении нуля это будет единица в контрольном разряде). В результате можно отличить отсутствие информации от нуля, если единица изображается наличием электрического сигнала, а нуль — его отсутствием. Впрочем, для представления чисел используются и другие способы.

Код с проверкой на четность имеет небольшую избыточность, но обладает значительной корректирующей способностью.

Коды Хэмминга

Коды Хэмминга имеют большую относительную избыточность, чем коды с проверкой на четность. Они также содержат m информационных и k контрольных разрядов, но каждый из последних является разрядом четности для определенной группы информационных разрядов. При декодировании, под которым в данном случае понимается расшифровка полученных информационных комбинаций, осуществляется k групповых проверок на четность. В результате каждой такой проверки в соответствующий разряд регистра ошибок записывается 0, если проверка была успешной, и 1 — в противном случае, т.е. при обнаружении нечетности. Группы для проверки образуются так, что в регистре после всех проверок оказывается k-разрядное двоичное число, показывающее номер позиции ошибочного двоичного разряда, который инвертируется.

Группы для каждой проверки выбираются по следующему алгоритму. Первая проверка, в результате которой в первый (младший) разряд регистра заносится 0 или 1, охватывает все нечетные позиции кодовой комбинации, включая и принадлежащий этой группе контрольный разряд. Вторая проверка перебирает все позиции, номер которых в двоичном счислении имеет 1 во втором разряде и т.д. Описанная последовательность действий для семиразрядного кода Хэмминга сведена в *табл. 2.1*.

Таблица 2.1

Групповые проверки

Номер проверки	Позиция контрольного знака	Проверяемые позиции
1	1	1,3,5,7
2	2	2,3,6,7
3	4	4,5,6,7

Как видно из этой таблицы, позиция *i*-го контрольного знака имеет номер 2^{i-1}, причем каждый из них входит лишь в одну проверку на четность.

В кодах Хэмминга с увеличением *n* возрастает как количество информационных *m*, так и контрольных *k* знаков. Соотношение между этими параметрами приведено в *табл. 2.2*.

Таблица 2.2

Разрядность кодов Хэмминга

N	3	4	5	6	7
M	1	1	2	3	4
K	2	3	3	3	3

Посмотрим, как число $5_{10} = 0101_2$ будет представлено семиразрядной кодовой комбинацией, структура которой имеет следующий вид:

$$m_4\, m_3\, m_2\, k_3\, m_1\, k_2\, k_1.$$

С информационными разрядами все ясно: $m_4 = m_2 = 0$, $m_3 = m_1 = 1$. Чтобы определиться со значениями контрольных разрядов, надо воспользоваться данными *табл. 2.2*. Согласно им, сумма единиц, содержащихся в первом, третьем, пятом и седьмом разрядах кодовой комбинации, должна быть четной, т.е.

$$k_1 + m_1 + m_2 + m_4 = k_1 + 1 + 0 + 0.$$

Следовательно, $k_1 = 1$. Аналогичным образом, воспользовавшись второй и третьей строками этой же таблицы, определяем, что $k_2 = 0$ и $k_3 = 1$, и окончательно получаем следующий результат: 0101101.

В *табл. 2.3* приведены представления чисел от нуля до девяти в виде семиразрядных кодовых комбинаций.

Таблица 2.3

Семиразрядные коды Хэмминга

Десятичное число	Простой двоичный код	Код Хэмминга	Десятичное число	Простой двоичный код	Код Хэмминга
0	0000	0000000	5	0101	0101101
1	0001	0000111	6	0110	0110011
2	0010	0011001	7	0111	0110100
3	0011	0011110	8	1000	1001011
4	0100	0101010	9	1001	1001100

У рассмотренного кода $d = 3$, поэтому с его помощью можно обнаруживать одиночные и двойные ошибки и исправлять одиночные. Процедура обнаружения и исправления однократной ошибки выглядит следующим образом.

Пусть передано число 6 = 0110011, а принятый код 0100011 содержит ошибку в пятом разряде. Первая проверка (1 + 0 + 0 + 0) даст 1 в младшем разряде регистра ошибок. Вторая проверка (1 + 0 + 1 + 0) даст 0 во втором разряде регистра. Наконец, третья проверка (0 + 0 + 1 + 0) даст 1 в третьем разряде регистра. В результате в регистре ошибок окажется номер ошибочного разряда принятой кодовой комбинации: $101_2 = 5_{10}$. Затем содержимое

пятого разряда инвертируется (в данном случае 0 меняется на 1) и получается правильная кодовая комбинация.

Хотя избыточность k кода Хэмминга не остается постоянной с увеличением длины кода n, но относительная избыточность r с ростом n падает. Это, кстати, относится и к коду с проверкой на четность, у которого $k = 1$ есть постоянная величина.

Контрольные вопросы

1. Какие элементы обеспечивают распространение информации?
2. Чем канал отличается от линии связи?
3. Что такое емкость канала связи?
4. Чем бод отличается от бит/с?
5. Чем определяется максимально возможная пропускная способность идеального канала связи?
6. Чем достоверность передачи информации отличается от ее надежности?
7. Как классифицируются линии связи?
8. Какие функции выполняет модем?
9. Что такое коммуникации и какие они бывают?
10. Что такое компьютерная коммуникационная среда?
11. Как классифицируются вычислительные сети?
12. Что представляет собой протокол обмена информацией?
13. Что представляет собой эталонная модель обмена информацией?
14. Чем отличаются равномерный и неравномерный коды?
15. Что такое избыточность кода и зачем она вводится?
16. Чем определяется корректирующая способность кода?
17. Как подсчитывается вес кодовой комбинации?
18. Что определяет минимальное кодовое расстояние?
19. Какие ошибки обнаруживаются кодом с проверкой на четность?
20. Как формируется семиразрядный код Хэмминга?

3. СТЕК ПРОТОКОЛОВ TCP/IP

3.1. История TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) — это промышленный стандарт стека протоколов, разработанный для глобальных сетей. Стандарты TCP/IP опубликованы в серии документов, названных Request for Comment (RFC). Документы RFC описывают внутреннюю работу сети Internet. Некоторые RFC описывают сетевые сервисы или протоколы и их реализацию, в то время как другие обобщают условия применения.

Стек был разработан по инициативе Министерства обороны США (Department of Defence, DoD) более 20 лет назад для связи экспериментальной сети ARPAnet с другими сателлитными сетями как набор общих протоколов для разнородной вычислительной среды. Сеть ARPA поддерживала разработчиков и исследователей в военных областях. В сети ARPA связь между двумя компьютерами осуществлялась с использованием протокола Internet Protocol (IP), который и по сей день является одним из основных в стеке TCP/IP и фигурирует в названии стека.

Большой вклад в развитие стека TCP/IP внес университет Беркли, реализовав протоколы стека в своей версии ОС UNIX. Широкое распространение ОС UNIX привело и к широкому распространению протокола IP и других протоколов стека. На этом же стеке работает всемирная информационная сеть Internet, чье подразделение Internet Engineering Task Force (IETF) вносит основной вклад в совершенствование стандартов стека, публикуемых в форме спецификаций RFC. Если в настоящее время стек TCP/IP распространен в основном в сетях с ОС UNIX, то реализация его в последних версиях сетевых операционных систем для персональных компьютеров является хорошей предпосылкой для быстрого роста числа установок стека TCP/IP.

Совместимость — одно из основных преимуществ TCP/IP, поэтому большинство ЛВС поддерживает его. Кроме того, TCP/IP предоставляет доступ к ресурсам Интернета, а также маршрутизируемый протокол для сетей масштаба предприятия. Поскольку TCP/IP поддерживает маршрутизацию, он обычно используется в качестве межсетевого протокола. Благодаря своей популярности TCP/IP стал стандартом де-факто для межсетевого взаимодействия (*рис. 3.1*).

Итак, лидирующая роль стека TCP/IP объясняется следующими его свойствами:

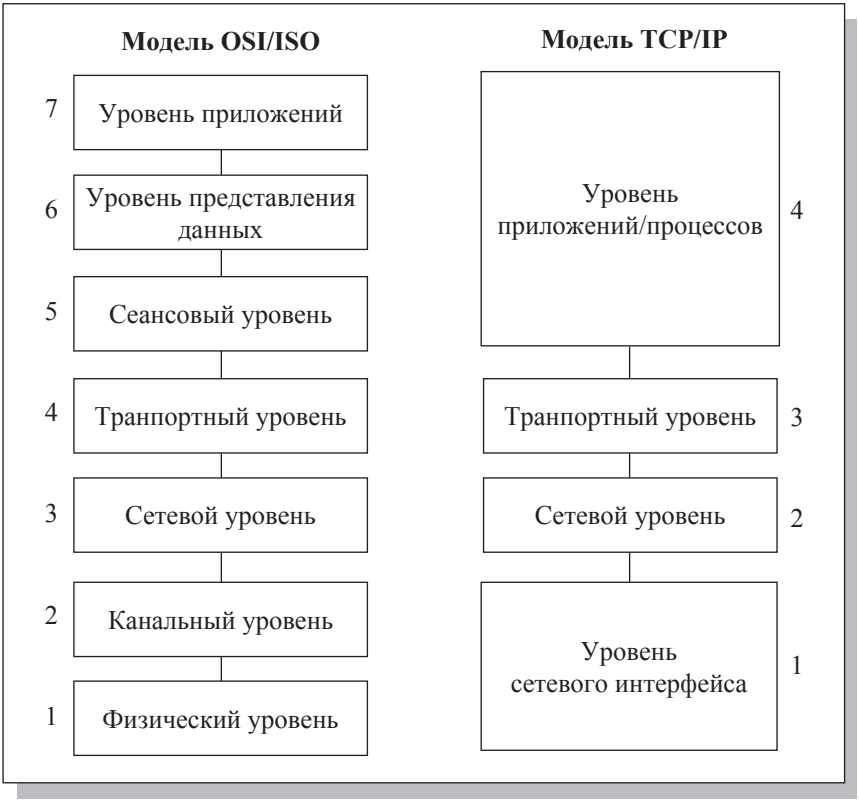


Рис. 3.1. Соотношение моделей OSI/ISO и TCP/IP

- это наиболее завершенный стандартный и в то же время популярный стек сетевых протоколов, имеющий многолетнюю историю;
- почти все большие сети передают основную часть своего трафика с помощью протокола TCP/IP;
- это метод получения доступа к сети Internet;
- этот стек служит основой для создания intranet-корпоративной сети, использующей транспортные услуги Internet и гипертекстовую технологию WWW, разработанную в Internet;
- все современные операционные системы поддерживают стек TCP/IP;
- это гибкая технология для соединения разнородных систем как на уровне транспортных подсистем, так и на уровне прикладных сервисов.

В семействе протоколов TCP/IP можно выделить четыре уровня:

- уровень сетевого интерфейса,
- уровень Internet,
- транспортный уровень,
- уровень приложений/процессов.

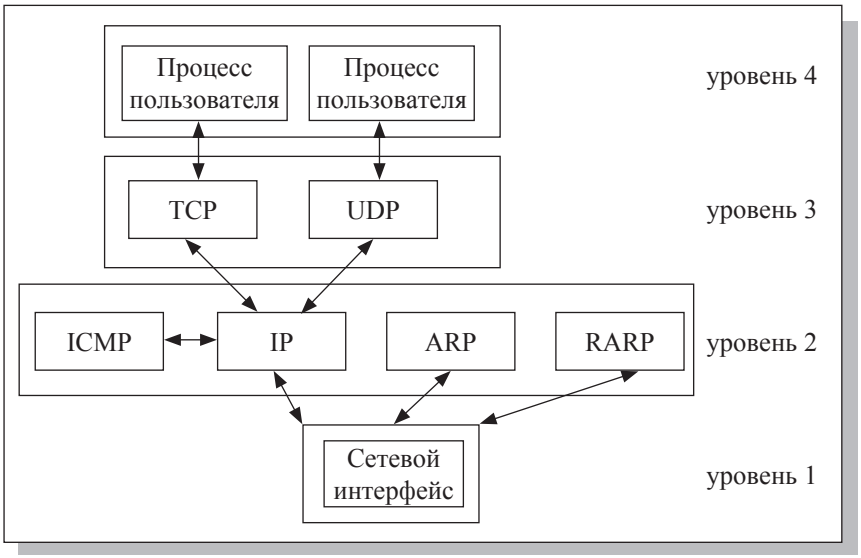


Рис. 3.2. Основные протоколы семейства TCP/IP

На каждом уровне семейства TCP/IP присутствует несколько протоколов. Связь между наиболее употребительными протоколами и их принадлежность уровням изображены на *рис. 3.2*.

3.2. Уровень сетевого интерфейса

Уровень сетевого интерфейса составляют протоколы, которые обеспечивают передачу данных между узлами связи, физически напрямую соединенными друг с другом, или, иначе говоря, подключенными к одному сегменту сети, и соответствующие физические средства передачи данных. К этому уровню относятся протоколы Ethernet, Token Ring, SLIP, PPP и т.д. и такие физические средства, как витая пара, коаксиальный кабель, оптоволоконный кабель и т.д. Формально протоколы уровня сетевого интерфейса не являются частью семейства TCP/IP, но существующие стандарты определяют, каким образом должна осуществляться передача данных семейства TCP/IP с использованием этих протоколов. На уровне сетевого интерфейса в операционной системе UNIX обычно функционируют драйверы различных сетевых плат.

3.3. Уровень Internet

На этом уровне работают следующие протоколы:

- ***ICMP*** — Internet Control Message Protocol. Протокол обработки ошибок и обмена управляющей информацией между узлами сети.
- ***IP*** — Internet Protocol. Это протокол, который обеспечивает доставку пакетов информации для протокола ICMP и протоколов транспортного уровня TCP и UDP.
- ***ARP*** — Address Resolution Protocol. Это протокол для отображения адресов уровня Internet в адреса уровня сетевого интерфейса.
- ***RARP*** — Reverse Address Resolution Protocol. Этот протокол служит для решения обратной задачи: отображения адресов уровня сетевого интерфейса в адреса уровня Internet.

Два последних протокола используются не для всех сетей; только некоторые сети требуют их применения.

Уровень Internet обеспечивает доставку информации от сетевого узла отправителя к сетевому узлу получателя без установления виртуального соединения с помощью дейтаграмм и не является надежным.

Центральным протоколом уровня является протокол IP.

3.4. Транспортный уровень

Задача транспортного уровня — это передача данных между различными приложениями, выполняемыми на всех узлах сети. После того как пакет доставляется с помощью IP-протокола на принимающий компьютер, данные должны быть отправлены специальному процессу-получателю. Каждый компьютер может выполнять несколько процессов, кроме того, приложение может иметь несколько точек входа, действуя в качестве адреса назначения для пакетов данных.

Пакеты, приходящие на транспортный уровень операционной системы, организованы в множества очередей к точкам входа различных приложений. В терминологии TCP/IP такие точки входа называются портами.

К протоколам транспортного уровня относятся протоколы TCP и UDP.

К уровню приложений/процессов можно отнести протоколы TFTP (Trivial File Transfer Protocol), FTP (File Transfer Protocol), Telnet, SMTP (Simple Mail Transfer Protocol) и другие, которые поддерживаются соответствующими системными утилитами.

3.5. Протокол IPv4

IP-адреса (Internet Protocol version 4, интернет-протокол версии 4) представляют собой основной тип адресов, используемый на сетевом уровне модели OSI для осуществления передачи пакетов между сетями. IP-адреса состоят из четырех байт, к примеру 192.168.100.111.

Присвоение IP-адресов хостам осуществляется:

- вручную, настраивается системным администратором во время настройки вычислительной сети;
- автоматически, с использованием специальных протоколов (в частности, с помощью протокола DHCP — Dynamic Host Configuration Protocol, протокол динамической настройки хостов).

Протокол IPv4 разработан в сентябре 1981 года.

Протокол IPv4 работает на межсетевом (сетевом) уровне стека протокола TCP/IP. Основной задачей протокола является осуществление передачи блоков данных (дейтаграмм) от хоста-отправителя, до хоста-назначения, где отправителями и получателями выступают вычислительные машины, однозначно идентифицируемые адресами фиксированной длины (IP-адресами). Также интернет протокол IP осуществляет, в случае необ-

ходимости, фрагментацию и сбор отправляемых дейтаграмм для передачи данных через другие сети с меньшим размером пакетов.

Недостатком протокола IP является ненадежность протокола, то есть перед началом передачи не устанавливается соединение, это говорит о том, что не подтверждается доставка пакетов, не осуществляется контроль корректности полученных данных (с помощью контрольной суммы) и не выполняется операция квитирования (обмен служебными сообщения с узлом-назначения и его готовностью приема пакетов).

Протокол IP отправляет и обрабатывает каждую дейтаграмму как независимую порцию данных, то есть не имея никаких других связей с другими дейтаграммами в глобальной сети интернет.

После отправки дейтаграммы протоколом IP в сеть дальнейшие действия с этой дейтаграммой никак не контролируются отправителем. В том случае, если дейтаграмма по каким-либо причинам не может быть передана дальше по сети, она уничтожается. При этом узел, уничтоживший дейтаграмму, имеет возможность сообщить о причине сбоя отправителю по обратному адресу (в частности, с помощью протокола ICMP). Гарантия доставки данных возложена на протоколы вышестоящего уровня (транспортный уровень), которые наделены для этого специальными механизмами (протокол TCP).

Как известно, на сетевом уровне модели OSI работают маршрутизаторы. Поэтому одна из самых основных задач протокола IP — это осуществление маршрутизации дейтаграмм, другими словами, определение оптимального пути следования дейтаграмм (с помощью алгоритмов маршрутизации) от узла-отправителя сети к любому другому узлу сети на основании IP-адреса.

Формат заголовка IP

Структура IP-пакетов версии 4 представлена на *рис. 3.3*.

Поля заголовка протокола IPv4

- *Версия* — для IPv4 значение поля должно быть равно 4.
- *IHL* — (Internet Header Length) — длина заголовка IP-пакета в 32-битных словах (dword). Именно это поле указывает на начало блока данных в пакете. Минимальное корректное значение для этого поля равно 5.

Октет	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31		
0	Версия IHL	Тип обслуживания	Длина пакета
4	Идентификатор		Флаги Смещение фрагмента
8	Время жизни	Протокол	Контрольная сумма заголовка
12	IP-адрес отправителя		
16	IP-адрес получателя		
20	Параметры от 0-я до 10-и 32-х битовых слов		
	Данные		

Рис. 3.3. Структура IP-пакетов

- *Тип обслуживания* (Type of Service, акроним TOS) — байт, содержащий набор критериев, определяющих тип обслуживания IP-пакетов, представлен на *рис. 3.4*.

0	1	2	3	4	5	6	7
Приоритет			D	T	R	ECN	

Рис. 3.4. Тип обслуживания IP-пакетов

Описание байта обслуживания побитно:

- ✓ 0–2 — приоритет (precedence) данного IP-сегмента
- ✓ 3 — требование ко времени задержки (delay) передачи IP-сегмента (0 — нормальная, 1 — низкая задержка)
- ✓ 4 — требование к пропускной способности (throughput) маршрута, по которому должен отправляться IP-сегмент (0 — низкая, 1 — высокая пропускная способность)
- ✓ 5 — требование к надежности (reliability) передачи IP-сегмента (0 — нормальная, 1 — высокая надежность)
- ✓ 6–7 — ECN — явное сообщение о задержке (управление IP-потоком).

- *Длина пакета* — длина пакета в октетах, включая заголовок и данные. Минимальное корректное значение для этого поля равно 20, максимальное 65535.

- *Идентификатор* — значение, назначаемое отправителем пакета и предназначенное для определения корректной последовательности фрагментов при сборке пакета. Для фрагментированного пакета все фрагменты имеют одинаковый идентификатор.
- *Флаги*. Первый бит флагов должен быть всегда равен нулю, второй бит DF (don't fragment) определяет возможность фрагментации пакета и третий бит MF (more fragments) показывает, не является ли этот пакет последним в цепочке пакетов.
- *Смещение фрагмента* — значение, определяющее позицию фрагмента в потоке данных. Смещение задается количеством восьми байтовых блоков, поэтому это значение требует умножения на 8 для перевода в байты.
- *Время жизни* (TTL) — число маршрутизаторов, которые должен пройти этот пакет. При прохождении маршрутизатора это число уменьшается на единицу. Если значения этого поля равно нулю то, пакет должен быть отброшен и отправителю пакета может быть послано сообщение Time Exceeded (ICMP код 11 тип 0).
- *Протокол* — идентификатор интернет-протокола следующего уровня указывает, данные какого протокола содержит пакет, например, TCP или ICMP.
- *Контрольная сумма заголовка* — вычисляется в соответствии с RFC 1071.

3.6. Фрагментация IP-пакетов

На пути пакета от отправителя к получателю могут встречаться локальные и глобальные сети разных типов с разными допустимыми размерами полей данных кадров канального уровня (Maximum Transfer Unit — MTU). Так, сети Ethernet могут передавать кадры, несущие до 1500 байт данных, для сетей X.25 характерен размер поля данных кадра в 128 байт, сети FDDI могут передавать кадры размером в 4500 байт, в других сетях действуют свои ограничения. Протокол IP умеет передавать дейтаграммы, длина которых больше MTU промежуточной сети, за счет фрагментирования — разбиения «большого пакета» на некоторое количество частей (фрагментов), размер каждой из которых удовлетворяет промежуточную сеть. После того как все фрагменты будут переданы через промежуточную сеть, они будут собраны на узле-получателе модулем протокола IP обратно в «большой пакет». Отметим, что сборку пакета из фрагментов осуществляет только

получатель, а не какой-либо из промежуточных маршрутизаторов. Маршрутизаторы могут только фрагментировать пакеты, но не собирать их. Это связано с тем, что разные фрагменты одного пакета не обязательно будут проходить через одни и те же маршрутизаторы.

Для того, чтобы не перепутать фрагменты разных пакетов, используется поле «Идентификации», значение которого должно быть одинаковым для всех фрагментов одного пакета и не повторяться для разных пакетов, пока у обоих пакетов не истекло время жизни. При делении данных пакета, размер всех фрагментов, кроме последнего, должен быть кратен 8 байтам. Это позволяет отвести меньше места в заголовке под поле «Смещение фрагмента».

Второй бит поля «Флаги» (More fragments), если равен единице, указывает на то, что данный фрагмент — не последний в пакете. Если пакет отправляется без фрагментации, флаг «More fragments» устанавливается в 0, а поле «Смещение фрагмента» — заполняется нулевыми битами.

Если первый бит поля «Флаги» (Don't fragment) равен единице, то фрагментация пакета запрещена. Если этот пакет должен быть передан через сеть с недостаточным MTU, то маршрутизатор вынужден будет его отбросить (и сообщить об этом отправителю посредством протокола ICMP). Этот флаг используется в случаях, когда отправителю известно, что у получателя нет достаточно ресурсов по восстановлению пакетов из фрагментов.

3.7. Классы IP-адресов

Все IP-адреса можно разделить на две логические части — номера сети и номера узла сети (номер хоста). Чтобы определить, какая именно часть IP-адреса принадлежит к номеру сети, а какая — к номеру хоста, определяются значения первых бит адреса. Также первые биты IP-адреса используются для того, чтобы определить, к какому классу относится тот или другой IP-адрес.

На *рис. 3.5*. показана структура IP-адреса разных классов.

Если адрес начинается с 0, то сеть относят к классу A и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса A имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей, о чем будет сказано ниже). Сетей класса A немного, зато количество узлов в них может достигать 2^{24}, то есть 16 777 216 узлов.

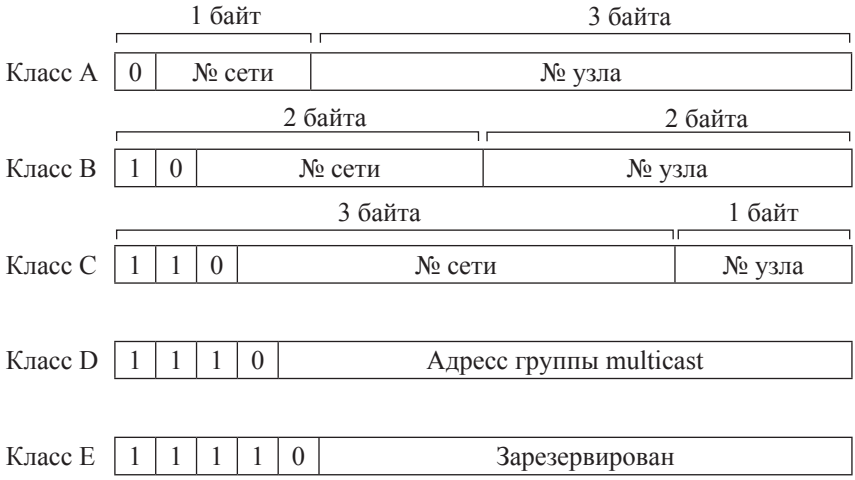


Рис. 3.5. Структура IP-адреса разных классов

Если первые два бита адреса равны 10, то сеть относится к классу B. В сетях класса B под номер сети и под номер узла отводится по 16 бит, то есть по 2 байта. Таким образом, сеть класса B является сетью средних размеров с максимальным числом узлов 2^{16}, что составляет 65 536 узлов.

Если адрес начинается с последовательности 110, то это сеть класса C. В этом случае под номер сети отводится 24 бита, а под номер узла — 8 бит. Сети этого класса наиболее распространены, число узлов в них ограничено 2^8, то есть 256 узлами.

Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает групповой адрес — multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.

Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к классу E. Адреса этого класса зарезервированы для будущих применений.

В *табл. 3.1* приведены диапазоны номеров сетей и максимальное число узлов, соответствующих каждому классу сетей.

Большие сети получают адреса класса A, средние — класса B, а маленькие — класса C.

Таблица 3.1

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	0	1.0.0.0	126.0.0.0	224–2
B	10	128.0.0.0	191.255.0.0	216–2
C	110	192.0.0.0	223.255.255.0	28–2
D	1110	224.0.0.0	239.255.255.255	Multicast
E	11110	240.0.0.0	247.255.255.255	Зарезервирован

3.8. Бесклассовая адресация

Для того, чтобы получить тот или иной диапазон IP-адресов, предприятиям предлагалось заполнить регистрационную форму, в которой перечислялось текущее число ЭВМ и планируемое увеличение количества вычислительных машин, и в итоге предприятию выдавался класс IP-адресов: A, B, C, в зависимости от указанных данных в регистрационной форме.

Данный механизм выдачи диапазонов IP-адресов работал штатно, это было связано с тем, что поначалу в организациях было небольшое количество ЭВМ и соответственно небольшие вычислительные сети. Но в связи с дальнейшим бурным ростом Интернета и сетевых технологий описанный подход к распределению IP-адресов стал выдавать сбои, в основном связанные с сетями класса «B». Действительно, организациям, в которых число компьютеров не превышало нескольких сотен (скажем, 500), приходилось регистрировать для себя целую сеть класса «B» (так как класс «C» только для 254 компьютеров, а класс «B» — для 65534). Из-за этого доступных сетей класса «B» стало просто напросто не хватать, но при этом большие диапазоны IP-адресов пропадали зря.

Традиционная схема деления IP-адреса на номер сети (NetID) и номер узла (HostID) основана на понятии класса, который определяется значениями нескольких первых бит адреса. Именно потому, что первый байт адреса 185.23.44.206 попадает в диапазон 128-191, мы можем сказать, что этот адрес относится к классу B, а значит, номером сети являются первые два байта, дополненные двумя нулевыми байтами — 185.23.0.0, а номером узла — 0.0.44.206.

А что если использовать какой-либо другой признак, с помощью которого можно было бы более гибко устанавливать границу между номером

сети и номером узла? В качестве такого признака сейчас получили широкое распространение маски.

Маска — это число, которое используется в паре с IP-адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Поскольку номер сети является цельной частью адреса, единицы в маске также должны представлять непрерывную последовательность.

Для стандартных классов сетей маски имеют следующие значения:

- класс A — 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс B — 11111111. 11111111. 00000000. 00000000 (255.255.0.0);
- класс C — 11111111. 11111111.11111111. 00000000 (255.255.255.0).

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации. Например, если рассмотренный выше адрес 185.23.44.206 ассоциировать с маской 255.255.255.0, то номером сети будет 185.23.44.0, а не 185.23.0.0, как это определено системой классов.

Расчет номера сети и номера узла с помощью маски:

	Десятичный вид		Двоичный вид	
IP адрес	192.28.44.206	&	11000000.00011100.00101100.11001110	Логическое умножение
маска	255.255.255.0		11111111.11111111.1111111.00000000	
			11000000.00011100.00101100.00000000	Номер сети
			00000000.00000000.00000000.11001110	Номер узла

11000000.00011100.00101100.00000000 → 192.28.44.0
00000000.00000000.00000000.11001110 → 0.0.0.206

В масках количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8, чтобы повторять деление адреса на байты. Пусть, например, для IP-адреса 129.32.156.5 указана маска 255.255.128.0, то есть в двоичном виде:

- IP-адрес 129.32.156.5 — 10000001. 01000000.10000110. 00000101;
- Маска 255.255.128.0 — 11111111.11111111.10000000. 00000000.

Если игнорировать маску, то в соответствии с системой классов адрес 129.32.156.5 относится к классу B, а значит, номером сети являются первые 2 байта — 129.32.0.0, а номером узла — 0.0.156.5.

Если же использовать для определения границы номера сети маску, то 17 последовательных единиц в маске, «наложенные» (логическое умножение) на IP-адрес, определяют в качестве номера сети в двоичном выражении число:

	Десятичный вид		Двоичный вид	
IP адрес	129.32.156.5	&	10000001.00100000.10011100.00000101	Логическое
маска	255.255.128.0		11111111.11111111.1000000.00000000	умножение
			10000001.00100000.10000000.00000000	Номер сети
			00000000.00000000.00111000.00000101	Номер узла

10000001.00100000.10000000.00000000 → 129.32.156.5
00000000.00000000.00111000.00000101 → 0.0.28.5

или в десятичной форме записи — номер сети 129.32.128.0, а номер узла 0.0.28.5.

Существует также короткий вариант записи маски, называемый *префиксом*, или короткой маской. В частности сеть 80.255.147.32 с маской 255.255.255.252, можно записать в виде 80.255.147.32/30, где «/30» указывает на количество двоичных единиц в маске, то есть тридцать бинарных единиц (отсчет ведется слева направо).

В *табл. 3.2* отображается соответствие префикса с маской:

Таблица 3.2

Маска	Префикс	Количество узлов в сети
255.255.255.252	/30	4
255.255.255.248	/29	8
255.255.255.240	/28	16
255.255.255.224	/27	32
255.255.255.192	/26	64
255.255.255.128	/25	128
255.255.255.0	/24	256

Механизм масок широко распространен в IP-маршрутизации, причем маски могут использоваться для самых разных целей. С их помощью администратор может структурировать свою сеть, не требуя от поставщика услуг дополнительных номеров сетей. На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения *префиксов* с целью уменьшения объема таблиц маршрутизации и повышения за счет этого производительности маршрутизаторов. Помимо этого записывать маску в виде префикса значительно короче.

Особые IP-адреса

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов:

- 0.0.0.0 — представляет адрес шлюза по умолчанию, т.е. адрес компьютера, которому следует направлять информационные пакеты, если они не нашли адресата в локальной сети;
- 255.255.255.255 — широковещательный адрес. Сообщения, переданные по этому адресу, получают все узлы локальной сети, содержащей компьютер-источник сообщения (в другие локальные сети оно не передается);
- «номер сети».«все нули» — адрес сети (например 192.168.10.0);
- «все нули».«номер узла» — узел в данной сети (например 0.0.0.23). Может использоваться для передачи сообщений конкретному узлу внутри локальной сети;

Если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номером сети. Например, пакет с адресом 192.190.21.255 доставляется всем узлам сети 192.190.21.0. Такая рассылка называется широковещательным сообщением (broadcast). При адресации необходимо учитывать те ограничения, которые вносятся особым назначением некоторых IP-адресов. Так, ни номер сети, ни номер узла не может состоять только из одних двоичных единиц или только из одних двоичных нулей. Отсюда следует, что максимальное количество узлов, приведенное в таблице для сетей каждого класса, на практике должно быть уменьшено на 2. Например, в сетях класса C под номер узла отводится 8 бит, которые позволяют задавать 256 номеров: от 0 до 255. Однако на практике максимальное число узлов в сети класса C не может превышать 254, так как адреса 0 и 255 имеют специальное назначение. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса A состоит из одних двоичных единиц.

3.9. Протокол ARP

ARP **—** протокол разрешения адресов (Address Resolution Protocol) является протоколом третьего (сетевого) уровня модели OSI, используется для преобразования IP-адресов в MAC-адреса, играет важную функцию в

множественном доступе сетей. ARP была определена RFC 826 в 1982 году (*рис 3.6*).

В сети Ethernet для идентификации источника и получателя информации используются IP и MAC адреса.

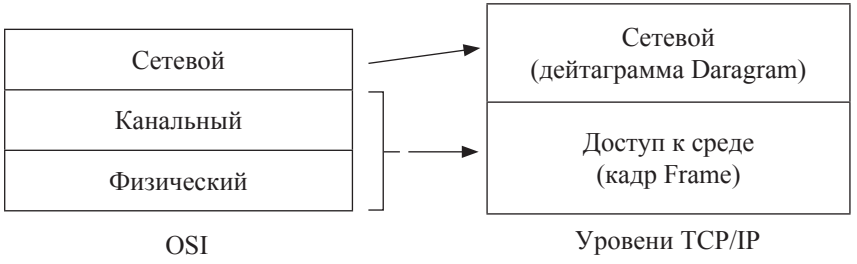


Рис. 3.6. Протокол ARP

Информация, пересылаемая от одного компьютера другому по сети, содержит в себе физический адрес отправителя, IP-адрес отправителя, физический адрес получателя и IP-адрес получателя.

ARP-протокол обеспечивает связь между этими двумя адресами, поскольку эти два адреса никак друг с другом не связаны.

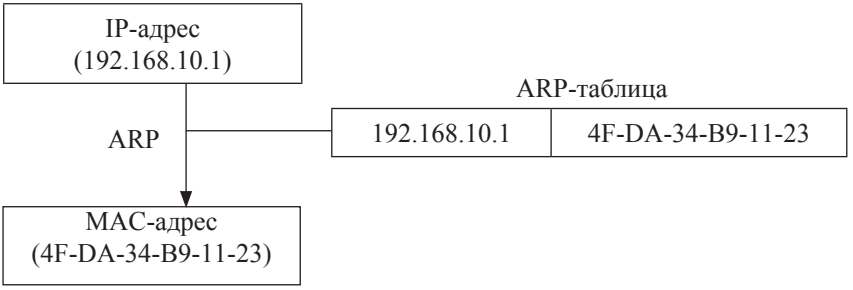


Рис. 3.7. Преобразование адресов ARP-протокол

Преобразование адресов выполняется путем поиска в таблице. Эта таблица, называемая ARP-таблицей, хранится в памяти и содержит строки для каждого узла сети. В двух столбцах содержатся IP- и Ethernet-адреса.

Если требуется преобразовать IP-адрес в Ethernet-адрес, то ищется запись с соответствующим IP-адресом. Ниже приведен пример упрощенной ARP-таблицы (*рис. 3.8*).

IP-адрес	Ethernet-адрес
223.1.2.1	08:00:39:00:2F:C3
223.1.2.3	08:00:5A:21:A7:22
223.1.2.4	08:00:10:99:AC:54

Рис. 3.8. Пример ARP-таблицы

ARP-таблица необходима потому, что IP-адреса и Ethernet-адреса выбираются независимо, и нет какого-либо алгоритма для преобразования одного в другой. IP-адрес выбирает менеджер сети с учетом положения машины в сети Internet.

Эффективность функционирования ARP во многом зависит от ARP кэша (ARP cache), который присутствует на каждом хосте. В кэше содержатся Internet адреса и соответствующие им аппаратные адреса. Стандартное время жизни каждой записи в кэше составляет 20 минут с момента создания записи.

3.10. Схема работы протокола ARP

ARP предоставляет динамическое сопоставление IP адресов и соответствующих аппаратных адресов.

В ходе обычной работы сетевая программа, такая как TELNET, отправляет прикладное сообщение, пользуясь транспортными услугами TCP. Модуль TCP посылает соответствующее транспортное сообщение через модуль IP. В результате составляется IP-пакет, который должен быть передан драйверу Ethernet. IP-адрес места назначения известен прикладной программе, модулю TCP и модулю IP. Необходимо на его основе найти Ethernet-адрес места назначения. Для определения искомого Ethernet-адреса используется ARP-таблица.

Запросы и ответы протокола ARP

Таблица заполняется автоматически модулем ARP, по мере необходимости. Когда с помощью существующей ARP-таблицы не удается преобразовать IP-адрес, то происходит следующее:

1) По сети передается широковещательный ARP-запрос;
2) Исходящий IP-пакет ставится в очередь.

Каждый сетевой адаптер принимает широковещательные передачи. Все драйверы Ethernet проверяют поле типа в принятом Ethernet-кадре и передают ARP-пакеты модулю ARP. ARP-запрос можно интерпретировать так: «Если ваш IP-адрес совпадает с указанным, то сообщите мне ваш Ethernet-адрес».

Пример ARP-запроса (*рис. 3.9*).

Р-адрес отправителя Ethernet-адрес отправителя	223.1.2.1 08:00:39:00:2F:C3
Искомый IP-адрес Искомый Ethernet-адрес	223.1.2.2 <пусто>

Рис. 3.9. Пример ARP-запроса

Каждый модуль ARP проверяет поле искомого IP-адреса в полученном ARP-пакете и, если адрес совпадает с его собственным IP-адресом, то посылает ответ прямо по Ethernet-адресу отправителя запроса. ARP-ответ можно интерпретировать так: «Да, это мой IP-адрес, ему соответствует такой-то Ethernet-адрес». Пакет с ARP-ответом выглядит примерно так (*рис. 3.10*):

IP-адрес отправителя Ethernet-адрес отправителя	223.1.2.2 08:00:28:00:38:A9
Искомый IP-адрес Искомый Ethernet-адрес	223.1.2.1 08:00:39:00:2F:C3

Рис. 3.10. Пакет с ARP-ответом

Этот ответ получает машина, сделавшая ARP-запрос. Драйвер этой машины проверяет поле типа в Ethernet-кадре и передает ARP-пакет модулю

ARP. Модуль ARP анализирует ARP-пакет и добавляет запись в свою ARP-таблицу.

Этот ответ получает машина, сделавшая ARP-запрос. Драйвер этой машины проверяет поле типа в Ethernet-кадре и передает ARP-пакет модулю ARP. Модуль ARP анализирует ARP-пакет и добавляет запись в свою ARP-таблицу.

Обновленная таблица выглядит следующим образом (*рис. 3.11*):

IP-адрес	Ethernet-адрес
223.1.2.1	08:00:39:00:2F:C3
223.1.2.2	08:00:28:00:38:A9
223.1.2.3	08:00:5A:21:A7:22
223.1.2.4	08:00:10:99:AC:54

Рис. 3.11. Обновленная таблица

RARP (англ. Reverse Address Resolution Protocol — Обратный протокол преобразования адресов) — протокол третьего (сетевого) уровня модели OSI, выполняет обратное отображение адресов, то есть преобразует аппаратный адрес в IP-адрес (*рис. 3.12*).

Существует четыре типа ARP-сообщений:

- ARP-запрос (ARPrequest);
- ARP-ответ (ARP reply);
- RARP-запрос (RARP-request);
- RARP-ответ (RARP-reply).

Структура заголовка ARP показана на *рис. 3.13.*

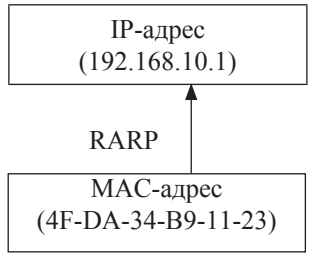


Рис. 3.12. Обратное преобразование адресов

Поля заголовка протокола ARP:

- Hardware type (HTYPE). Каждый канальный протокол передачи данных имеет свой номер, который хранится в этом поле. Например, Ethernet имеет номер 0x0001.
- Protocol type (PTYPE). Код сетевого протокола. Например, для IPv4 будет записано 0x0800.
- Hardware length (HLEN). Длина физического адреса в байтах. Адреса Ethernet имеют длину 6 байт.

Октет	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	
0	Hardware Type (HTYPE)	Protocol Type (PTYPE)
4	Hardware length (HLEN) Protocol length (PLEN)	Operation (OPER)
	Sender hardware address (SHA)	
	Sender protocol address (SPA)	
	Target hardware address (THA)	
	Target protocol address (TPA)	

Рис. 3.13. Структура заголовка ARP

- Protocol length (PLEN). Длина логического адреса в байтах. IPv4 адреса имеют длину 4 байта.
- Operation. Код операции отправителя: 1 в случае запроса и 2 в случае ответа.
- Sender hardware address (SHA). Физический адрес отправителя.
- Sender protocol address (SPA). Логический адрес отправителя.
- Targethardwareaddress (THA). Физический адрес получателя. Поле пусто при запросе.
- Target protocol address (TPA). Логический адрес получателя.

Сообщения ARP поддерживаются в различных локальных сетях:

- адаптеры Ethernet LAN (поддерживает протоколы Ethernet и 802.3);
- адаптеры Token-Ring;
- адаптеры FDDI (Оптоволоконного интерфейса распределенных данных).

Таблицы преобразования обслуживаются ядром ОС, поэтому у пользователей и приложений нет доступа непосредственно к ARP. При отправке IP-пакета одному из драйверов интерфейса драйвер запрашивает преобразование соответствующего адреса. Если в таблице нет соответствующего аппаратного адреса, ARP рассылает пакет с запросом драйвера интерфейса всем хостам локальной сети.

Полные записи хранятся в таблице ARP в течение 20 минут, неполные — в течение 3 минут.

3.11. Протокол управляющих сообщений Internet (ICMP)

Протокол ICMP — обязательная часть любой реализации IP. ICMP отправляет сообщения об ошибках и управляющие сообщения протоколу IP.

С помощью этого протокола шлюзы и хосты отправляют источнику пакетов отчеты о неполадках. ICMP выполняет следующие функции:

- проверяет, что хост-получатель активен и доступен;
- сообщает об неправильных параметрах в заголовке дейтаграммы;
- синхронизует часы и определяет время передачи данных по маршруту;
- определяет IP-адреса и маски подсетей.

Как и протоколы более высокого уровня, протокол ICMP использует базовые функции IP. Однако в действительности ICMP представляет собой часть протокола IP и должен быть реализован в каждом модуле IP.

Протокол ICMP — это всего лишь средство обмена информацией о неполадках в сети. Он не повышает надежность протокола IP. Таким образом, ICMP не гарантирует надежной доставки IP-пакета, а также доставки сообщения ICMP в случае, если IP-пакет не был получен или был получен в искаженном виде.

Сообщения ICMP отправляются в следующих ситуациях:

- когда пакет невозможно доставить получателю;
- когда размер буфера шлюза недостаточен для пересылки пакета;
- когда шлюз может предложить хосту более короткий маршрут для доставки пакета.

TCP/IP отправляет и принимает сообщения ICMP нескольких типов. Протокол ICMP встроен в ядро и для него не предусмотрен интерфейс прикладных программ (API).

3.12. Сетевая маршрутизация

Маршрутизация (routing) — процесс определения в коммуникационной сети (наилучшего) пути, по которому пакет может достигнуть адресата или, точнее — это набор правил, определяющих маршрут следования информации в сетях связи. Часто критерием при выборе маршрута является время.

Для маршрутизации пакета маршрутизатор должен владеть следующей информацией:

- адрес назначения;
- соседний маршрутизатор, от которого он может узнать об удаленных сетях;
- доступные пути ко всем удаленным сетям;

- наилучший путь к каждой удаленной сети;
- методы обслуживания и проверки информации о маршрутизации.

Маршрутизатор узнает об удаленных сетях от соседних маршрутизаторов или от сетевого администратора. Затем маршрутизатор строит таблицу маршрутизации, которая описывает, как найти удаленные сети.

Динамическая маршрутизация — это процесс протокола маршрутизации, определяющий взаимодействие устройства с соседними маршрутизаторами. Маршрутизатор будет обновлять сведения о каждой изученной им сети. Если в сети произойдет изменение, протокол динамической маршрутизации автоматически информирует об изменении все маршрутизаторы. Если же используется статическая маршрутизация, обновить таблицы маршрутизации на всех устройствах придется системному администратору.

В стеке TCP/IP маршрутизаторы и конечные узлы на основании так называемых таблиц маршрутизации (routing tables) принимают решения о том, кому передавать пакет для его успешной доставки узлу назначения.

Таблица маршрутизации может составляться двумя способами: статично и динамично. В случае *статической маршрутизации* записи в таблице вводятся и изменяются вручную. Такой способ требует вмешательства администратора каждый раз, когда происходят изменения в топологии сети. С другой стороны, он является наиболее стабильным и требующим минимума аппаратных ресурсов маршрутизатора для обслуживания таблицы.

При *динамической маршрутизации* записи в таблице обновляются автоматически при помощи одного или нескольких протоколов маршрутизации — *RIP*, *OSPF*, *IGRP*, *EIGRP* и др.

Кроме того, *маршрутизатор* строит таблицу оптимальных путей к сетям назначения на основе различных критериев (метрик), таких как: количество промежуточных узлов, пропускная способность каналов, задержка передачи данных и т.п. *Динамическая маршрутизация* оказывает дополнительную нагрузку на устройства, а высокая нестабильность сети может приводить к ситуациям, когда маршрутизаторы не успевают синхронизировать свои таблицы, что приводит к противоречивым сведениям о топологии сети в различных её частях и потере передаваемых данных.

Статическая маршрутизация — вид маршрутизации, при котором информация о маршрутах заносится в таблицы маршрутизации каждого маршрутизатора вручную администратором сети. Статическая *маршрутизация* успешно используется при организации работы компьютерных

сетей небольшого размера (1–2 маршрутизатора) в силу легкости конфигурации и отсутствия дополнительной нагрузки на сеть в виде широковещательного служебного трафика, характерного для динамических протоколов маршрутизации. Также статическая маршрутизация используется на компьютерах внутри сети. В таком случае обычно задается маршрут шлюза по умолчанию.

Динамическая маршрутизация — вид маршрутизации, при котором таблица маршрутизации заполняется и обновляется автоматически при помощи одного или нескольких протоколов маршрутизации (RIP, OSPF, EIGRP, BGP).

Каждый протокол маршрутизации использует свою систему оценки маршрутов (метрику). Маршрут к сетям назначения строится на основе таких критериев:

- количество ретрансляционных переходов;
- пропускная способность канала связи;
- задержки передачи данных и др.

Маршрутизаторы обмениваются друг с другом информацией о маршрутах с помощью служебных пакетов по протоколу *UDP*. Такой обмен информации увеличивает наличие дополнительного трафика в сети и нагрузку на эту *сеть*. Возможна также ситуация, при которой таблицы маршрутизации на роутерах не успевают согласоваться между собой, что может повлечь появление ошибочных маршрутов и потерю данных.

Протоколы маршрутизации делятся на три типа:

- дистанционно векторные протоколы (RIP);
- протоколы с отслеживанием состояния каналов (OSPF);
- смешанные протоколы (EIGRP) и др.

3.13. Протокол RIP

RIP — протокол дистанционно-векторной маршрутизации, использующий для нахождения оптимального пути *алгоритм* Беллмана — Форда.

Алгоритм маршрутизации RIP — один из самых простых протоколов маршрутизации. Каждые 30 секунд он передает в сеть свою таблицу маршрутизации. Максимальное количество «хопов» (шагов до места назначения), разрешенное в RIP1, равно 15 (метрика 15). Ограничение в 15 хопов не дает применять RIP в больших сетях, поэтому протокол наиболее распространен в небольших компьютерных сетях.

Вторая версия протокола — протокол RIP2 — была разработана в 1994 году и является улучшенной версией первого. В этом протоколе повышена безопасность за счет введения дополнительной маршрутной информации. Принцип дистанционно-векторного протокола: каждый маршрутизатор, использующий протокол RIP периодически широковещательно рассылает своим соседям специальный пакет-вектор, содержащий расстояния (измеряются в метрике) от данного маршрутизатора до всех известных ему сетей. Маршрутизатор, получивший такой вектор, наращивает компоненты вектора на величину расстояния от себя до данного соседа и дополняет вектор информацией об известных непосредственно ему самому сетях или сетях, о которых ему сообщили другие маршрутизаторы. Дополненный вектор маршрутизатор рассылает всем своим соседям. Маршрутизатор выбирает из нескольких альтернативных маршрутов маршрут с наименьшим значением метрики, а маршрутизатор, передавший информацию о таком маршруте, помечается как следующий (next hop). Протокол непригоден для работы в больших сетях, так как засоряет сеть интенсивным трафиком, а узлы сети оперируют только векторами-расстояний, не имея точной информации о состоянии каналов и топологии сети. Сегодня даже в небольших сетях протокол вытесняется превосходящими его по возможностям протоколами EIGRP и OSPF.

3.14. Алгоритмы маршрутизации

Основные требования к алгоритмам маршрутизации:

- точность;
- простота;
- надёжность;
- стабильность;
- справедливость;
- оптимальность.

Существуют различные алгоритмы построения таблиц для одношаговой маршрутизации. Их можно разделить на три класса:

- алгоритмы простой маршрутизации;
- алгоритмы фиксированной маршрутизации;
- алгоритмы адаптивной маршрутизации.

Простая маршрутизация

Это способ маршрутизации не изменяющийся при изменении топологии и состоянии сети передачи данных (СПД).

Простая маршрутизация обеспечивается различными алгоритмами, типичными из которых являются перечисленные ниже.

- Случайная маршрутизация — это передача сообщения из узла в любом случайно выбранном направлении, за исключением направлений, по которым сообщение поступило узел.
- Лавинная маршрутизация — это передача сообщения из узла во всех направлениях, кроме направления, по которому сообщение поступило в узел. Такая маршрутизация гарантирует малое время доставки пакета, засчет ухудшения пропускной способности.
- Маршрутизация по предыдущему опыту — каждый пакет имеет счетчик числа пройденных узлов, в каждом узле связи анализируется счетчик и запоминается тот маршрут, который соответствует минимальному значению счетчика. Такой алгоритм позволяет приспосабливаться к изменению топологии сети, но процесс адаптации протекает медленно и неэффективно.

В целом простая маршрутизация не обеспечивает направленную передачу пакета и имеет низкую эффективности. Основным ее достоинством является обеспечение устойчивой работы сети при выходе из строя различных частей сети.

Фиксированная маршрутизация

Этот алгоритм применяется в сетях с простой топологией связей и основан на ручном составлении таблицы маршрутизации администратором сети. Алгоритм часто эффективно работает также для магистралей крупных сетей, так как сама магистраль может иметь простую структуру с очевидными наилучшими путями следования пакетов в подсети, присоединенными к магистрали. Выделяют несколько алгоритмов.

- Однопутевая фиксированная маршрутизация — установление между хостами единственного пути. Сеть с такой маршрутизацией неустойчива к отказам и перегрузкам.
- Многопутевая фиксированная маршрутизация — установление нескольких возможных путей и введение правила выбора пути. Эф-

фективность такой маршрутизации падает при увеличении нагрузки. При отказе какой-либо линии связи необходимо менять таблицу маршрутизации, для этого в каждом узле связи храниться несколько таблиц.

Адаптивная маршрутизация

Это основной вид алгоритмов маршрутизации, применяющихся маршрутизаторами в современных сетях со сложной топологией. Адаптивная маршрутизация основана на том, что маршрутизаторы периодически обмениваются специальной топологической информацией об имеющихся в интерсети сетях, а также о связях между маршрутизаторами. Обычно учитывается не только топология связей, но и их пропускная способность и состояние.

Адаптивные протоколы позволяют всем маршрутизаторам собирать информацию о топологии связей в сети, оперативно отрабатывая все изменения конфигурации связей. Основные алгоритмы:

- локальная адаптивная маршрутизация — каждый узел содержит информацию о состоянии линии связи, длины очереди и таблицу маршрутизации;
- глобальная адаптивная маршрутизация — основана на использовании информации, получаемой от соседних узлов. Для этого каждый узел содержит таблицу маршрутизации, в которой указано время прохождения сообщений. На основе информации, получаемой из соседних узлов, значение таблицы пересчитывается с учетом длины очереди в самом узле;
- централизованная адаптивная маршрутизация — существует некоторый центральный узел, который занимается сбором информации о состоянии сети. Этот центр формирует управляющие пакеты, содержащие таблицы маршрутизации и рассылает их в узлы связи;
- гибридная адаптивная маршрутизация — основана на использовании таблицы, периодически рассылаемой центром, и на анализе длины очереди на самом узле.

Показатели алгоритмов (метрики), которые используются в алгоритмах маршрутизации:

- длина маршрута;
- надежность;

- задержка;
- ширина полосы пропускания.

Длина маршрута

Длина маршрута является наиболее общим показателем маршрутизации. Некоторые протоколы маршрутизации позволяют администраторам сети назначать произвольные цены на каждый канал сети. В этом случае длиной тракта является сумма расходов, связанных с каждым каналом, который был траверсирован. Другие протоколы маршрутизации определяют «количество пересылок» (количество хопов), т.е. показатель, характеризующий число проходов, которые пакет должен совершить на пути от источника до пункта назначения через элементы объединения сетей (такие, как маршрутизаторы).

Надежность

Надежность, в контексте алгоритмов маршрутизации, относится к надежности каждого канала сети (обычно описываемой в терминах соотношения бит/ошибка). Некоторые каналы сети могут отказывать чаще, чем другие. Отказы одних каналов сети могут быть устранены легче или быстрее, чем отказы других каналов. При назначении оценок надежности могут быть приняты в расчет любые факторы надежности.

Задержка

Под задержкой маршрутизации обычно понимают отрезок времени, необходимый для передвижения пакета от источника до пункта назначения через объединенную сеть. Задержка зависит от многих факторов, включая полосу пропускания промежуточных каналов сети, очереди в порт каждого маршрутизатора на пути передвижения пакета, перегруженность сети на всех промежуточных каналах сети и физическое расстояние, на которое необходимо переместить пакет.

Полоса пропускания

Полоса пропускания относится к имеющейся мощности трафика какого-либо канала. Хотя полоса пропускания является оценкой максимально

достижимой пропускной способности канала, маршруты, проходящие через каналы с большей полосой пропускания, не обязательно будут лучше маршрутов, проходящих через менее быстродействующие каналы.

3.15. Транспортный протокол TCP

Transmission Control Protocol (TCP, протокол управления передачей) является обязательным протоколом стандарта TCP/IP, определенным в стандарте RFC 793, «Transmission Control Protocol (TCP)».

TCP — это протокол транспортного уровня, предоставляющий транспортировку (передачу) потока данных с необходимостью предварительного установления соединения, благодаря чему гарантирует уверенность в целостности получаемых данных, также выполняет повторный запрос данных в случае потери данных или искажения. Помимо этого протокол TCP отслеживает дублирование пакетов и в случае обнаружения — уничтожает дублирующиеся пакеты.

В отличие от протокола UDP гарантирует целостность передаваемых данных и подтверждение отправителя о результатах передачи. Используется при передаче файлов, где потеря одного пакета может привести к искажению всего файла.

TCP обеспечивает свою надежность благодаря ряду факторов:

- Данные от приложения разбиваются на блоки определенного размера, которые будут отправлены.
- При отправке сегмента протокол TCP устанавливает таймер, ожидая, что с удаленного конца придет подтверждение на этот сегмент. Если подтверждение не получено по истечении времени, сегмент передается повторно.
- После того как TCP принял данные от удаленной стороны соединения, он отправляет подтверждение. Это подтверждение не отправляется немедленно, а обычно задерживается на доли секунды.
- TCP осуществляет расчет контрольной суммы для своего заголовка и данных. Это контрольная сумма, рассчитываемая на концах соединения, целью которой является выявить любое изменение данных в процессе передачи. Если сегмент прибывает с неверной контрольной суммой, TCP отбрасывает его и подтверждение не генерируется. Ожидается, что отправитель отработает тайм-аут и осуществит повторную передачу.

Так как TCP сегменты передаются в виде IP дейтаграмм, а IP дейтаграммы могут прибывать беспорядочно, также беспорядочно могут прибывать и TCP сегменты. После получения данных TCP может по необходимости изменить их последовательность, в результате приложение получает данные в правильном порядке.

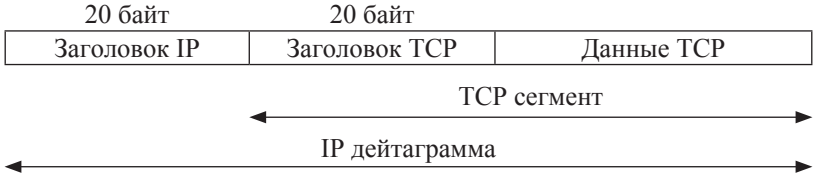
Так как IP дейтаграмма может быть продублирована, принимающий TCP должен отбрасывать продублированные данные.

TCP осуществляет контроль потока данных. Каждая сторона TCP соединения имеет определенное пространство буфера. TCP на принимающей стороне позволяет удаленной стороне посылать данные только в том случае, если получатель может поместить их в буфер. Это предотвращает от переполнения буферов медленных хостов быстрыми хостами.

Заголовок TCP

Поля заголовка протокола TCP (*рис. 3.14*):

- Порт отправителя.
- Порт получателя.
- Порядковый номер. Выполняет две задачи. Если установлен флаг SYN, то это начальное значение номера последовательности — ISN (Initial Sequence Number), и первый байт данных, которые будут переданы в следующем пакете, будет иметь номер последовательности, равный ISN + 1. В противном случае, если SYN не установлен, первый байт данных, передаваемый в данном пакете, имеет этот номер последовательности.
- Номер подтверждения. Если установлен флаг ACK, то это поле содержит номер последовательности, ожидаемый получателем в следующий раз. Помечает этот сегмент как подтверждение получения.
- Длина заголовка — задается словами по 32бита.
- Размер окна — количество байт, которые готов принять получатель без подтверждения.
- Контрольная сумма — включает псевдо заголовок, заголовок и данные.
- Указатель срочности — указывает последний байт срочных данных, на которые надо немедленно реагировать.
- URG — флаг срочности, включает поле «Указатель срочности», если его значение равно нулю, то поле игнорируется.



Октет	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31																															
0	Порт отправителя																Порт получателя															
4	Порядковый номер																															
8	Номер подтверждения																															
12	Длина заг. TCP				Зарезервированное поле						U R G	A C K	P S H	R S T	S S T	F I N	Размер окна															
16	Контрольная сумма																								Указатель срочности							
20	Дополнительные параметры (переменная длина)																												Заполнение (нули)			
	Данные																															

Рис. 3.14. Поля заголовка TCP

- ACK — флаг подтверждения, включает поле «Номер подтверждения», если его значение равно нулю, то поле игнорируется.
- PSH — флаг требует выполнения операции push, модуль TCP должен срочно передать пакет программе.
- RST — флаг прерывания соединения, используется для отказа в соединении
- SYN – флаг синхронизации порядковых номеров, используется при установлении соединения.
- FIN – флаг окончания передачи со стороны отправителя.

TCP порты

Так как на одном и том же компьютере могут быть запущены несколько программ, то для доставки TCP-пакета конкретной программе, используется уникальный идентификатор каждой программы или номер порта.

Номер порта — это условное 16-битное число от 1 до 65535, указывающее, какой программе предназначается пакет.

TCP порты используют определенный порт программы для доставки данных, передаваемых с помощью протокола управления передачей (TCP). TCP порты являются более сложными и работают иначе, чем порты UDP. В то время как порт UDP работает как одиночная очередь сообщений и как точка входа для UDP-соединения, окончательной точкой входа для всех соединений TCP является уникальное соединение. Каждое соединение TCP однозначно идентифицируется двумя точками входа.

Каждый отдельный порт сервера TCP может предложить общий доступ к нескольким соединениям, потому что все TCP соединения идентифицируются двумя значениями: IP-адресом и TCP портом (сокет).

Все номера портов TCP, которые меньше чем 1024, зарезервированы и зарегистрированы в Internet Assigned Numbers Authority (IANA).

Номера портов UDP и TCP не пересекаются.

В *табл. 3.3* приведены зарезервированные или хорошо известные номера портов для протоколов.

Таблица 3.3

Номер порта TCP	Описание
20	FTP сервер (канал данных)
21	FTPсервер (канал контроля)
23	Telnet сервер
53	Система перевода зоны Доменных имен
80	Web-сервер (http)
139	Служба сеансов NetBIOS

Установление TCP-соединения

Рассмотрим процесс установления TCP-соединения (*рис. 3.15*). Предположим, что процесс, работающий на одном хосте, хочет установить соединение с другим процессом на другом хосте. Хост, который инициирует соединение, называется «клиентом», в то время как другой узел называется «сервером».

Перед началом передачи каких-либо данных, согласно протоколу TCP, стороны должны установить соединение. Соединение устанавливается в три этапа (процесс «трёхкратного рукопожатия» TCP) (*рис. 3.15*).

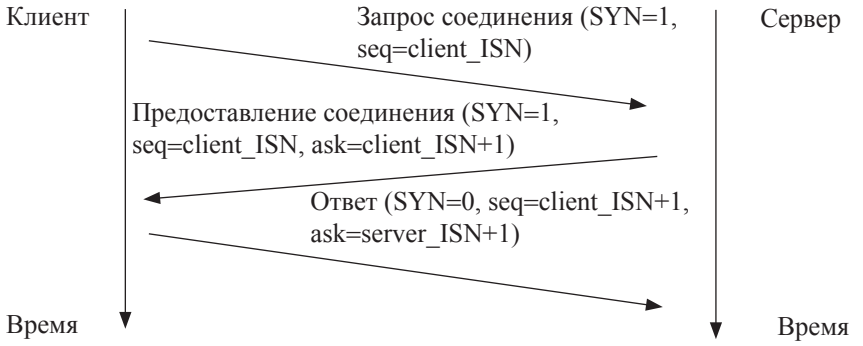


Рис. 3.15. Установка соединения

- Запрашивающая сторона (которая, как правило, называется «клиент») отправляет SYN сегмент, указывая номер порта сервера, к которому клиент хочет подсоединиться, и исходный номер последовательности клиента (ISN).
- Сервер отвечает своим сегментом SYN, содержащим исходный номер последовательности сервера. Сервер также подтверждает приход SYN клиента с использованием ACK (ISN + 1). На SYN используется один номер последовательности.
- Клиент должен подтвердить приход SYN от сервера своим сегментом SYN, содержащим исходный номер последовательности клиента (ISN+1) и с использованием ACK (ISN+1). Бит SYN установлен в 0, так как соединение установлено.

После установления соединения TCP эти два хоста могут передавать данные друг другу. Так как TCP-соединение является полнодуплексным, они могут передавать данные одновременно.

Сервер, получив SYN, откликается посылкой другого SYN. Когда C получает SYN от S (но не получает ACK, например, из-за его потери или злого умысла), он предполагает, что имеет место случай одновременного открытия соединения. В результате он посылает syn_ack, отключает таймер установления соединения и переходит в состояние syn_received. Сервер получает syn_ack от C, но не посылает отклика. Тогда C ожидает получения syn_ack в состоянии syn_received. Так как время пребывания в этом состоянии не контролируется таймером, C может остаться в состоянии syn_received вечно. Из-за того, что переходы из состояния в со-

стояние не всегда четко определены, протокол TCP допускает и другие виды атак

Хотя TCP-соединение является полнодуплексным, при рассмотрении процесса разрыва связи проще его рассматривать как два полудуплексных канала, каждый из которых ликвидируется независимо. Сначала инициатор разрыва посылает сегмент с флагом FIN, сообщая этим партнеру, что не намерен более что-либо передавать (FIN посылается, как правило, в результате вызова приложением функции *close*). Когда получение этого сегмента будет подтверждено (ACK), данное направление передачи считается ликвидированным (реализуется полузакрытие соединения). При этом передача информации в противоположном направлении может беспрепятственно продолжаться. Когда партнер закончит посылку данных, он также пошлет сегмент с флагом FIN. По получении отклика ACK виртуальный канал считается окончательно ликвидированным.

Таким образом, для установления связи требуется обмен тремя сегментами, а для разрыва — четырьмя. Но протокол допускает совмещение первого ACK и второго FIN в одном сегменте, сокращая полное число закрывающих сегментов с четырех до трех.

3.16. Протокол UDP

User Datagram Protocol (UDP) (протокол пользовательских дейтаграмм) является протоколом стандарта TCP/IP, определенным в стандарте RFC 768 «User Datagram Protocol (UDP)». UDP используется вместо TCP для быстрой и ненадежной транспортировки данных между TCP/IP хостами.

UDP протокол обеспечивает обслуживание без установления соединения, таким образом UDP не гарантирует доставку или проверки последовательности для любой дейтаграммы. Хост, который нуждается в надежной связи, должен использовать либо протокол TCP, либо программу, которая будет сама следить за последовательностью дейтаграмм и подтверждать прием каждого пакета.

Автором протокола UDP, созданного в 1980 году, является Дэвид П. Рид.

Чувствительные ко времени приложения часто используют UDP (видеоданные), так как предпочтительнее сбросить пакеты, чем ждать задержавшиеся пакеты, что может оказаться невозможным в системах реального времени. Также потеря одного или нескольких кадров при передаче видео-

данных по UDP не так критична, в отличие от передачи бинарных файлов, где потеря одного пакета может привести к искажению всего файла.

UDP-сообщения инкапсулируются и передаются в IP-дейтаграммы (*рис. 3.16*).

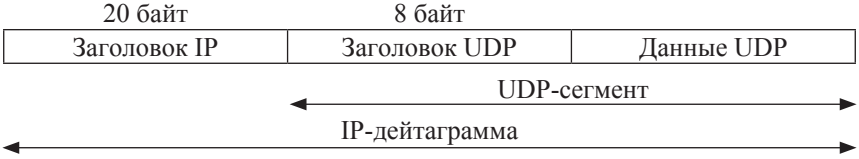


Рис. 3.16. UDP-заголовок

На *рис. 3.17* показаны поля, присутствующие в UDP-заголовке.

Октет	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 1516 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	
0	Порт отправителя	Порт получателя
4	Длина дейтаграммы	Контрольная сумма
	Данные	

Рис. 3.17. Поля, присутствующие в UDP-заголовке

Поля заголовка протокола UDP:

- Порт отправителя — в этом поле указывается номер порта отправителя.
- Порт получателя — это поле обязательно и содержит порт получателя.
- Длина дейтаграммы — поле, задающее длину всей дейтаграммы (заголовка и данных) в байтах. Минимальная длина равна длине заголовка — 8 байт. Теоретически максимальный размер поля — 65 535 байт для UDP-дейтаграммы (8 байт на заголовок и 65 527 на данные). Фактический предел для длины данных при использовании IPv4 — 65 507 (помимо 8 байт на UDP-заголовок требуется еще 20 на IP-заголовок).
- Контрольная сумма — поле контрольной суммы используется для проверки заголовка и данных на ошибки. Если сумма не сгенерирована передатчиком, то поле заполняется нулями.

3.17. Уровень приложений

Прикладной уровень OSI модели обеспечивает сопряжение человека с сетевыми технологиями, что позволяет пользователям общаться между собой через сеть. Другими словами, прикладной уровень создает интерфейс между приложениями конечных устройств при передаче сообщений по сети. Прикладной уровень представляет собой комплекс программных средств, представленных в двух формах: в виде приложений (applications) и программ служб сервиса (services).

Широко известны такие приложения этого уровня, как веб-браузеры гипертекстовой информационной службы (World Wide Web — WWW), которые позволяют людям готовить сообщения для передачи по сети и принимать такие сообщения. Наиболее известными веб-браузерами являются Internet Explorer, Mozilla Firefox, Opera.

Программы служб сервиса готовят данные для передачи по сети, обеспечивая эффективное использование ресурсов сети. Разные типы информации (аудио-, видео-, текстовая информация) требуют различных услуг, поскольку разнотипную информацию необходимо передать через общую сеть.

Протоколы прикладного уровня определяют правила обмена данными между узлом — источником информации и узлом назначения. Каждый вид приложений и сервиса использует свои протоколы, которые определяют стандарты и форматы передаваемых данных.

Протоколы и службы прикладного уровня обычно представлены соответствующими серверами. Однако сервер как отдельное устройство может объединять функции нескольких служб сервиса или, наоборот, служба одного вида услуг может быть представлена многими серверами разного уровня.

Наиболее распространенными протоколами и службами прикладного уровня являются:

- протоколы электронной почты (Simple Mail Transfer Protocol — SMTP, Post Office Protocol — POP, Internet Messaging Access Protocol — IMAP);
- протокол передачи гипертекстовой информации, или веб-сервер (Hypertext Transfer Protocol — HTTP);
- протокол передачи файлов (File Transfer Protocol — FTP) и простой протокол передачи файлов (Trivial FTP — TFTP);

- система доменных имен (Domain Name System — DNS);
- протоколы удаленного доступа (Telnet и SSH), обеспечивающие виртуальное соединение с удаленными сетевыми устройствами;
- протокол динамического назначения адресов узлов (Dynamic Host Configuration Protocol — DHCP).

Существуют две модели построения сети:

1) модель «клиент-сервер»;
2) модель соединения равноправных узлов сети (peer-to-peer).

В сети peer-to-peer связанные через сеть конечные узлы разделяют общие ресурсы (принтеры, файлы) без выделенного сервера. Каждое конечное устройство (peer) может функционировать либо как сервер, либо как клиент. Компьютер может выполнять роль сервера для одного соединения и роль клиента для другого.

Согласно ***модели «клиент-сервер»*** клиент запрашивает информацию, пересылая запрос выделенному серверу (upload), который в ответ на запрос посылает (download) файл, принимаемый клиентом. Следовательно, клиент инициирует процесс обмена информацией в среде «клиент-сервер» и получает от сервера требуемую информацию. Главным достоинством модели «клиент-сервер» является централизация управления сетью и обеспечение безопасности.

Ниже приведены краткие характеристики некоторых наиболее широко используемых протоколов прикладного уровня.

3.18. Протокол динамического выделения адресов (DHCP)

Протокол динамического выделения адресов (DHCP) — это сетевой сервис, который позволяет компьютерам в сети автоматически получать настройки с сервера вместо того, чтобы настраивать каждый сетевой хост вручную.

В общем случае настройки, передаваемые DHCP-сервером DHCP-клиентам, включают:

1) IP-адрес и сетевую маску;
2) IP-адрес шлюза по умолчанию;
3) IP-адрес DNS-серверов.

Однако DHCP-сервер может также предоставить такие параметры настройки, как:

1) имя хоста;
2) имя домена;

3) адрес сервера времени;
4) адрес сервера печати.

DHCP построен по схеме клиент-сервер, где DHCP-сервер выделяет сетевые адреса и доставляет конфигурационные параметры динамически конфигурируемым ЭВМ.

DHCP-сервер может выделять IP-адреса, используя 3 описанных ниже метода.

Выделение вручную (по MAC-адресу)

Этот метод подразумевает использование DHCP для определения уникального аппаратного адреса каждой сетевой карты, подключенной к сети, и затем продолжительного предоставления неизменной конфигурации каждый раз, когда DHCP клиент делает запрос на DHCP-сервер, используя это сетевое устройство. Это гарантирует, что определенный адрес будет автоматически присваиваться этой сетевой карте на основе ее MAC-адреса.

Динамическое выделение (пул адресов)

При этом методе, DHCP-сервер будет выделять IP-адрес из пула адресов (иногда называемых диапазоном или областью) на период времени (или в аренду), который настраивается на сервере, или пока клиент не проинформирует сервер, что больше вообще не нуждается в адресе. Таким образом, клиенты получают свои настройки динамически по принципу «первый пришел — первый обслужился». Когда DHCP-клиент отсутствует в сети определенное время, настройка считается просроченной и возвращается в пул адресов для использования другими DHCP-клиентами. Это означает, что адрес арендуется или выдается на определенный период времени. По истечении этого периода клиент должен повторно договариваться об использовании адреса с сервером.

Автоматическое выделение

Используя этот метод, DHCP автоматически присваивает постоянный IP-адрес устройству, выбранному из пула доступных адресов. Обычно DHCP используется для выдачи временного адреса, но DHCP-сервер может использовать бесконечное время аренды.

Динамическое присвоение адресов представляет собой единственный механизм, позволяющий автоматически повторно использовать адрес, который не нужен клиенту.

3.19. Протоколы передачи электронной почты

При передаче электронной почты и взаимодействии почтовых серверов между собой используется простой протокол передачи почты (Simple Mail Transfer Protocol — SMTP), у которого номер порта 25. Для получения клиентом сообщения с сервера используется протокол почтового отделения (Post Office Protocol — POP) с номером порта 110 или протокол доступа к сообщениям (Internet Messaging Access Protocol –IMAP), см. *рис. 3.18*.

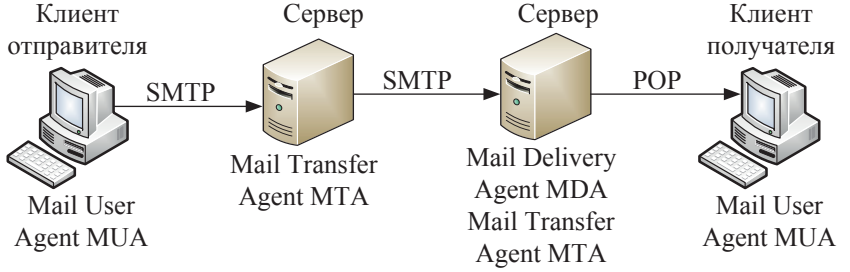


Рис. 3.18. Передача электронной почты по сети

При пересылке почты от клиента на сервер используется протокол SMTP, при этом происходит процесс *upload*.

Когда почтовый сервер получает сообщение, предназначенное для клиента, он хранит это сообщение и ждет, когда адресат назначения заберет свою почту. Почтовые клиенты забирают сообщения (процесс *download*), используя один из сетевых протоколов. Самые популярные почтовые протоколы клиента — POP3 и IMAP4, которые на транспортном уровне применяют протокол TCP для надежной доставки данных.

Почтовые серверы общаются друг с другом, используя протокол SMTP, который транспортирует почтовые сообщения в текстовом формате, используя TCP. Протокол SMTP характеризуется низким уровнем защиты информации, поэтому серверы предоставляют услуги только пользователям своей сети.

В процессе подготовки электронной почты используется клиентское приложение, называемое *агентом пользователя* (Mail User Agent–MUA). Приложение MUA позволяет посылать сообщения и помещать полученные сообщения в почтовый ящик клиента.

При передаче сообщений между серверами используется Агент передачи почты (Mail Transfer Agent–MTA). Агент MTA получает сообщения от MUA или от другого MTA и передает их по сети. Агенты MTA применяют протокол SMTP для передачи электронной почты между серверами. Если сообщение из сервера может быть отправлено сразу клиенту локальной сети, то подключается Агент доставки почты (Mail Delivery Agent — MDA). Агент MDA получает прибывающую почту от MTA и помещает ее в соответствующие почтовые ящики пользователей, используя протокол POP.

3.20. Протокол HTTP

Самым распространенным протоколом прикладного уровня в настоящее время является протокол передачи гипертекстовой информации (Hypertext Transfer Protocol — HTTP), который работает в сети Интернет. Его основным приложением является веб-браузер, который отображает данные на веб-страницах, используя текст, графику, звук и видео.

Веб-страницы создаются с применением языка разметки гипертекста Hypertext Markup Language (HTML), который определяет местоположения для размещения текста, файлов и объектов, которые должны быть переданы от сервера по сети до веб-браузера. Номер порта протокола HTTP — 80, он функционирует совместно с протоколом транспортного уровня TCP.

В ответ на запрос сервер посылает клиенту сети текст, аудио-, видео- и графические файлы, указанные в командах HTML. Браузер клиента повторно собирает все файлы, чтобы создать изображение веб-страницы, которая представляется пользователю.

Протокол HTTP характеризуется сравнительно невысоким уровнем безопасности, поскольку передаваемые по сети сообщения не зашифрованы. Для повышения уровня безопасности передачи сообщений через Интернет был разработан протокол HTTP Secure (HTTPS). В этом протоколе используется процесс криптографирования данных и аутентификации, что существенно повышает уровень безопасности. Номер порта протокола HTTPS — 443.

3.21. Протоколы передачи файлов FTP и TFTP

Протокол передачи файлов (File Transfer Protocol — FTP) — служба, ориентированная на предварительное соединение (connection-oriented), которая взаимодействует с протоколом транспортного уровня TCP. Главная цель протокола FTP состоит в том, чтобы передавать файлы от одного компьютера другому или копировать и перемещать файлы от серверов клиентам и от клиентов серверам. Это является главным отличием от протокола HTTP, который позволяет клиенту «скачивать» файлы с сервера, но не позволяет пересылать файлы на сервер.

Протокол передачи файлов FTP сначала устанавливает соединение между клиентом и сервером, используя команды запроса клиента и ответы сервера. При этом номер порта — 21. Затем производится обмен данными, когда номер порта — 20. Передача данных может производиться в режиме кода ASCII или в двоичном коде. Эти режимы определяют кодирование, используемое для файла данных, которое в модели OSI является задачей представительского (presentation) уровня. После завершения передачи файла соединение для передачи данных заканчивается автоматически. Управление сеансом связи происходит на сеансовом (Session) уровне.

Простой протокол передачи файлов (Trivial File Transfer Protocol — TFTP) — служба без установления соединения (connectionless), которая работает совместно с протоколом транспортного уровня (User Datagram Protocol — UDP).

Протокол TFTP применяется на маршрутизаторах, чтобы передавать файлы конфигурации и операционную систему, а также для передачи файлов между системами, которые поддерживают TFTP. Протокол TFTP характеризует простота и малый объем программного обеспечения. Он может читать или записывать файлы при соединении с сервером, но не ведет списки и каталоги.

3.22. Система доменных имен DNS

Система доменных имен (Domain Name System — DNS) используется в Интернете для того, чтобы переводить имена сайтов или доменов в числовые значения IP-адреса. Поскольку в ряде случаев требуется знание числового адреса, хост может обратиться к DNS-серверу и по имени получить соответствующий адрес. DNS использует распределенный набор серверов

разного уровня иерархии, чтобы получить соответствие между именем и числовым адресом.

Операционные системы компьютеров содержат утилиту *nslookup*, которая позволяет пользователю вручную запрашивать имя сервера и идентифицировать название хоста. Когда клиент делает запрос, локальный сервер сначала проверяет собственные записи. Если соответствующих пар «имя-адрес» у него нет, то он связывается с другими серверами DNS более высокого уровня иерархии.

Служба прикладного уровня DNS характеризуется номером порта 53 и взаимодействует как с протоколом транспортного уровня TCP, так и с протоколом UDP.

3.23. Протокол удаленного доступа Telnet

Протокол Telnet обеспечивает виртуальное соединение пользователя с удаленными сетевыми устройствами: компьютерами, маршрутизаторами, коммутаторами. Чтобы осуществить подключение клиента по протоколу Telnet, обычно задают имя удаленного хоста. В качестве имени хоста используется IP-адрес или имя доменной системы DNS удаленного устройства. Вся обработка информации и использование памяти производятся на процессоре удаленного устройства, а отображение результатов конфигурирования протокол Telnet транслирует на монитор пользователя. Telnet работает на прикладном уровне модели TCP/IP, поэтому охватывает все уровни модели OSI. Номер порта — 23.

TELNET (Terminals NET work) — стандартный протокол TCP/IP для услуг виртуального терминала. TELNET дает возможность устанавливать соединение с удаленным компьютером таким образом, что создается впечатление, как будто местный терминал — это терминал удаленной системы.

TELNET был разработан в эпоху, когда большие операционные системы, такие как UNIX, работали с внешней средой по принципу разделения времени. Согласно этому принципу, большой компьютер поддерживал множество пользователей, предоставляя им часть общего времени. Взаимодействие между пользователем и компьютером осуществляется с помощью терминала, который обычно состоит из комбинации клавиатуры, монитора и мышки. Даже микрокомпьютер может моделировать терминал с помощью терминального эмулятора.

В среде с разделением времени вся обработка информации проводится в центральном компьютере. Когда пользователь печатает символ на клавиатуре, символ обычно посылается компьютеру и отражается на мониторе. Разделение по времени создается средой, в которой для каждого пользователя создается иллюзия специализированного компьютера. Пользователь выполняет программу доступа к системным ресурсам, переключается от одной программы к другой и так далее.

Логин

В среде с разделением времени пользователь — это часть системы с некоторыми правами. Каждый полномочный пользователь имеет идентификатор и пароль. Пользовательская идентификация определяет пользователя как часть системы. Для доступа к системе пользователь начинает сеанс с пользовательского идентификатора (id) или с регистрационного имени (login name). Система помогает проверке пароля, чтобы предотвратить доступ к ресурсу неполномочного пользователя.

Местный логин

Когда пользователь входит в местную систему с разделением времени, это называется местный логин. Как только пользователь напечатает некое слово на терминале или рабочей станции, выполняющей эмуляцию терминала, сразу начинает работать терминальная программа (драйвер), которая распознает значение введенных символов. Терминальный драйвер передает символы операционной системе, в рамках этой системы комбинация символов интерпретируется и вызывает желаемую прикладную программу или утилиту (*рис. 3.19*).

Дистанционный логин

Когда пользователь хочет иметь доступ к прикладной программе или утилите, размещенным на удаленном компьютере, он выполняет дистанционный вход в систему (логин), см. *рис. 3.20*. Протокол TELNET берет на себя функции клиента и сервера. Пользователь посылает сигнал нажатия кнопки терминальному драйверу, где местная операционная система принимает символы и интерпретирует их. Эти символы посылает TELNET-

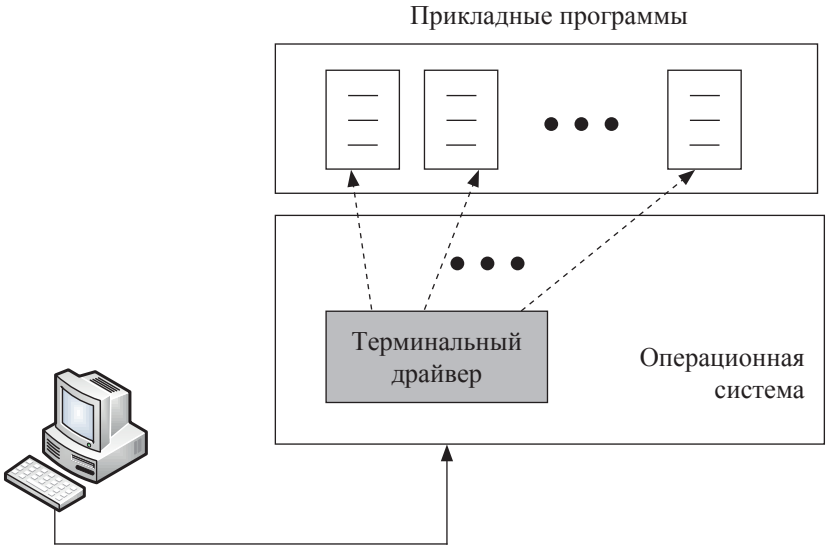


Рис. 3.19. Местный логин

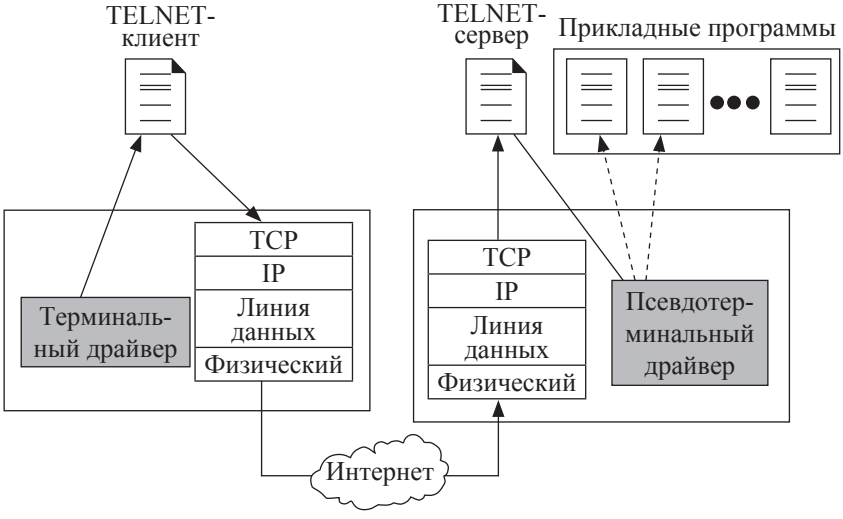


Рис. 3.20. Дистанционный login

клиент, который преобразует символы к универсальному набору, называемому символом виртуального сетевого терминала (Network Virtual Terminal Characters), и доставляет их к местному стеку протоколов TCP/IP.

Команды или текст в форме сетевого виртуального терминала (NVT) перемещаются через Интернет и прибывают на стек протоколов TCP/IP в удаленной машине. Здесь символы доставляются операционной системе и проходят к TELNET-серверу, который преобразует их в символы, понятные удаленному компьютеру. Однако символы не могут пройти прямо на операционную систему, потому что удаленная операционная система не разработана для получения трактовки этих символов от TELNET. Она спроектирована так, чтобы принимать символы от драйвера терминала. Решение, добавляющее необходимое программное обеспечение, называется псевдотерминальным драйвером, который преобразовывает поступившие символы как символы, поступающие от местного терминала. Операционная система затем предает символы к соответствующей прикладной программе (*рис. 3.20*).

Протокол Telnet поддерживает аутентификацию, поэтому на удаленном устройстве задается пароль, который должен знать пользователь. Однако данный протокол не поддерживает криптографирование данных, которые передаются по сети как простой текст. Это означает, что данные могут быть перехвачены. Для защиты передаваемой информации разработан протокол SSH (Secure Shell). Он обеспечивает криптографирование данных и более надежную аутентификацию.

Контрольные вопросы и задания

1. Приведите основные урони стека протоколов TCP/IP.
2. Приведите примеры протоколов канального уровня.
3. Приведите примеры протоколов сетевого уровня.
4. Приведите примеры протоколов транспортного уровня.
5. Сколько двоичных разрядов содержат логические адреса узлов в IP-сетях версии IPv4?
6. Что определяют старшие и младшие разряды сетевого адреса?
7. Какие классы уникальных адресов используются в сетях?
8. Какие размеры имеют стандартные маски адресов классов A, B, C?
9. Какое максимальное число узлов могут задавать адреса класса C?
10. Как называется общая часть адреса нескольких устройств?

11. Каковы две формы программных средств прикладного уровня?
12. Где находятся основные ресурсы сети модели «клиент-сервер»?
13. Где находятся основные ресурсы сети модели «peer-to-peer»?
14. Назовите протоколы передачи электронной почты.
15. Какие функции выполняет протокол HTTP?
16. В чем различие между протоколами HTTP и HTTPS?
17. В чем различие между протоколом FTP и HTTP?
18. Для чего используется система доменных имен DNS?
19. Какие протоколы обеспечивают виртуальное соединение пользователя с удаленными сетевыми устройствами?
20. Какой протокол обеспечивает динамическое назначение адресов узлов?
21. В чем различие между протоколами TCP и UDP?
22. За сколько этапов выполняется предварительное установление соединения у протокола TCP?
23. Чем определяется размер поля данных сегмента?

СЛОВАРЬ ОСНОВНЫХ ТЕРМИНОВ

Адаптер (лат. *adapto* —приспособляю), или контроллер — устройство подключения блоков компьютера к системной шине.

Адаптер сетевой — внешний интерфейс компьютера, устройство его сопряжения с каналом связи; в глобальных сетях функции сетевого адаптера выполняет модем.

Базовая система ввода/вывода, BIOS (англ. Basic Input/Output System) — аппаратно-программный модуль операционной системы MS DOS, управляющий внешними устройствами компьютера. При включении персонального компьютера осуществляет поиск и загрузку с диска в оперативную память программы-загрузчика ОС, тестирование аппаратной части, а также инициализацию системы прерываний нижнего уровня. Хранится в ПЗУ, содержит программу начальной загрузки ОС, тестовые программы и драйверы стандартных внешних устройств.

Блок системный — металлический корпус, в котором размещены процессор, оперативная память, накопители на жестких и гибких дисках, блок питания, адаптеры внешних устройств и др. элементы компьютера.

Бод — единица скорости передачи дискретной информации по каналу связи, измеряемая в бит/с.

Браузер (англ. *browser*) — программа просмотра Web-страниц, обеспечивающая переход на другой объект по гиперссылке.

Видеоадаптер (видеоконтроллер) — устройство управления дисплеем и выводом информации на его экран; находится на видеокарте, устанавливаемой в разъем материнской платы, и включает в себя схему управления экраном, видеопамять (растровую память) для хранения воспроизводимой на экране информации, сменные микросхемы ПЗУ (матрицы знаков) и порты ввода/вывода.

Виртуальная машина — (лат. *virtualis* — способный, возможный) — воображаемая машина, предоставляемая пользователю операционной системой.

Виртуальная память — воображаемая память, выделяемая операционной системой для размещения пользовательской программы, ее рабочих полей и информационных массивов.

Всемирная паутина (англ. WWW — World Wide Web) — совокупность информационных источников, находящихся на разных компьютерах в виде гипертекстов, которые представляют собой Web-страницы.

Гиперссылка — выделенный объект гипертекста, связывающий его с другим информационным источником и реагирующий на щелчок мыши.

Гипертекст — текст, документ, представляющий собой сочетание алфавитно-цифровой информации в различных форматах и стилях с графическими изображениями, аудио- и видеоинформацией; содержит гиперссылки, связывающие его с другими источниками.

Декодирование — процесс, обратный кодированию, т.е. восстановление чисел и слов по соответствующим комбинациям символов.

Декодер, дешифратор (франц. *decoder, dechiffrer* — расшифровывать, разбирать) — логическое устройство, преобразующее коды входных сигналов в однозначно соответствующие им выходные сигналы. Применяется в компьютерах и вычислительных системах для преобразования кода операции команды в управляющий сигнал, кода адреса в сигнал выборки соответствующей ячейки запоминающего устройства, а также для распределения сигналов по цепям управления, выборки требуемых каналов связи и т.д.

Демодулятор (франц. *démodulateur*) — электронный узел устройств, отделяющий полезный (модулирующий) сигнал от несущей составляющей, в зависимости от вида которой (гармоническая или импульсная) и типа модуляции демодуляторы подразделяют на амплитудные, частотные и фазовые или на амплитудно-, частотно-, фазо- и широтно-импульсные.

Диск — машинный носитель информации с прямым доступом. Различают накопители на магнитных и оптических дисках.

Дискета (англ. *diskette*) — сменный носитель информации на гибком магнитном диске в виде неразборной съемной кассеты.

Дисковод — устройство чтения/записи информации на диске.

Доступ (к информации) — возможность использования информации, хранящейся в ЭВМ или системе.

Драйвер — программа, осуществляющая форматное преобразование данных при обмене информацией между основной памятью компьютера и соответствующим внешним устройством.

Запись — неоднородная упорядоченная статическая структура прямого доступа.

Запоминающее устройство (ЗУ) — устройство для приема, хранения и выдачи информации в компьютерах и вычислительных системах. Оно состоит из накопителя, блоков приема, записи, выборки, считывания, выдачи числа и местного управления. По характеру обращения к ЗУ различают адресные, безадресные и ассоциативные ЗУ; по способу выборки информации из отдельных ячеек — ЗУ с произвольным, последовательным и циклическим обращением; по функциональному назначению — ОЗУ, ПЗУ, внешнее, буферное, магазинное ЗУ и т.п. В зависимости от типа ЗУ возможно совмещение функций приема и выдачи числа в одном блоке, отсутствие блоков приема и записи числа в долговременном ПЗУ и т.д.

Инсталляция (англ. *installation* — размещение, ввод в действие) — установка программного продукта, при которой он размещается на магнитном диске. Напр., при инсталляции принтера устанавливается его драйвер (один раз при подключении к компьютеру новой модели принтера).

Информации передача — процесс ее распространения от источника к потребителю. Полная информационная модель передачи информации в реальных условиях включает в себя передатчик, кодирующее устройство, кодер канала, канал связи, декодер канала, декодирующее устройство и приемник.

Информации передачи теория — часть теории информации, основы которой разработал К. Шеннон, решивший проблему нахождения максимально достижимой скорости передачи информации при сколь угодно малой вероятности ошибок, связав ее с количеством информации.

Информации плотность записи — количество информации в битах, записываемой на единицу площади или объема запоминающей среды.

Информации регенерация (лат. *regenero* — возрождаю) — восстановление информации в вычислительных устройствах для ее длительного сохранения.

Информации сжатие — процесс преобразования информации для уменьшения избыточности в ее представлении, применяется для компактного размещения информации и сокращения времени ее передачи по каналам связи в компьютерных сетях.

Информации стирание — процесс перехода запоминающей среды в состояние, при котором параметры запоминающих элементов становятся идентичными. Процесс стирания по воздействию на среду противоположен записи информации. Стирание может происходить самопроизвольно или под влиянием окружающей среды и принудительно при воздействии технических средств для повторной записи. Иногда стирание происходит при считывании информации, тогда необходима регенерация информации.

Информации считывание, чтение — процесс распознавания состояния запоминающей среды. В некоторых типах ЗУ этот процесс осуществляется автоматически без внесения изменений в текущее состояние, т.е. без разрушения хранимой информации.

Информации теория — раздел кибернетики, занимающийся математическим описанием и оценкой методов передачи, хранения, получения и классификации информации, представляет собой совокупность теорий, общими для которых являются методы теории вероятностей, что отражает присутствие в процессах случайных факторов, связанных с информацией. Ее составными частями являются теория передачи информации, теория кодирования и многие задачи математической статистики.

Информационная модель — совокупность информации, описывающей свойства и состояния объекта, процесса или явления, а также их связь с внешним миром; различают знаковые и вербальные (лат. verbalis — «устный») информационные модели.

Информационная система — совокупность методов и средств ввода, хранения, обработки и вывода информации, а также персонала, используемых для принятия обоснованных решений в интересах достижения поставленной цели.

Информационная технология — совокупность методов и программно-технических средств, объединенных в технологическую цепочку, обеспечивающую сбор, хранение, обработку, передачу и отображение информации.

Информация документальная — информация, закрепленная на какомлибо материальном носителе.

Канал машинный — совокупность технических средств обмена информацией между центральным процессором и внешними устройствами компьютера.

Канал связи — технические средства, которые обеспечивают распространение электрических сигналов по линиям связи от передатчика к приемнику.

Канала связи пропускная способность — верхнее значение скорости передачи информации по всем возможным распределениям сообщений на входе канала. Распределение вероятностей сообщений на выходе канала при известном сообщении на его входе является фиксированной характеристикой канала.

Кластер (англ. *cluster* — группа) — минимальная единица размещения информации на диске, содержит один или несколько смежных секторов дорожки.

Код (франц. *code*) — множество кодовых слов, которое поставлено в соответствие некоторым элементам сообщений. Число символов в кодовом слове называется его длиной, число букв алфавита — основанием кода, если основание равно двум, код называют двоичным. Коды бывают равномерными и неравномерными в зависимости от того, одинаковую ли длину имеют кодовые слова.

Кода избыточность — величина, определяемая как разность между длиной равномерного кода и числом информационных знаков кодовой комбинации; относительная избыточность определяется отношением избыточности кода к числу информационных знаков.

Кодер, шифратор (франц. *coder, chiffrer* — нумеровать, шифровать) — логический блок, преобразующий комбинации входных сигналов в комбинации выходных сигналов, эквивалентные исходным.

Кодирование — универсальный способ отображения информации, задаваемый взаимно однозначным соответствием между элементами сообщений (например, буквами печатного текста) и словами (конечными наборами символов), с помощью которых они фиксируются в данном алфавите. Другими словами, под кодированием информации можно понимать представление чисел и слов соответствующими им комбинациями символов.

Кодирования теория — раздел теории информации, изучающий коды, отображающие сообщения заданного вида в слова из символов некоторого алфавита, а также задачи кодирования и декодирования со-

общений, посылаемых источниками информации и поступающих к потребителям.

Коды корректирующие — коды, используя которые можно обнаруживать и исправлять некоторые ошибки в символах кодовых слов. Основным принципом построения таких кодов является введение избыточности кода, когда часть символов кодовой комбинации можно использовать для обнаружения и исправления ошибок. Двоичные коды характеризуются минимальным кодовым расстоянием (числом несовпадений соответствующих символов) между кодовыми словами, которое определяет корректирующую способность кода — максимальное число обнаруживаемых и исправляемых ошибок в символах или вероятность обнаружения и исправления различных ошибок.

Коммуникация (лат. *communicatio* — сообщение, передача) — процесс обмена информацией, в котором различают технический, семантический и прагматический аспекты. Техническую сторону связывают с передачей информации по каналам связи, что рассматривается методами теории информации. Семантический аспект отражает передачу и прием информации, включая ее понимание получателем. Прагматический аспект учитывает влияние принятой информации на поведение получателей и эффективность ее использования.

Линия связи — физическая среда, обеспечивающая передачу информации.

Машинное слово — последовательность единиц информации — битов, представляющих двоично-цифровую информацию, полубайтов (десятичные цифры) или байтов (буквенно-символьная информация). Обычно машинное слово занимает одну ячейку памяти компьютера, и при обращении к нему устройства оперируют с ним как с единым целым.

Многоканальная система — система обработки данных, осуществляющая обмен информацией со многими потребителями посредством каналов связи, разновидностью которой являются вычислительные системы.

Модем (модулятор-демодулятор) — устройство в составе аппаратуры автоматической передачи данных по каналам связи, осуществляет преобразование для передачи по линиям связи (модуляцию) и обратное преобразование (демодуляцию) — при приеме.

Модулятор (лат. *modulator* — соблюдающий ритм) — устройство, изменяющее параметр несущего сигнала под действием информационного

(модулирующего) сигнала. В зависимости от того, какой из параметров несущего сигнала (а им могут быть гармонические или импульсные колебания) модулируется, различают амплитудные, частотные и фазовые, а также амплитудно-, частотно-, фазо- и широтно-импульсные модуляторы.

Накопитель — блок запоминающего устройства компьютера, предназначенный для хранения информации. Представляет собой упорядоченную совокупность запоминающих ячеек, в каждой из которых хранится одно машинное слово.

Накопитель на гибких магнитных дисках — то же, что дискета.

Накопитель на жестких магнитных дисках типа винчестер — один или несколько жестких дисков и блок магнитных головок записи/чтения в герметически закрытом корпусе; в последнее время в них стал использоваться метод зонной записи, что увеличивает емкость жестких дисков примерно на треть.

Обращение к запоминающему устройству — этап работы запоминающего устройства по извлечению из него или введению в него информации. Различные способы обращения к ЗУ влияют на структуру компьютера в целом и определяют тип ЗУ, а скорость работы последнего характеризуется временем обращения к ЗУ.

Оперативное запоминающее устройство (ОЗУ, RAM) (лат. *operor* — действую, обрабатываю и англ. Random Access Memory — память с произвольным доступом) — запоминающее устройство, предназначенное для записи, хранения и выдачи информации, непосредственно участвующей в вычислительном процессе при функционировании компьютера. ОЗУ состоит из больших интегральных схем, содержащих матрицы запоминающих элементов; последние расположены на пересечении горизонтальных и вертикальных шин матрицы, подачей импульсов по которым осуществляется запись и считывание информации, причем она теряется при отключении напряжения питания.

Отношение сигнал-помеха — отношение основной характеристики полезного сигнала, например эффективного напряжения, к соответствующей характеристике помехи, используется при оценке помехоустойчивости систем связи, управления, вычислительной техники и т.д.

Очередь — однородная линейно упорядоченная динамическая структура данных последовательного доступа.

Постоянное запоминающее устройство (ПЗУ, ROM) (англ. *Read-Only Memory* — память только для чтения) — энергонезависимое ЗУ, предназначенное для хранения постоянной информации, в частности, программ базовой системы вво-да/вывода.

Протокол — совокупность единых правил передачи данных по каналам связи в компьютерной сети; регулирует взаимодействие параллельно функционирующих объектов сети, обеспечивая их согласованное поведение.

Регистровая кэш-память (англ. *cache* — тайный склад) — быстродействующая буферная память большой емкости на регистрах между микропроцессором и ОЗУ для увеличения скорости выполнения операций в компьютере; недоступна для пользователя — отсюда и ее название.

Сверхоперативное запоминающее устройство (СОЗУ) — быстродействующее ЗУ для объединения функций нескольких регистров процессора, а также для временного хранения часто используемых данных, констант, коротких подпрограмм и промежуточных результатов; является разновидностью буферного ЗУ. Оно используется при существенном различии в скоростях работы процессора и ОЗУ, предназначено для немедленного предоставления процессору тех блоков информации, которые необходимо обработать в данный момент. В СОЗУ запоминается самая последняя информация, выбранная из ОЗУ, вместе с непосредственно примыкающими данными, исходя из предположения, что данные этого блока скоро снова будут нужны процессору. Обмен блоками информации между СОЗУ и ОЗУ производится аппаратно.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. 3DNews: Daily Digital Digest. Новости программного и аппаратного обеспечения [Электронный ресурс]. URL: http://3dnews.ru/ (дата обращения 01.08.2017).
2. BYTE/Россия [Электронный ресурс]. URL: http://www.bytemag.ru (дата обращения 01.08.2017).
3. CITforum. Аналитическая информация по всем областям компьютерной сферы [Электронный ресурс]. URL: http://www.citforum.ru (дата обращения 01.08.2017).
4. iXBT.com. Русскоязычное интернет-издание о компьютерной технике, информационных технологиях и программных продуктах [Электронный ресурс]. URL: http://www.ixbt.com (дата обращения 01.08.2017).
5. PC-Magazine [Электронный ресурс]. URL: http://ru.pcmag.com (дата обращения 01.08.2017).
6. *Бройдо В.Л., Ильина О.П.* Вычислительные системы, сети и телекоммуникации: учеб. пособие. – 4-е изд. – СПб.: Питер, 2011. – 560 с.
7. *Брукс Ч.Д.* CompTIA A+. Установка, настройка, обслуживание и ремонт ПК: учеб. пособие. – 3-е изд. – СПб.: БХВ-Петербург, 2010. – 1215 с.
8. *Горнец Н.Н.* и др. Организация ЭВМ и систем. – М.: Академия 2006. – 320 с.
9. *Жмакин А.П.* Архитектура ЭВМ. – 2-е изд. – СПб.: БХВ-Петербург, 2010. – 352 с.
10. Компьютерный форум Ru.Board [Электронный ресурс]. URL: http://forum.ru-board.com (дата обращения 01.08.2017).
11. Компьютер-Пресс [Электронный ресурс]. URL: http://compress.ru (дата обращения 01.08.2017).

12. *Кузин А.В., Жаворонков М.А.* Микропроцессорная техника: учебник для студ. сред. проф. образования. – 6-е изд., испр. – М.: Академия, 2011. – 304 с.
13. *Максимов Н.В.* и др. Архитектура ЭВМ и вычислительных систем: учебник. – 5-е изд., перераб. и доп. – М.: Форум, Инфра-М, 2013. – 512 с.: ил. – (Профессиональное образование).
14. Мир nVidia. Портал новостей, обзоров и статей об аппаратном и программном обеспечении [Электронный ресурс]. URL: http://nv-world.ru/. (дата обращения 01.08.2017).
15. Мир ПК [Электронный ресурс]. URL: http://journal-off.info/tags. (дата обращения 01.08.2017).
16. *Мэлоун Д., Мэрфи Н.* IPv6. Администрирование сетей: учебник. – М.: КУДИЦ-ОБРАЗ, 2007. – 320 с.
17. Национальный открытый университет ИНТУИТ [Электронный ресурс]. URL: http://www.intuit.ru. (дата обращения: 01.08.2017).
18. *Олифер В.Г.* Компьютерные сети: принципы, технологии, протоколы: учебник для вузов. – 4-е изд. – СПб.: Питер, 2010. – 945 с.
19. Программные продукты и системы [Электронный ресурс]. URL: http://www.swsys.ru (дата обращения 01.08.2017).
20. *Пятибратов А.П., Гудыно Л.П., Кириченко А.А.* Вычислительные системы, сети и телекоммуникации: учеб. пособие /под. ред. А.П. Пятибратова. – М.: КНОРУС, 2013. – 376 с.
21. Системный администратор [Электронный ресурс]. URL: http://sa-mag.ru (дата обращения 01.08.2017).
22. *Суворов А.Б.* Телекоммуникационные системы, компьютерные сети и Интернет: учеб. пособие. – Ростов н/Д: Феникс, 2007. – 384 с.: ил. + библ., список англояз. сокращ. – (Высшее образование).
23. *Таненбаум Э., Уэзеролл Д.* Компьютерные сети: учебное пособие по компьютерным сетям. – 5-е изд. СПб.: Питер, 2012. – 960 с.
24. Хабрахабр [Электронный ресурс]. URL: http://habrahabr.ru (дата обращения: 01.08.2017).
25. *Чеппел Л., Титтел Э.* TCP/IP. Учебный курс. – СПб.: БХВ-Петербург, 2003. – 960 с.
26. *Щербатов И.А.* Вычислительные системы, сети и телекоммуникации: учеб. пособие для вузов /Астраханск. гос. технич. ун-т. – Астрахань: АстГТУ, 2011. – 191 с.

Учебное издание

Елена Валерьевна **Петрунина**,
Оксана Николаевна **Савельева**,
Татьяна Валерьевна **Гончарук**

КОМПЬЮТЕРНЫЕ СЕТИ

Учебное пособие

Ответственный редактор	С.А. Бобко
Редактор/корректор	Ю.Ф. Кравчинская
Технический редактор	К.А. Антонов
Компьютерная верстка	К.А. Антонов
Дизайн обложки	А.В. Свешников

Подписано в печать 06.12.2017. Формат 60x84 $^{1}/_{16}$.
Бумага офисная. Гарнитура *Times New Roman*. Печ. лист 7.
Тираж 66 экз. Заказ № 40.

Московский государственный гуманитарно-экономический университет
107150, Москва, ул. Лосиноостровская, д. 49.
Отпечатано в типографии МГГЭУ по технологии CtP.