

## Раздел 1. ОБЩИЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ СЕТЕЙ ЭВМ

### 1.1. Основные понятия и терминология

В данном параграфе вводятся используемые при изложении материала понятия и определения, позволяющие систематизировать средства вычислительной техники и средства телекоммуникаций, являющиеся объектами изучения в дисциплине "Сети ЭВМ и телекоммуникации".

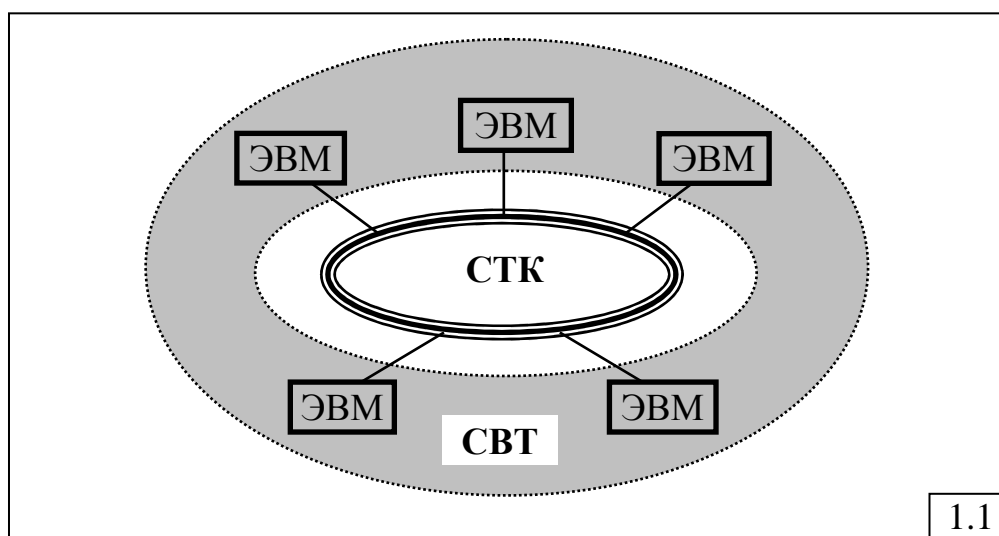
К сожалению, для вычислительной техники характерна терминологическая неоднозначность и неопределённость, что проявляется в различном толковании в разных литературных источниках одного и того же термина (например «вычислительная система» или «вычислительный комплекс»), либо в использовании разных терминов для обозначения одного и того же понятия (например «вычислительная сеть», «сеть ЭВМ», «компьютерная сеть»). Всё это зачастую усложняет восприятие и усвоение материала.

Целью излагаемого в данном разделе материала является устранение терминологической неоднозначности и уточнение используемых ниже терминов и понятий. Предлагаемая классификация различных систем и объектов вычислительной техники направлена на выявление классов систем, характеризующихся одинаковыми или близкими свойствами, что позволяет унифицировать процесс изучения и исследования вычислительных систем и сетей.

#### 1.1.1. Понятие сети ЭВМ

*Сеть ЭВМ* (рис.1.1) – совокупность *средств вычислительной техники* (СВТ), представляющих собой множество ЭВМ, объединённых с помощью *средств телекоммуникаций* (СТК). Сеть ЭВМ реализует две основные функции:

- обработку данных;
- передачу данных.



Наряду с термином «сеть ЭВМ» широко используются близкие по смыслу термины «компьютерная сеть» и «вычислительная сеть», которые обычно рассматриваются как синонимы. Однако некоторые незначительные различия между указанными терминами мы будем иметь в виду при дальнейшем изложении материала.

Из данного выше определения (рис.1.1) следует, что «сеть ЭВМ» представляет собой множество ЭВМ (компьютеров), объединённых в единую сеть с помощью средств телекоммуникаций, образующих **базовую сеть передачи данных (СПД)**. Другими словами, «сеть ЭВМ» или «компьютерная сеть» – это объединение ЭВМ (компьютеров), в отличие, например, от телефонной сети, объединяющей автоматические телефонные станции (АТС). Поэтому эти два термина будем рассматривать и использовать ниже как эквивалентные. Термин же «вычислительная сеть» скорее характеризует назначение сети – выполнение вычислений, что отличает её, например, от «информационной сети», предоставляющей информационные услуги, или от «телекоммуникационной сети», предназначенной для передачи данных.

Отдельные сети ЭВМ могут объединяться между собой, образуя большие компьютерные сети, которые в свою очередь могут объединяться и образовывать сверхбольшие глобальные сети. Такое объединение сетей приводит к иерархической структуре, в которой небольшие сети являются подсетями сетей более высокого ранга.

Итак, сеть ЭВМ реализует передачу и обработку *данных*. Однако часто можно услышать или прочитать, что в сети передаётся и обрабатывается *информация*. Так что же на самом деле передаётся и обрабатывается в сети: данные или информация? Для ответа на этот вопрос необходимо определить понятия «данные» и «информация».

Существуют различные подходы к определению понятий «данные» и «информация» в разных областях человеческой деятельности: в биологии, в кибернетике, в философии и т.д. Создана даже специальная научная дисциплина «Теория информации».

Среди всех существующих определений понятий «данные» и «информация» воспользуемся общепринятыми традиционными определениями, для чего обратимся к «Словарю русского языка» С.И.Ожегова, и попытаемся сформулировать разницу между этими двумя терминами.

### 1.1.2. Данные и информация

*«Данные* – сведения, необходимые для какого-нибудь вывода, решения.

*Информация* – сведения, осведомляющие о положении дел, о состоянии чего-нибудь». (Ожегов С.И. Словарь русского языка).

Из этих определений следует, что данные – это любое множество сведений, а информация – это сведения, полученные с некоторой целью и несущие в себе новые знания для того, кто эту информацию получает.

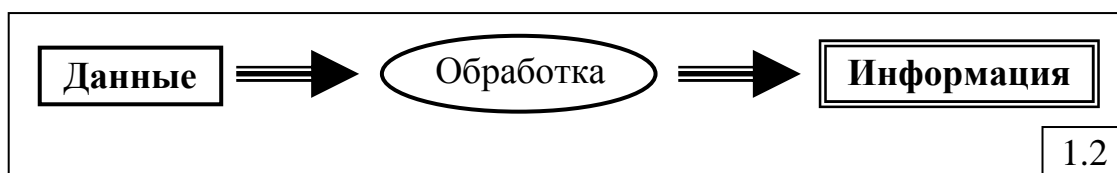
Например, телефонная книга содержит *данные* в виде множества телефонных номеров различных организаций. Извлекая же номер некоторой конкретной организации, в которую мы хотим позвонить, мы получаем *информацию* в виде телефонного номера (или нескольких телефонных номеров) этой организации. По этой же причине мы говорим «база данных» (а не «база информации»), но, формируя запрос к базе данных, мы получаем информацию в виде сведений, представляющих для нас определённый интерес.

«Информация» – понятие субъективное. Сведения, которые являются информацией для одного человека, могут не быть информацией для другого. Например, сведения типа «Париж – столица Франции, а Лондон – столица Англии» являются информацией для школьника, который впервые узнал об этом, и не являются информацией (чем-то новым и ранее не известным) для взрослого человека.

Следует также иметь в виду, что количественной мерой данных является **объём** – количество единиц данных, измеренных в байтах, словах, страницах, количестве телефонных номеров в телефонной книге и т.п. В то же время, количественной мерой информации является **энтропия** – мера неопределенности информации. Чем больше энтропия, тем более ценной является информация.

Таким образом, можно сказать, что в компьютерной сети передаются и данные, и информация.

Взаимосвязь понятий «данные» и «информация» в рассматриваемом контексте иллюстрируется рис.1.2, показывающим, что информация извлекается из множества данных в результате некоторых манипуляций (обработки данных).



### 1.1.3. Средства вычислительной техники

*Средства вычислительной техники* (СВТ) реализуют обработку данных и представляют собой совокупность *ЭВМ, вычислительных комплексов* и *вычислительных систем* различных классов.

Определим смысловое значение каждого из упомянутых терминов – «ЭВМ», «вычислительный комплекс», «вычислительная система» – и покажем существующую между ними разницу.

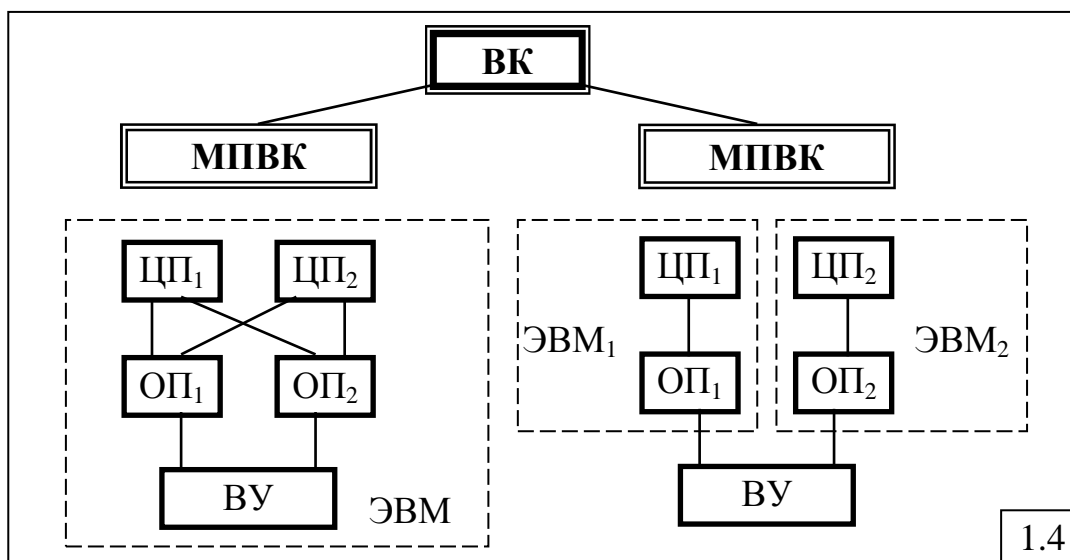
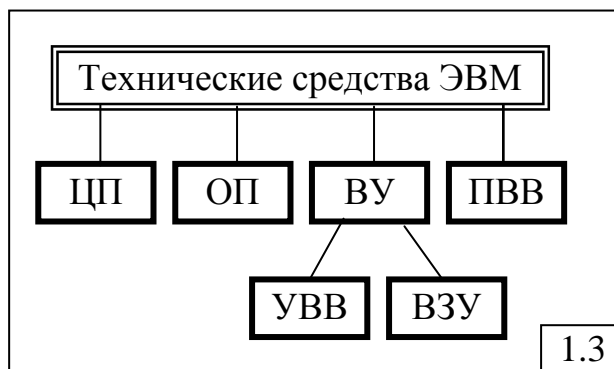
*ЭВМ (электронная вычислительная машина, компьютер)* – совокупность *технических средств*, предназначенных для организации ввода, хранения, автоматической обработки по заданной программе и вывода данных (информации).

К техническим средствам относятся (рис.1.3):

- центральный процессор (ЦП);

- оперативная (основная) память (ОП);
- внешние устройства (ВУ), включающие устройства ввода-вывода (УВВ) и внешние запоминающие устройства (ВЗУ);
- процессоры (каналы) ввода-вывода (ПВВ, КВВ).

**Вычислительный комплекс (ВК)** – совокупность технических средств, содержащая несколько центральных процессоров и представляющая собой одну ЭВМ с несколькими ЦП (**МПВК** – многопроцессорный ВК) или объединение нескольких однопроцессорных ЭВМ (**ММВК** – многомашинный ВК) (рис.1.4).



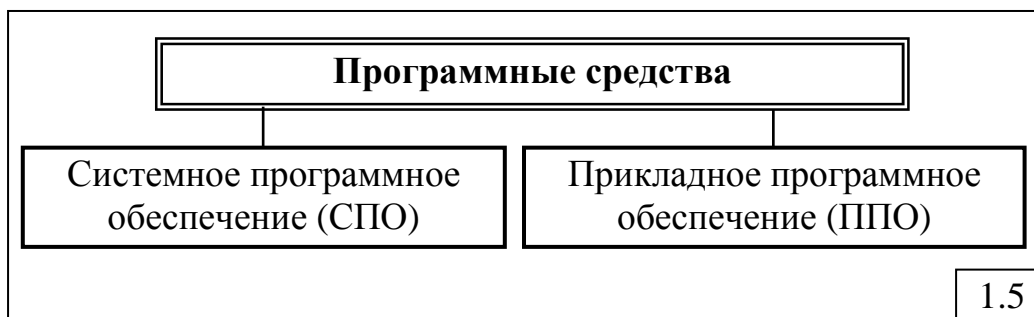
Основной целью построения ВК является обеспечение высокой надежности и/или производительности, не достижимой для однопроцессорных ЭВМ.

**Вычислительная система (ВС)** – совокупность технических и программных средств, ориентированных на решение определенной совокупности задач.

К программным средствам относятся (рис.1.5):

- **системное программное обеспечение**, представляющее собой совокупность стандартных программных средств, обеспечивающих функционирование ВС и включающих операционную систему (ОС), основными составляющими которой для организации эффективного функционирования ВС, являются управляющие программы (УП), а также трансляторы с алгоритмических языков и библиотеки математических и служебных программ;

- **прикладное программное обеспечение** в виде множества прикладных программ (ПП), обеспечивающих ориентацию ВС на решение задач конкретной области применения.



Понятие «вычислительная система» в рассматриваемом контексте полностью согласуется с понятием «система», сформулированным в общей теории систем, в соответствии с которым *система должна обладать структурной и функциональной организацией*, а также фундаментальными свойствами: *целостностью, связностью, организованностью и интегративностью*. Последнее означает, что система обладает свойствами (функциями), которые не присущи ни одному из элементов, входящих в состав системы.

Именно программные средства обеспечивают функциональную организацию ВС, реализуемую управляющими программами операционной системы и прикладными программами. Свойство интегративности в значительной степени обеспечивается прикладными программами. Действительно, элементы (устройства) ЭВМ обеспечивают функции обработки данных (ЦП), хранения данных (ОП, ВЗУ), ввода и вывода данных (УВВ). В то же время вычислительная система с соответствующим программным обеспечением может выполнять функции перевода с одного языка на другой, играть в шахматы и другие игры, воспроизводить звук, фото- и видеоизображения и т.д., то есть ВС обладает функциями, не присущими отдельным устройствам ЭВМ.

Таким образом, многопроцессорный (многомашинный) ВК, рассматриваемый в совокупности с программным обеспечением, можно называть *многопроцессорной (многомашинной) вычислительной системой* – МПВС (ММВС), а суперЭВМ с программным обеспечением – *высокопроизводительной ВС* (ВПВС).

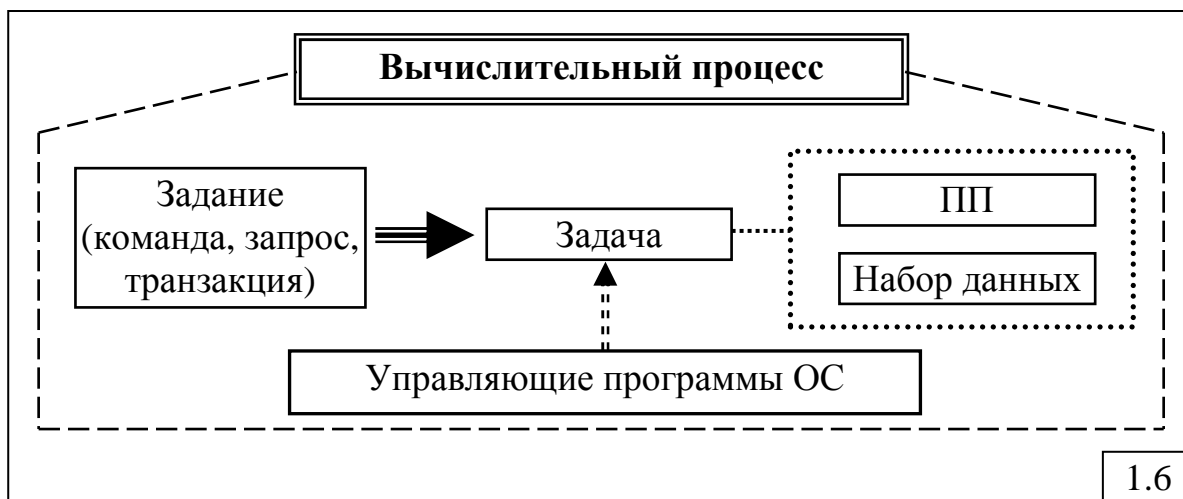
Ещё одной отличительной особенностью ЭВМ от ВС является единица измерения производительности. Производительность ЭВМ измеряется в MIPS (миллион инструкций, команд или операций в секунду) или в FLOPS (операций с плавающей точкой в секунду – для суперЭВМ), а производительность ВС – в количестве задач, выполняемых системой за единицу времени, называемой *системной производительностью*. Очевидно, что системная производительность зависит как от параметров технических средств ВС, так и от параметров программных средств, в частности, прикладных программ. Ясно, что количество «коротких» задач, выполняемых системой за единицу времени в ВС, всегда будет больше, чем «длинных» задач.

На системном уровне в качестве основной единицы работы ВС рассматривается **задача**, представляющая совокупность определенной прикладной программы с определенным набором данных (рис.1.6).

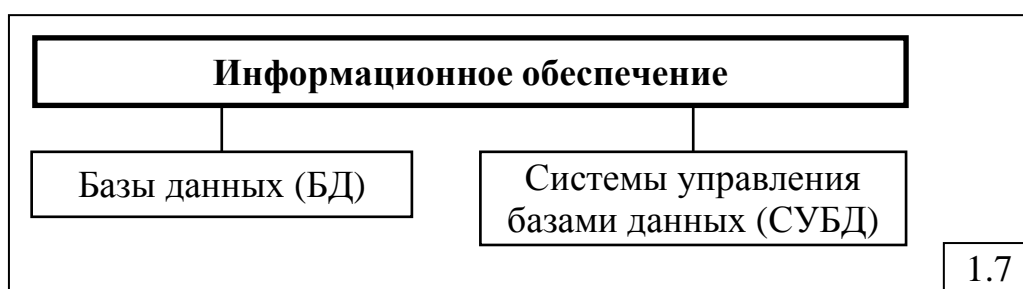
Причиной инициализации задачи может быть **задание (команда, запрос, транзакция)**.

Выполнение задач в ВС называется **вычислительным процессом**.

Определенный порядок (последовательность) прохождения задач через систему, то есть управление вычислительным процессом, осуществляется **управляющими программами ОС**.



К программным средствам ВС тесно примыкают **базы данных и системы управления базами данных**, которые можно рассматривать как самостоятельную составляющую ВС – **информационное обеспечение** (рис.1.7).



**База данных (БД)** – упорядоченные наборы данных (файлы), имеющие определенную структуру.

**Системы управления базами данных (СУБД)** – специальные программные средства, предназначенные для формирования, модификации и выборки данных.

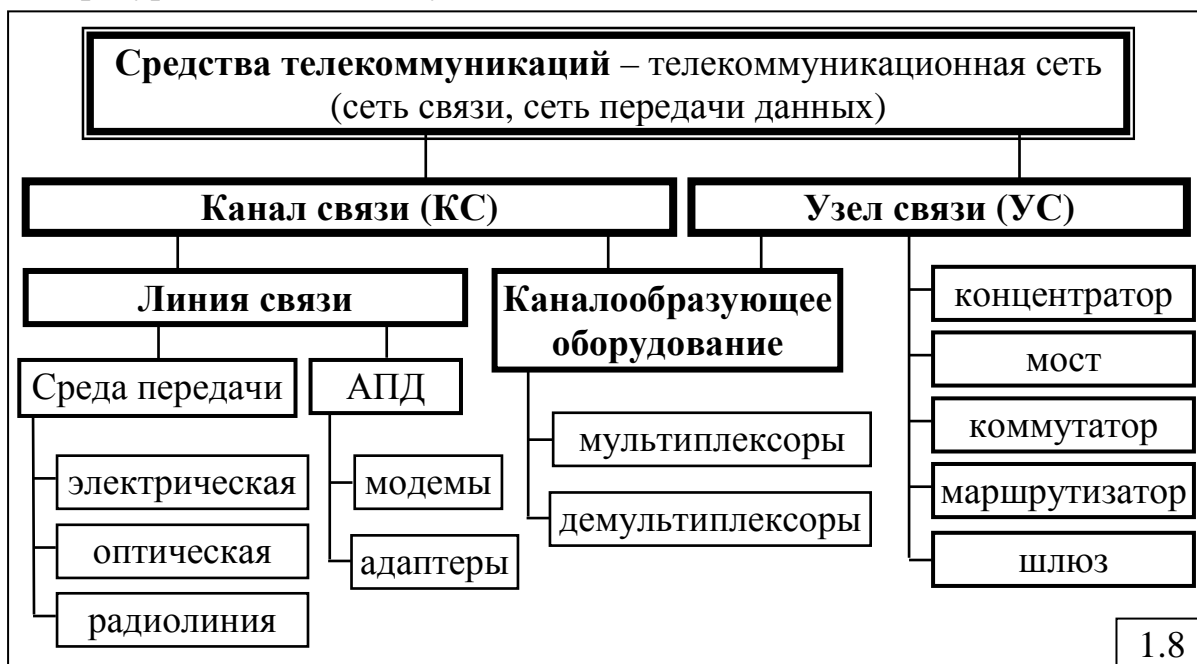
Часто термин "вычислительная система" используется в качестве обобщенного понятия. При этом предполагается, что ВС может быть построена на базе однопроцессорной ЭВМ, многомашинного или многопроцессорного вычислительного комплекса, а компьютерная сеть, представляющая собой объединение нескольких ВС, может рассматриваться как система более высокого уровня.

Компьютерная сеть кроме функций ввода, хранения, обработки и вывода данных реализует функции по передаче данных на значительные расстояния между абонентами сети, в качестве которых выступают ВС и пользователи сети, имеющие доступ к ресурсам сети с помощью удаленных терминалов.

### 1.1.4. Средства телекоммуникаций

**Средства телекоммуникаций** (СТК) реализуют передачу данных и образуют *телекоммуникационную сеть* (*сеть связи, сеть передачи данных*), состоящую из *узлов связи* (УС), объединенных *каналами связи* (КС) для передачи данных (рис.1.8).

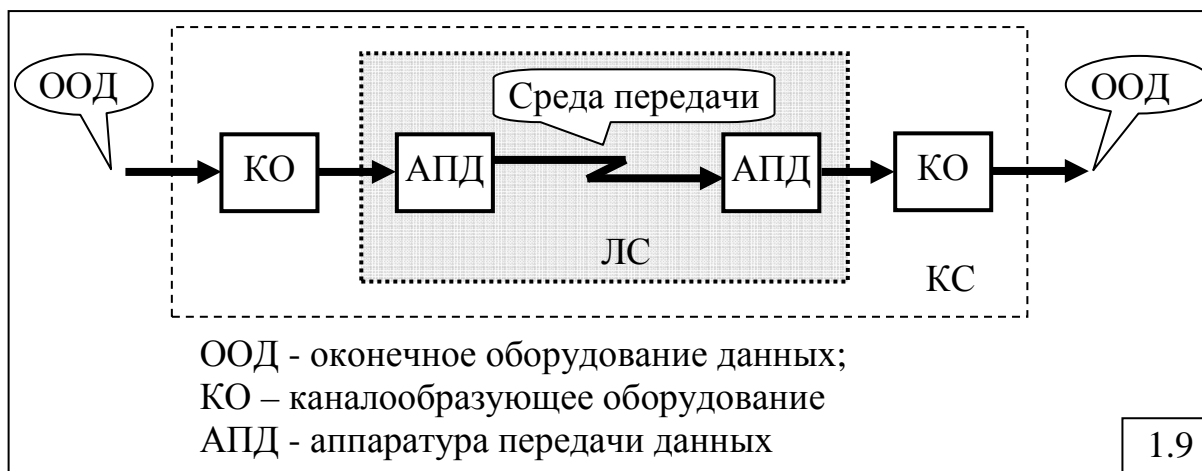
Способ объединения УС и КС определяет **топологию** (конфигурацию) телекоммуникационной сети.



1.8

**Канал связи** (КС) включает в себя *линию связи* (ЛС) и *каналообразующее оборудование*.

**Линия связи** (ЛС) представляет собой физическую среду передачи, по которой передаются сигналы, вместе с *аппаратурой передачи данных* (АПД), формирующей сигналы, соответствующие типу ЛС (рис.1.9).



ООД - окончное оборудование данных;  
 КО – каналообразующее оборудование  
 АПД - аппаратура передачи данных

1.9

**Аппаратура передачи данных (АПД)** осуществляет преобразование сигналов в соответствии с типом среды передачи (линии связи). К АПД относятся различного типа *модемы* (модуляторы-демодуляторы), используемые в телефонных и высокочастотных КС: телефонные, кабельные, радиомодемы, xDSL-модемы, адаптеры и т.д.

**Каналообразующее оборудование (КО)** предназначено для формирования канала передачи данных между двумя взаимодействующими абонентами, при этом в одной и той же линии связи одновременно может быть сформировано несколько каналов за счет использования различных методов уплотнения.

Технология уплотнения и формирования многоканальных систем передачи данных в компьютерных сетях называется **мультиплексированием** и реализуется мультиплексорами и демультиплексорами. Обычно каналообразующее оборудование входит в состав узлов телекоммуникационной сети.

Основными функциями *узлов связи* являются:

- *маршрутизация*, заключающаяся в выборе направления передачи (маршрута) данных;
- *коммутация*, заключающаяся в установлении физического или логического соединения между входными и выходными портами узла;
- *мультиплексирование*, заключающееся в объединении нескольких входящих в узел потоков данных в один выходящий из узла поток;
- *демультиплексирование*, заключающееся в разделении одного входящего в узел потока данных на несколько выходящих из узла потоков.

В качестве узлов связи в вычислительных сетях используются специализированные сетевые устройства: концентраторы, мосты, коммутаторы, маршрутизаторы и шлюзы.

В качестве оконечного оборудования данных (ООД) (рис.1.9) могут выступать компьютеры и сетевое оборудование (мосты, коммутаторы, маршрутизаторы), находящееся в узлах сети.

Состав ЭВМ, вычислительного комплекса, системы и сети, а также взаимосвязь между рассмотренными понятиями иллюстрируется рис.1.10.

### 1.1.5. Понятия архитектуры и технологии компьютерной сети

В широком смысле под **архитектурой компьютерной сети** будем понимать множество технических и инженерных решений по структурной и функциональной организации сети, обеспечивающих определенную совокупность ее свойств и характеристик, рассматриваемую с точки зрения пользователя сети и отличающую данную конкретную сеть от любой другой сети.

Под **технологией компьютерной сети** (сетевой технологией) будем понимать совокупность способов организации (реализации) передачи и обработки данных, обеспечивающих достижение определенных целей, формулируемых в виде требований к качеству (эффективности) обработки и передачи данных.



Вычислительная сеть											
Вычислительная система (ВС)											
Вычислительный комплекс (ВК)											
Электронная вычислительная машина (ЭВМ)											
Технические средства (ТС)					Программные средства (ПС)			Информационное обеспечение		Средства связи	
ВУ (ВЗУ, УВВ)	ОП	ПВВ (КВВ)	ЦП	...ЦП	ОС	ПП	СУБД	БД	КС	УС	
									ЛС	КО	

1.10

## 1.2. Состав и типы компьютерных сетей

### 1.2.1. Состав компьютерной сети

Концептуально компьютерную сеть можно представить как совокупность взаимосвязанных узлов (рис.1.11).

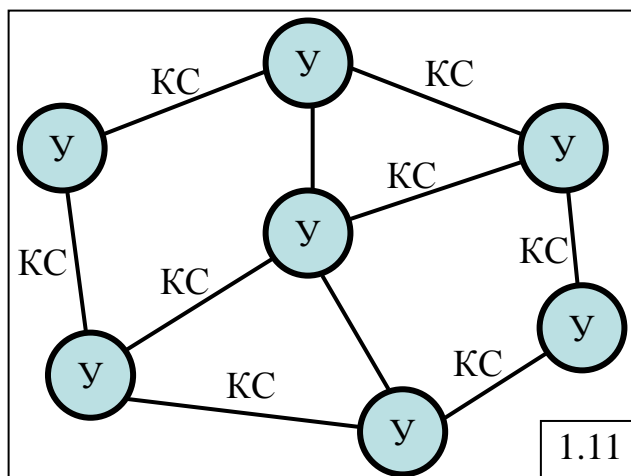
**Узел сети** – совокупность средств, объединенных каналами связи и реализующих функции:

- выбора направления и организации передачи данных (средства коммутации и маршрутизации); такие узлы называются *узлами связи (узлами коммутации, узлами передачи данных)*;
- обработки данных (средства обработки данных); такие узлы называются *узлами (центрами) обработки данных*.

В качестве узлов связи могут использоваться коммутаторы и маршрутизаторы, а в качестве узлов обработки данных – компьютеры, предоставляющие свои информационные и вычислительные ресурсы пользователям сети и называемые *хост-машинами* или просто *хостами* (host).

В сети Internet термин "хост-машина" трактуется более широко: *хост-машиной* называется любой пользовательский компьютер, подключенный к сети.

Совокупность средств коммуникаций (связи) для передачи данных, состоящая из каналов связи и узлов связи, образует *сеть связи*, называемую также *телекоммуникационной сетью* или *сетью передачи данных (СПД)*. Следует иметь в виду, что понятие «сеть передачи данных»

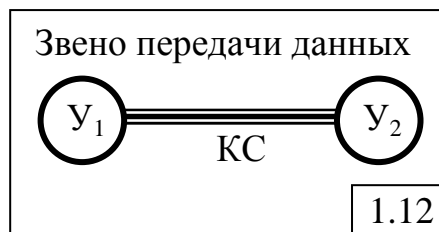


имеет более узкий смысл по сравнению с понятием «сеть связи». СПД предназначена для передачи компьютерных (цифровых) данных, в то время как в сети связи (или, что то же самое, телекоммуникационной сети) могут передаваться как цифровые (дискретные), так и непрерывные (аналоговые) данные, к которым относятся мультимедийные данные – речь, аудио и видео.

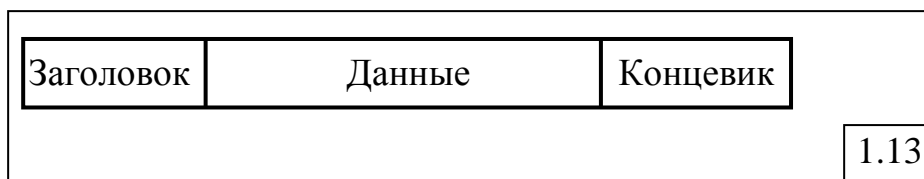
Два узла, связанные каналом связи, образуют *звено передачи данных* (рис.1.12).

Данные в компьютерной сети передаются в виде сообщений.

**Сообщение** представляет собой единицу данных, передаваемую *между*



*пользователями* сети как *единое целое* и имеющую определённый смысл. В качестве сообщений могут выступать программные файлы, электронные письма, неподвижные изображения, видеофильмы и т.п. Сообщение представляется в определённом формате (рис.1.13), содержащем в общем случае заголовок и концевик. В заголовке указывается адрес получателя данного сообщения и адрес отправителя, а также дополнительная служебная информация (тип и длина сообщения, приоритет и т.д.), необходимая для эффективной передачи сообщения в сети. Концевик обычно содержит контрольную сумму, используемую для обнаружения ошибок, которые могут появиться при передаче сообщения по сети.



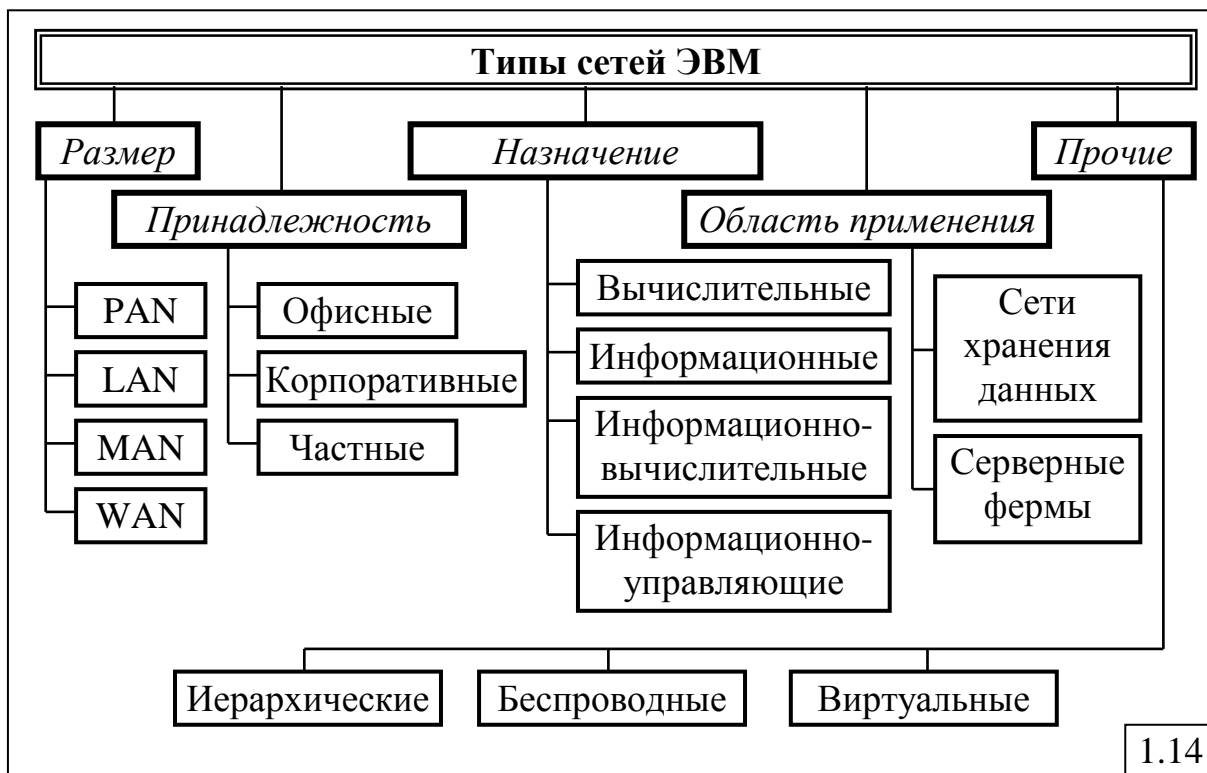
Сообщение при передаче через телекоммуникационную сеть может быть разбито на несколько блоков данных, каждый из которых представляется в формате, аналогичном сообщению (рис.1.13). Такой блок данных отличается от сообщения тем, что имеет ограниченную длину, в то время как длина сообщения в принципе не ограничена. Так, например, сообщение, представляющее собой видеофильм длиной в несколько гигабайт, при передаче через телекоммуникационную сеть может быть разбито на множество блоков данных, длина которых не будет превышать 1000 байт, причём каждый такой блок данных будет иметь заголовок с одинаковыми адресами отправителя и получателя.

### 1.2.2. Классификация сетей ЭВМ

Классификация сетей ЭВМ (компьютерных сетей), как любых больших и сложных систем, может быть выполнена на основе различных признаков, в качестве которых могут быть использованы (рис.1.14):

- размер (территориальный охват) сети;
- принадлежность;
- назначение;

- область применения.



1. По размеру (территориальному охвату) сети ЭВМ делятся на:

- персональные;
- локальные;
- городские (региональные).
- глобальные.

**Персональная сеть** (*Personal Area Network, PAN*) — это сеть, объединяющая персональные электронные устройства пользователя (телефоны, карманные персональные компьютеры, смартфоны, ноутбуки и т.п.) и характеризующаяся:

- небольшим числом абонентов;
- малым радиусом действия (до нескольких десятков метров);
- не критичностью к отказам.

К стандартам таких сетей в настоящее время относятся Bluetooth, Zigbee, Пиконет.

**Локальная вычислительная сеть (ЛВС)** (*Local Area Network, LAN*) – сеть со скоростью передачи данных, как правило, не менее 1 Мбит/с, обеспечивающая связь на небольших расстояниях – от нескольких десятков метров до нескольких километров. Оборудование, подключаемое к ЛВС, может находиться в одном или нескольких соседних зданиях.

Примеры ЛВС: Ethernet, Token Ring.

**Городская вычислительная сеть** (*Metropolitan Area Network, MAN*) – сеть, промежуточная по размеру между ЛВС и глобальной сетью. Протоколы и кабельная система для городской вычислительной сети описываются в стандартах комитета IEEE 802.6. MAN реализуется на основе протокола DQDB (*Distributed Queue Dual Bus*) – двойная шина с

распределенной очередью и использует волоконно-оптический кабель для передачи данных со скоростью 100 Мбит/с на территории до 100 км<sup>2</sup>. MAN может применяться для объединения в одну сеть группы сетей, расположенных в разных зданиях. Последние разработки, связанные с высокоскоростным беспроводным доступом в соответствии со стандартом IEEE 802.16, привели к созданию MAN в виде широкополосных беспроводных ЛВС.

**Глобальная сеть** (*Wide Area Network, WAN*) – в отличие от ЛВС охватывает большую территорию и представляет собой объединение нескольких ЛВС, связанных с помощью специального сетевого оборудования (маршрутизаторов, коммутаторов и шлюзов), образующих в случае использования высокоскоростных каналов магистральную сеть передачи данных (магистральную сеть связи). Наиболее широкое применение находят глобальные сети для нужд информационного обмена в коммерческих, научных и других профессиональных целях.

Для построения глобальных сетей могут использоваться различные сетевые технологии, в том числе TCP/IP, X.25, Frame Relay, ATM, MPLS.

Настоящей глобальной сетью, пожалуй, можно считать только сеть Интернет. Вряд ли глобальной можно считать сеть, объединяющую 2-3 ЛВС, находящиеся в разных городах, расположенных на расстоянии нескольких десятков или даже сотен километров друг от друга. Однако, поскольку для построения такой «простой» сети используются обычно те же сетевые технологии и технические средства, что и в сети Интернет, то такие сети обычно тоже относят к классу глобальных сетей.

## 2. По принадлежности сети ЭВМ делятся на:

- *офисные* – сети, расположенные на территории офиса компании, ограниченной обычно пределами одного здания, и построенные на технологиях LAN;
- *корпоративные (ведомственные)* – сети, представляющие собой объединение нескольких офисных сетей компании, расположенных в разных территориально разнесенных зданиях, находящихся возможно в разных городах и регионах, и построенные на технологиях MAN или WAN;
- *частные* – сети, построенные обычно на технологии **виртуальной частной сети** (*Virtual Private Network, VPN*), позволяющей обеспечить одно или несколько сетевых соединений, которые могут быть трёх видов: узел-узел, узел-сеть и сеть-сеть, образующих логическую сеть поверх другой сети (например, Интернет).

## 3. По назначению сети ЭВМ делятся на:

- *вычислительные*, предназначенные для решения задач пользователей, ориентированных, в основном, на вычисления;
- *информационные*, ориентированные на предоставление информационных услуг; примерами таких сетей могут служить сети, предоставляющие справочные и библиотечные услуги;

- *информационно-вычислительные*, предназначенные для решения задач пользователей и предоставления информационных услуг;
- *информационно-управляющие*, предназначенные для управления реальными объектами и процессами.

4. По области применения сети можно разделить на:

- сети хранения данных;
- серверные фермы.

**Сеть хранения данных (СХД) (Storage Area Network, SAN)** представляет собой множество внешних устройств хранения данных, таких как дисковые массивы, ленточные библиотеки, оптические накопители, подключённые к серверам, при этом операционная система рассматривает подключённые ресурсы, как локальные.

Следует не путать сеть хранения данных с **сетевой системой хранения данных (Network Attached Storage, NAS)**, представляющей собой компьютер с дисковым массивом, подключенный обычно к локальной сети и поддерживающий работу по принятым в этой сети протоколам. Часто диски в NAS объединены в RAID массив. Несколько таких компьютеров могут быть объединены в одну систему, обеспечивая надёжность хранения данных, простой доступ для пользователей и хорошую масштабируемость.

**Серверная ферма** – это множество серверов, соединенных сетью передачи данных и работающих как единое целое. Серверная ферма обычно является ядром крупного центра обработки данных (ЦОД), обеспечивающего распределенную обработку данных.

К перечисленным типам сетей следует добавить:

- беспроводные ЛВС;
- виртуальные локальные вычислительные сети;
- иерархические сети;

**Беспроводная ЛВС (wireless LAN – WLAN)** – локальная сеть, использующая для передачи данных инфракрасное излучение или чаще всего радиоволны.

**Виртуальная локальная вычислительная сеть (ВЛВС) (virtual LAN – VLAN)** – логическое объединение узлов локальной сети, позволяющее выделить пользователей одной рабочей группы с общими интересами в отдельный сетевой сегмент. При этом объединяемые узлы могут принадлежать различным физическим сегментам.

**Иерархическая сеть (hierarchical network)** – сеть, в которой главным вычислительным центром является одна хост-машина, а терминалами – остальные сетевые устройства. Это традиционная архитектура, противоположная современной архитектуре распределенных вычислений, в которых интеллектуальные рабочие станции играют более активную роль в вычислительном процессе.

### 1.2.3. Администрирование компьютерных сетей

Важным требованием к любой компьютерной сети, обеспечивающим эффективное функционирование, является её **управляемость**, заключающаяся в возможности:

- централизованного наблюдения и контроля состояния основных элементов сети, отдельных подсистем и сети в целом;
- выявления и устранения возникающих в процессе функционирования сети проблем, таких как сбои и отказы отдельных устройств сети, определение и устранение перегрузок и т.д.;
- сбора и анализа данных для оценки производительности сети и планирования развития сети;
- обеспечения информационной безопасности и защиты данных и т.п.

Для реализации перечисленных возможностей необходимо в сети иметь специальные автоматизированные средства администрирования, взаимодействующие с техническими и программными средствами сети с помощью коммуникационных протоколов.

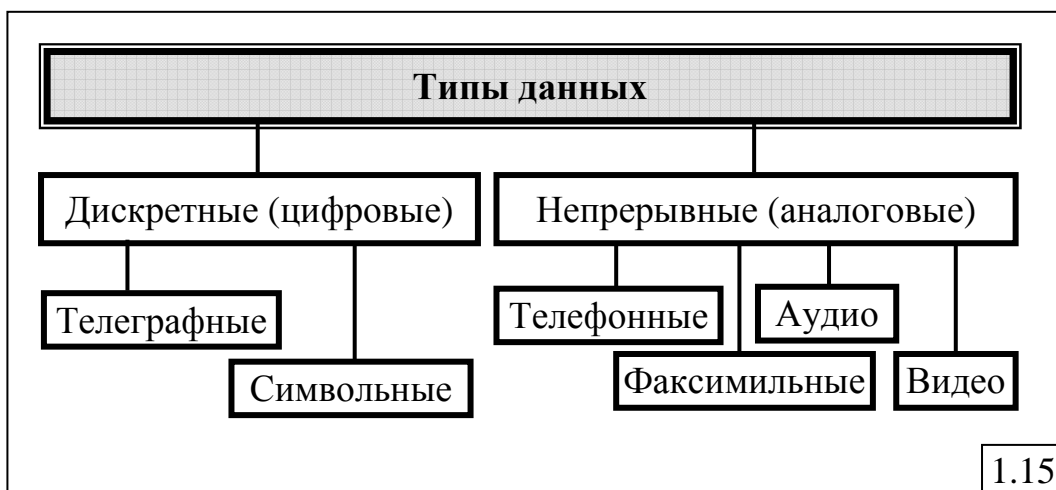
Поддержка и обеспечение эффективного функционирования компьютерной сети за счет принятия своевременных организационных решений по управлению сетью на основе анализа характеристик функционирования и текущего состояния сети реализуется в рамках **администрирования** компьютерной сети сетевым администратором.

К основным функциям администрирования сети относятся:

- наблюдение за потоками данных;
- установка новых версий программного обеспечения;
- создание и поддержание таблиц маршрутизации и коммутации;
- диагностика состояния компонентов сети;
- контроль ошибок и устранение простых отказов;
- замена отказавших узлов резервными;
- реконфигурация сети;
- поддержка отказоустойчивости компьютерной сети;
- добавление новых пользователей;
- определение прав пользователей сети при их обращении к разным ресурсам: файлам, каталогам, принтерам и т.д.;
- ограничение возможностей пользователей в выполнении тех или иных системных действий.

### 1.2.4. Типы данных

Первоначально сети ЭВМ строились для обработки и передачи компьютерных данных, представляемых в цифровой (дискретной) форме. Современные компьютерные сети ориентированы на передачу и обработку самых разнообразных данных, которые могут быть разделены на следующие типы (рис.1.15).



1. **Телеграфные данные** – дискретные данные, представляемые в виде импульсов постоянного или переменного тока, передаваемые по телеграфным каналам связи (ТгКС).

2. **Телефонные (голосовые) данные** – речь в спектре частот от 80 до 12000 Гц, передаваемая по телефонным КС (ТфКС), называемым также *каналами тональной частоты (ТЧ)*. Речь по таким каналам передаётся в ограниченной полосе частот от 300 Гц до 3400 Гц, что обеспечивает разборчивость фраз более 99%.

3. **Факсимильные данные** – неподвижные изображения.

4. **Аудиоданные** (звуковое вещание) – в отличие от телефонных, кроме речи передается музыка, пение и т.п. в спектре частот от 20 Гц до 20 кГц. Для качественной передачи аудио данных достаточна полоса частот от 30 Гц до 15 кГц.

5. **Видеоданные** (телевизионное вещание) – совокупность движущихся изображений и звукового сопровождения в спектре частот от 40 Гц до 6 МГц. В современных компьютерных сетях различают видеоданные трёх типов, отличающиеся требованиями к качеству передачи:

- **видеоконференцсвязь**, представляющая собой медленно изменяющиеся изображения и характеризующаяся *невысокими требованиями к качеству передачи*;

- **телевизионное вещание обычного качества**;

- **телевизионное вещание высокой чёткости**.

6. **Символьные (цифровые, компьютерные) данные** – совокупность символов, например двоичных символов в компьютерах.

Телеграфные и цифровые данные по своей природе относятся к *дискретным* данным, остальные – к *непрерывным* данным, но которые могут быть представлены (закодированы) в цифровой форме.

Телефонные, аудио- и видеоданные относятся к так называемым **мультимедийным данным**, к которым предъявляются специфические требования к качеству передачи по сравнению с обычными компьютерными (цифровыми) данными.

### 1.3. Многоуровневая организация вычислительных сетей

#### 1.3.1. Требования к организации компьютерных сетей

Для обеспечения эффективного функционирования компьютерных сетей предъявляются требования, основными среди которых являются (рис.1.16):

1) **открытость** – возможность добавления в сеть новых компонентов (узлов и каналов связи, средств обработки данных) без изменения существующих технических и программных средств;

2) **гибкость** – сохранение работоспособности при изменении структуры сети в результате сбоев и отказов отдельных компонентов сети или при замене оборудования;

3) **совместимость** – возможность работы в сети оборудования разного типа и разных производителей;

4) **масштабируемость** – способность сети увеличивать свою производительность при добавлении ресурсов (узлов и каналов связи);

5) **эффективность** – обеспечение требуемого качества обслуживания пользователей, задаваемого в виде показателей производительности, временных задержек, надежности и т.д., при минимальных затратах.



Указанные требования реализуются за счет *многоуровневой организации управления процессами* в сети, в основе которой лежат понятия **процесса, уровня, интерфейса и протокола** (рис.1.17).





### 1.3.2. Понятия процесса и уровня

Функционирование вычислительных систем и сетей удобно описывать в терминах процессов.

**Процесс** – динамический объект, реализующий целенаправленный акт обработки или передачи данных.

Процессы делятся на:

1) **прикладные** – обработка данных в ЭВМ и терминальном оборудовании, а также передача данных в СПД;

2) **системные** – обеспечение прикладных процессов (активизация терминала для прикладного процесса, организация связи между процессами и др.).

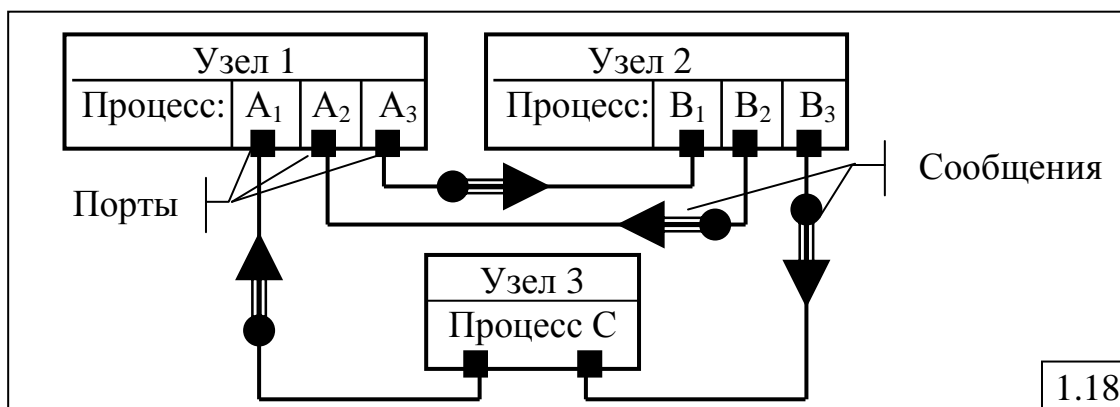
Данные между процессами передаются в виде сообщений через логические программно-организованные точки, называемые **портами**.

Порты разделяются на *входные* и *выходные*.

Промежуток времени, в течение которого взаимодействуют процессы, называется **сеансом** или **сессией**.

В каждом узле обработки данных (компьютере) могут одновременно выполняться несколько независимых прикладных процессов, связанных, например, с обработкой данных (такие процессы называются вычислительными процессами). Эти процессы путём обмена сообщениями через соответствующие порты могут взаимодействовать с прикладными процессами, протекающими в других узлах вычислительной сети так, как это показано на рис.1.18.

Здесь в узле 1 и 2 выполняются по 3 прикладных процесса  $A_1, A_2, A_3$  и  $B_1, B_2, B_3$  соответственно, а в узле 3 выполняется один прикладной процесс  $C$ . Эти процессы через соответствующие порты обмениваются сообщениями, причем процесс  $C$  обменивается сообщениями через два порта: входной, через который поступают сообщения от процесса  $B_3$ , и выходной, который служит для передачи сообщений от процесса  $C$  к процессу  $A_1$ .



Одним из основных понятий многоуровневой организации управления процессами в компьютерных сетях является понятие уровня, которое лежит в основе моделей всех сетевых технологий.

**Уровень (layer)** – понятие, позволяющее разделить всю совокупность функций обработки и передачи данных в вычислительной сети на несколько иерархических групп. На каждом уровне реализуются определенные функции обработки и передачи данных с помощью аппаратных и/или программных средств сети. Каждый уровень обслуживает вышележащий уровень и, в свою очередь, пользуется услугами нижележащего.

### 1.3.3. Модель взаимодействия открытых систем (OSI-модель)

Международная Организация по Стандартам (МОС, International Standards Organization – ISO) предложила в качестве стандарта открытых систем семиуровневую коммуникационную модель (рис.1.19), известную как **OSI-модель** (Open Systems Interconnection) – модель Взаимодействия Открытых Систем (ВОС).



Каждый уровень OSI-модели отвечает за отдельные специфические функции в коммуникациях и реализуется техническими и программными средствами вычислительной сети.

#### 1.3.3.1. Физический уровень

**Уровень 1 – физический** (*physical layer*) – самый низкий уровень OSI-модели, определяющий процесс прохождения сигналов через среду передачи между сетевыми устройствами (узлами сети).

*Реализует* управление каналом связи:

- подключение и отключение канала связи;
- формирование передаваемых сигналов и т.п.

*Описывает:*

- механические, электрические и функциональные характеристики среды передачи;
- средства для установления, поддержания и разъединения физического соединения.

*Обеспечивает* при необходимости:

- кодирование данных;
- модуляцию сигнала, передаваемого по среде.

Данные физического уровня представляют собой поток битов (последовательность нулей или единиц), закодированные в виде электрических, оптических или радио сигналов.

Из-за наличия помех, воздействующих на электрическую линию связи, **достоверность передачи**, измеряемая как вероятность искажения одного бита, составляет  $10^{-4} - 10^{-6}$ . Это означает, что в среднем на 10000 – 1000000 бит передаваемых данных один бит оказывается искажённым.

### 1.3.3.2. Канальный уровень

**Канальный уровень** или **уровень передачи данных** (*data link layer*) является вторым уровнем OSI-модели.

*Реализует управление:*

- доступом сетевых устройств к среде передачи, когда два или более устройств могут использовать одну и ту же среду передачи;
- надежной передачей данных в канале связи, позволяющей увеличить достоверность передачи данных на 2-4 порядка.

*Описывает* методы доступа сетевых устройств к среде передачи, основанные, например, на передаче маркера или на соперничестве.

*Обеспечивает:*

- функциональные и процедурные средства для установления, поддержания и разрыва соединения;
- управление потоком для предотвращения переполнения приемного устройства, если его скорость меньше, чем скорость передающего устройства;
- надежную передачу данных через физический канал с вероятностью искажения данных  $10^{-8} - 10^{-9}$  за счёт применения методов и средства контроля передаваемых данных и повторной передачи данных при обнаружении ошибки.

Таким образом, канальный уровень обеспечивает достаточно надежную передачу данных через ненадежный физический канал.

Блок данных, передаваемый на канальном уровне, называется **кадром** (*frame*).

На канальном уровне появляется свойство *адресуемости* передаваемых данных в виде *физических (машинных) адресов*, называемых также **MAC-адресами** и являющихся обычно уникальными идентификаторами сетевых устройств.

Как будет показано в разделе 3, универсальные MAC-адреса в ЛВС Ethernet и Token Ring являются 6-байтными и записываются в шестнадцатеричном виде, причём байты адреса разделены дефисом, например: **00-19-45-A2-B4-DE** .

К процедурам канального уровня относятся:

- добавление в кадры соответствующих адресов;
- контроль ошибок;
- повторная, при необходимости, передача кадров.

На канальном уровне работают ЛВС Ethernet, Token Ring и FDDI.

### 1.3.3.3. Сетевой уровень

**Сетевой уровень** (network layer), в отличие от двух предыдущих, отвечает за передачу данных в СПД и управляет маршрутизацией сообщений – передачей через несколько каналов связи по одной или нескольким сетям, что обычно требует включения в пакет *сетевого адреса* получателя.

Блок данных, передаваемый на сетевом уровне, называется **пакетом** (*packet*).

**Сетевой адрес** – это специфический идентификатор для каждой промежуточной сети между источником и приемником информации.

Сетевой уровень реализует:

- обработку ошибок,
- мультиплексирование пакетов;
- управление потоками данных.

Самые известные протоколы этого уровня:

- X.25 в сетях с коммутацией пакетов;
- IP в сетях TCP/IP;
- IPX/SPX в сетях NetWare.

Кроме того, к сетевому уровню относятся протоколы построения маршрутных таблиц для маршрутизаторов: OSPF, RIP, ES-IS, IS-IS.

### 1.3.3.4. Транспортный уровень

**Транспортный уровень** (transport layer) наиболее интересен из высших уровней для администраторов и разработчиков сетей, так как он управляет сквозной передачей сообщений между оконечными узлами сети ("end-end"), обеспечивая надежность и экономическую эффективность передачи данных независимо от пользователя. При этом оконечные узлы возможно взаимодействуют через несколько узлов или даже через несколько транзитных сетей.

На транспортном уровне реализуется:

- 1) *преобразование длинных сообщений в пакеты* при их передаче в сети и обратное преобразование;
- 2) *контроль последовательности прохождения пакетов*;
- 3) *регулирование трафика в сети*;
- 4) *распознавание дублированных пакетов* и их уничтожение.

Способ коммуникации "end-end" облегчается еще одним способом адресации – **адресом процесса**, который соотносится с определенной прикладной программой (прикладным процессом), выполняемой на компьютере. Компьютер обычно выполняет одновременно несколько программ, в связи с чем необходимо знать какой прикладной программе (процессу) предназначено поступившее сообщение. Для этого на транспортном уровне используется специальный адрес, называемый **адресом порта**. Сетевой уровень доставляет каждый пакет на конкретный адрес компьютера, а транспортный уровень передает полностью собранное сообщение конкретному прикладному процессу на этом компьютере.

Транспортный уровень может предоставлять различные типы сервисов, в частности, передачу данных без установления соединения или с предварительным установлением соединения. В последнем случае перед началом передачи данных с использованием специальных управляющих пакетов устанавливается соединение с транспортным уровнем компьютера, которому предназначены передаваемые данные. После того как все данные переданы, подключение заканчивается. При передаче данных без установления соединения транспортный уровень используется для передачи одиночных пакетов, называемых *дейтаграммами*, не гарантируя их надежную доставку. Передача данных с установлением соединения применяется для надежной доставки данных.

#### **1.3.3.5. Сеансовый уровень**

**Сеансовый уровень** (session layer) обеспечивает обслуживание двух "связанных" на уровне представления данных объектов сети и управляет ведением диалога между ними путем синхронизации, заключающейся в установке служебных меток внутри длинных сообщений. Эти метки позволяют после обнаружения ошибки повторить передачу данных не с самого начала, а только с того места, где находится ближайшая предыдущая метка по отношению к месту возникновения ошибки.

Сеансовый уровень предоставляет услуги по организации и синхронизации обмена данными между процессами уровня представлений.

На сеансовом уровне реализуется:

- 1) *установление соединения с адресатом и управление сеансом;*
- 2) *координация связи прикладных программ на двух рабочих станциях.*

#### **1.3.3.6. Уровень представления**

**Уровень представления** (presentation layer) обеспечивает совокупность служебных операций, которые можно выбрать на прикладном уровне для интерпретации передаваемых и получаемых данных. Эти служебные операции включают в себя:

- *управление информационным обменом;*
- *преобразование (перекодировка) данных* во внутренний формат каждой конкретной ЭВМ и обратно;
- *шифрование и дешифрование данных* с целью защиты от несанкционированного доступа;
- *сжатие данных*, позволяющее уменьшить объём передаваемых данных, что особенно актуально при передаче мультимедийных данных, таких как аудио и видео.

Служебные операции этого уровня представляют собой основу всей семиуровневой модели и *позволяют связывать воедино терминалы и средства вычислительной техники (компьютеры) самых разных типов и производителей.*

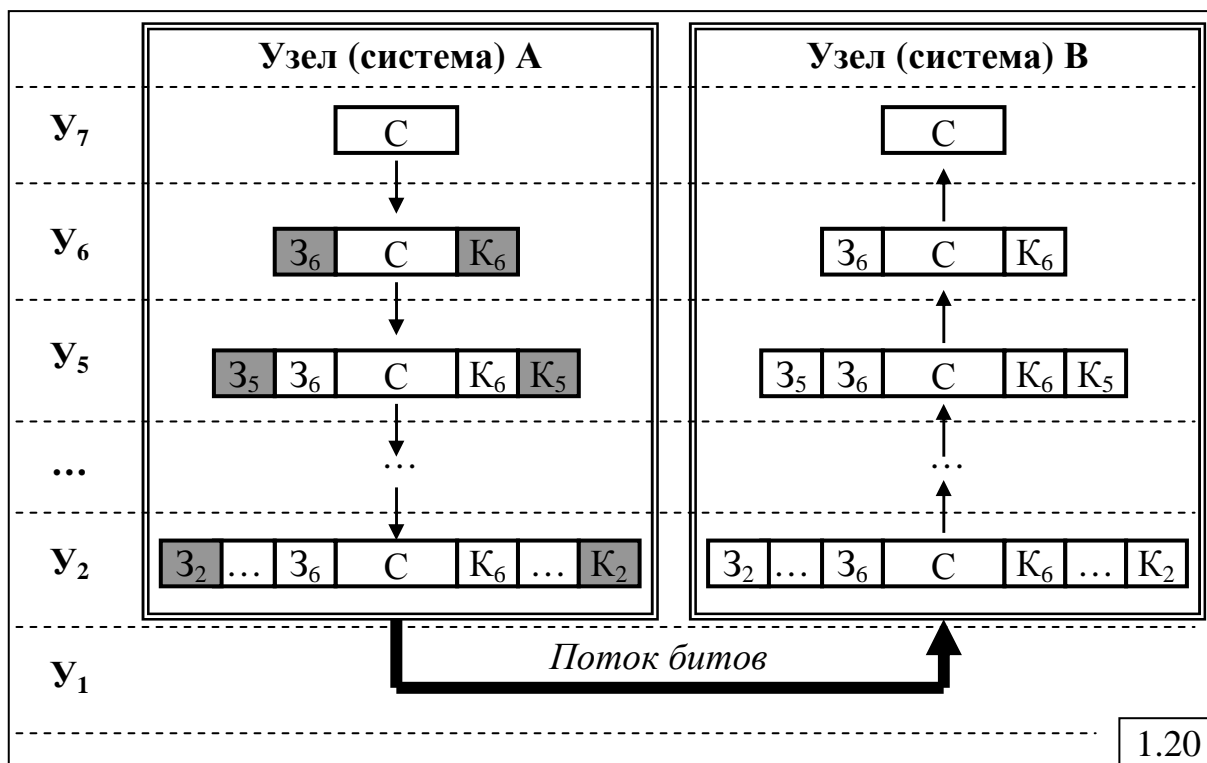
### 1.3.3.7. Прикладной уровень

**Прикладной уровень** (application layer) обеспечивает непосредственную поддержку прикладных процессов и программ конечного пользователя, а также управление взаимодействием этих программ с различными объектами сети. Другими словами, прикладной уровень обеспечивает интерфейс между прикладным ПО и системой связи. Он предоставляет прикладной программе доступ к различным сетевым службам, включая передачу файлов и электронную почту.

### 1.3.3.8. Процесс передачи сообщений в OSI-модели

Транспортный, сеансовый, представительский и прикладной уровни (уровни 4 – 7) относятся к **высшим уровням OSI-модели**. В отличие от **нижних уровней** (1 – 3) они отвечают за коммуникации типа "end-end", т.е. коммуникации между источником и приемником сообщения.

В соответствии с OSI-моделью сообщения в передающем узле **А** (компьютере) проходят вниз через все уровни от верхнего  $Y_7$  до самого нижнего  $Y_1$  (рис.1.20), причем многоуровневая организация управления процессами в сети порождает необходимость модифицировать на каждом уровне передаваемые сообщения применительно к функциям, реализуемым на этом уровне. Модификация заключается в добавлении к сообщению на каждом уровне соответствующих заголовков  $Z_i$  и концевиков  $K_i$ , называемых **обрамлением сообщения**, в которых содержится информация об адресах взаимодействующих объектов, а также информация, необходимая для обработки сообщения на данном уровне.



Когда сообщение достигает низшего (физического) уровня  $Y_1$ , оно пересылается к другому узлу **В** в виде потока битов, представляющего собой физические сигналы (электрические, оптические или радиоволны)

передающей среды. В приемном узле (компьютере) сообщение от нижнего физического уровня  $У_1$  проходит вверх через все уровни, где от него отсекаются соответствующие заголовки и концевики. Таким образом, каждый уровень оперирует с собственным заголовком и концевиком, за счет чего обеспечивается независимость данных, относящихся к разным уровням управления передачей сообщений.

### 1.3.4. IEEE-модель локальных сетей

Институт инженеров по электронике и электротехнике (Institute of Electrical and Electronics Engineers – IEEE) предложил вариант OSI-модели, используемый при разработке и проектировании локальных сетей и получивший название *IEEE-модели*.

В IEEE-модели канальный уровень разбивается на два подуровня (рис.1.21):

- *подуровень управления доступом к среде передачи (Medium Access Control, MAC-подуровень)*, описывающий способ доступа сетевого устройства к среде передачи данных;
- *подуровень управления логическим соединением (Logical Link Control, LLC-подуровень)*, описывающий способ установления и завершения соединения, а также способ передачи данных.

**LLC-подуровень** предоставляет более высоким уровням возможность управлять качеством услуг и обеспечивает сервис трех типов:

- 1) *сервис без установления соединения и без подтверждения доставки;*
- 2) *сервис без установления соединения с подтверждением доставки;*
- 3) *сервис с установлением соединения.*

Сервис без установления соединения и подтверждения доставки не гарантирует доставку данных и обычно применяется в приложениях, использующих для контроля передачи данных и защиты от ошибок протоколы более высоких уровней.

Сервис с установлением соединения обеспечивает надежный обмен данными.

Главной функцией **MAC-уровня** является обеспечение доступа к каналу передачи данных. На этом уровне формируется физический адрес устройства, который называется *MAC-адресом*. Каждое устройство сети идентифицируется этим *уникальным* адресом, который присваивается всем сетевым устройствам.

Уровни OSI-модели	Подуровни IEEE-модели
7 - прикладной	
6 - представления	
5 - сеансовый	
4 - транспортный	
3 - сетевой	
2 - канальный	LLC
	MAC
1 - физический	

1.21

### 1.3.5. Понятия интерфейса и протокола

Описание сетевой технологии и алгоритма функционирования компьютерной сети связано с описанием соответствующих интерфейсов и протоколов.

**Интерфейс** – соглашение о взаимодействии (границе) между уровнями одной системы, определяющее структуру данных и способ (алгоритм) обмена данными *между соседними уровнями OSI-модели*.

Интерфейсы подразделяются на:

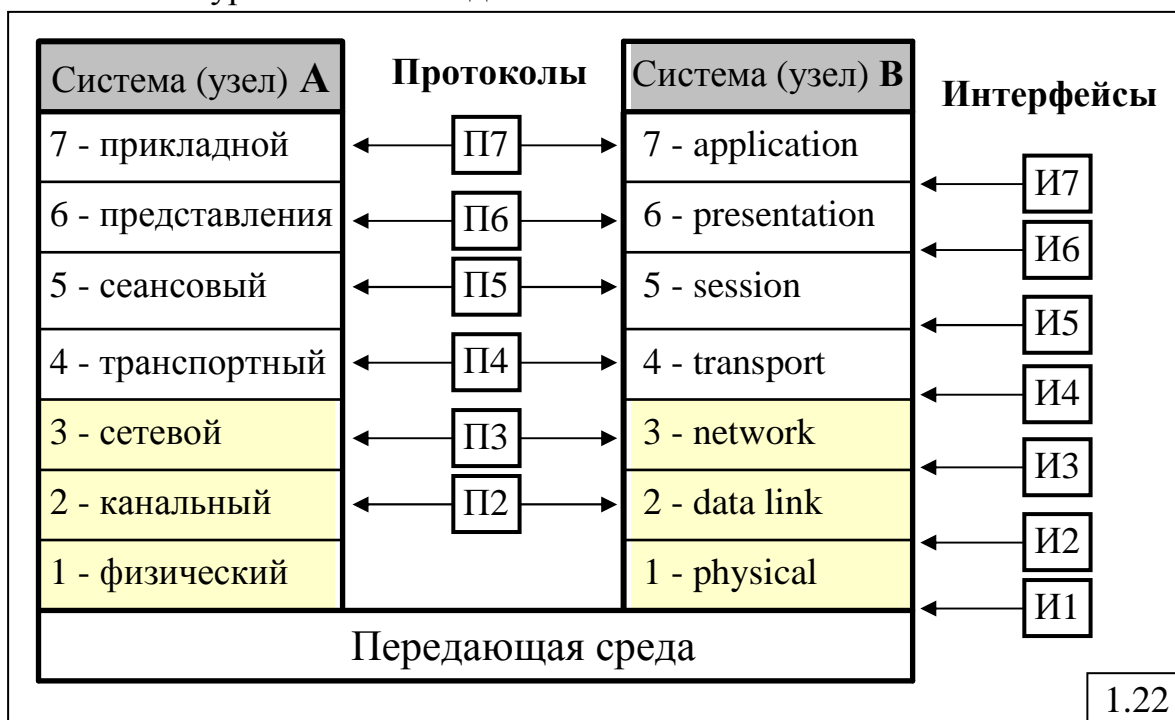
- 1) *схемные* – совокупность интерфейсных шин;
- 2) *программные* – совокупность процедур реализующих порядок взаимодействия между уровнями.

**Протокол** – совокупность правил, регламентирующих формат и процедуры взаимодействия процессов *одноименных уровней* на основе обмена сообщениями.

Описание протокола предполагает задание:

- 1) **логической характеристики протокола**, определяющей *структуру* (формат) и *содержание* (семантику) *сообщений* путём перечисления типов сообщений и их смысла;
- 2) **процедурной характеристики протокола**, представляющей собой *правила выполнения действий*, предписанных протоколом взаимодействия и задаваемых в форме: операторных схем алгоритмов, автоматных моделей, сетей Петри и др.

Рис.1.22 иллюстрирует понятия интерфейсов и протоколов и их соответствие уровням OSI-модели.



Как сказано выше, каждая сетевая технология характеризуется совокупностью протоколов и интерфейсов разных уровней OSI-модели. Совокупность протоколов всех уровней некоторой сетевой технологии



называется **стеком протоколов**. В настоящее время существует большое количество разнообразных сетевых технологий и соответствующих им стеков протоколов, наиболее известными и распространёнными среди которых являются стеки протоколов: TCP/IP, XNS, IPX, AppleTalk, DECnet, SNA. Краткое описание перечисленных стеков протоколов приводится в конце данного раздела (см. п.1.7).

### 1.3.6. Протокольные блоки данных (PDU)

Данные, передаваемые на разных уровнях в сети, формируются в виде блоков, называемых **протокольными блоками данных** (Protocol Data Unit – PDU). PDU представляет собой единицу данных, передаваемую как единое целое и имеющую оформление в виде заголовка со служебной информацией (адрес отправителя, адрес получателя, длина блока и т.п.) и, возможно, концевика.

На разных уровнях OSI-модели используются разные PDU, имеющие специальные названия. Наибольшее распространение получили следующие названия блоков данных: *сообщение*, *дейтаграмма*, *пакет*, *кадр* (рис. 1.23).

Уровни OSI-модели		PDU	
7	Прикладной	<i>Сообщение</i>	Message
...	...	...	...
4	Транспортный	<i>Дейтаграмма</i>	Datagram
3	Сетевой	<i>Пакет</i>	Packet
2	Канальный	<i>Кадр</i>	Frame

1.23

**Сообщение (message)** – блок данных, рассматриваемых как единое целое при передаче между двумя пользователями (процессами) и имеющих определенное смысловое значение. Сообщения используются на 7-м уровне OSI-модели для передачи данных между прикладными процессами и могут иметь произвольную длину.

**Кадр (frame)** – блок данных 2-го (канального) уровня OSI-модели, имеющий ограниченную длину и передаваемый как единое целое в локальной сети или по выделенному каналу связи между двумя узлами.

**Пакет (packet)** – блок данных на 3-го (сетевого) уровня OSI-модели, имеющий ограниченную длину и представляющий собой единицу передачи данных в СПД.

**Дейтаграмма (datagram)** – блок данных 4-го (транспортного) уровня OSI-модели, передаваемый дейтаграммным способом без установления соединения.

Предельный размер кадра, пакета и дейтаграммы зависит от сетевой технологии и устанавливается соответствующими протоколами, определяющими формат и допустимый размер блока данных.

Кроме перечисленных названий в стеке протоколов TCP/IP блок данных протокола TCP называется **сегментом**, который получается путём вырезания из неструктурированного **потока** байтов, поступающих к протоколу TCP в рамках логического соединения от протоколов более высокого уровня.

Для блоков данных 5-го и 6-го уровней OSI-модели нет устоявшихся общепринятых названий, что в значительной степени обусловлено отсутствием этих уровней в наиболее распространённом стеке протоколов TCP/IP.

Отметим, что в ATM-сетях данные передаются в виде блоков фиксированного размера в 53 байта, которые называются **ячейками (cell)**.

### 1.3.7. Сетевая операционная система

Основной задачей сетевой операционной системы (ОС) является организация процессов обработки и передачи данных в компьютерной сети, связанная, в том числе, с разделением ресурсов сети (например, дискового пространства) и администрированием сети (определение разделяемых ресурсов, паролей и прав доступа для каждого пользователя или группы пользователей).

Для решения этих задач сетевая операционная система, в отличие от операционной системы ЭВМ, должна обладать встроенными возможностями для работы в сети за счёт **дополнительных функций**, таких как:

- поддержка функционирования сетевого оборудования – маршрутизаторов, коммутаторов, шлюзов и т.п.;
- поддержка сетевых протоколов, включая протоколы маршрутизации и протоколы авторизации;
- реализация доступа к среде передачи данных и к удалённым ресурсам сети и т.д.

Совокупность операционных систем отдельных ЭВМ, входящих в состав вычислительной сети можно рассматривать как составную часть сетевой операционной системы. При этом разные ЭВМ могут работать под управлением как одинаковых, так и разных ОС (Windows XP, Windows Vista, UNIX, NetWare, Solaris и т.д.). Последнее характерно для современных вычислительных сетей, объединяющих обычно множество компьютеров разных типов различных производителей. Все эти ОС обеспечивают управление вычислительным процессом и распределением ресурсов в каждой из конкретной ВС, выполняя следующие функции:

- управление памятью, включая распределение и защиту памяти;
- планирование и управление пользовательскими и системными процессами;
- управление файлами и внешними устройствами;
- защита данных и администрирование, включая поддержку отказоустойчивости аппаратных и программных средств;

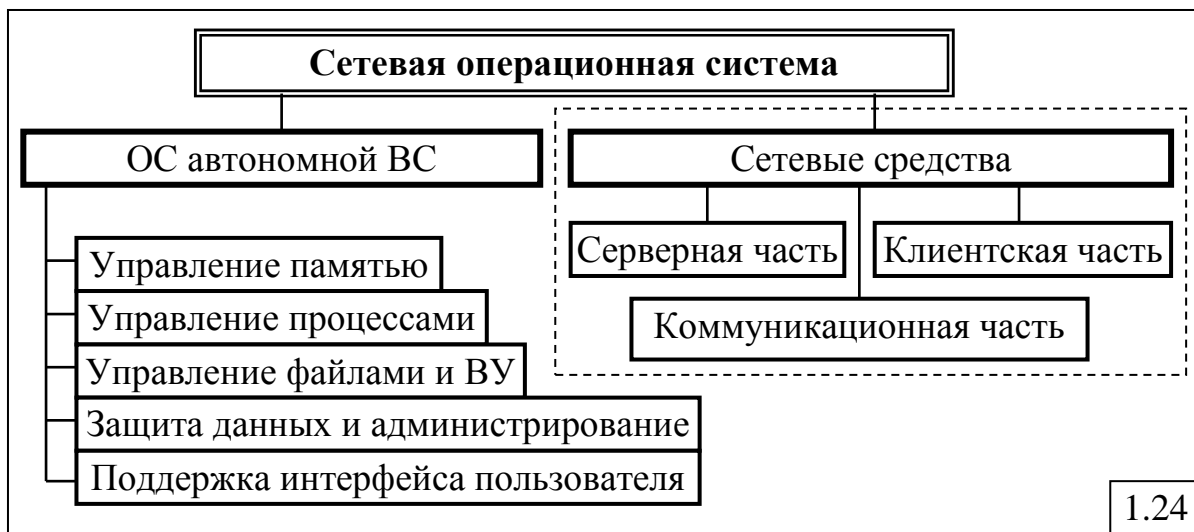
- обеспечение удобного интерфейса для прикладных программ и пользователей и т.д.

Для обеспечения функций по обмену данными между ЭВМ сети операционные системы всех ЭВМ имеют в своём составе дополнительные компоненты – **сетевые средства**, организующие взаимодействие процессов, выполняющихся в разных ЭВМ, и разделение общих ресурсов между пользователями сети. Сетевые средства можно рассматривать как совокупность трёх составляющих:

- **серверная часть ОС**, предназначенная для предоставления локальных ресурсов и услуг в общее пользование;
- **клиентская часть ОС**, обеспечивающая реализацию запросов доступа к удалённым ресурсам и услугам;
- **коммуникационная** (транспортная) часть ОС, обеспечивающая совместно со средствами телекоммуникаций передачу данных в виде сообщений между пользователями вычислительной сети.

Состав сетевой ОС показана на рис.1.24.

Существуют специальные сетевые ОС, которым приданы функции обычных систем (например Windows NT) и обычные ОС, которым приданы сетевые функции (например Windows XP). Сегодня практически все современные ОС имеют встроенные сетевые функции.



Примерами сетевых операционных систем могут служить:

- Microsoft Windows (95, NT и более поздние);
- Novell NetWare;
- различные UNIX системы, такие как Solaris, и т.д.

Реализация обмена данными между удаленными пользователями – одна из основных функций вычислительной сети. Эффективность передачи данных характеризуется совокупностью показателей (характеристик), в частности, временем и надежностью доставки сообщений, и в значительной степени зависит от структурной и функциональной организации вычислительной сети.

## 1.4. Принципы структурной организации компьютерных сетей

Структурная организация компьютерной сети определяется:

- 1) *составом узлов* (номенклатура и количество сетевых устройств, компьютеров и терминалов) и *топологией* сети передачи данных;
- 2) *производительностью узлов* обработки и передачи данных и *пропускной способностью каналов* связи.

Одной из важнейшей составляющей структурной организации компьютерной сети является её топология, оказывающая существенное влияние как на качество передачи, так и на эффективность обработки данных.

Ниже рассматриваются типовые топологии, используемые при построении компьютерных сетей, и проводится их сравнительный анализ.

### 1.4.1. Сетевые топологии

Многообразие типов компьютерных сетей обуславливает многообразие топологий, обеспечивающих выполнение заданных требований к качеству их функционирования. В современных компьютерных сетях наибольшее распространение получили следующие топологии (рис. 1.25):

- а) «Общая шина»;
- б) «Дерево»;
- в) «Звезда (узловая)»;
- г) «Кольцо»;
- д) «Полносвязная»;
- е) «Многосвязная (ячеистая)»;
- ж) «Смешанная».

Следует различать *физическую* и *логическую* топологию сети.

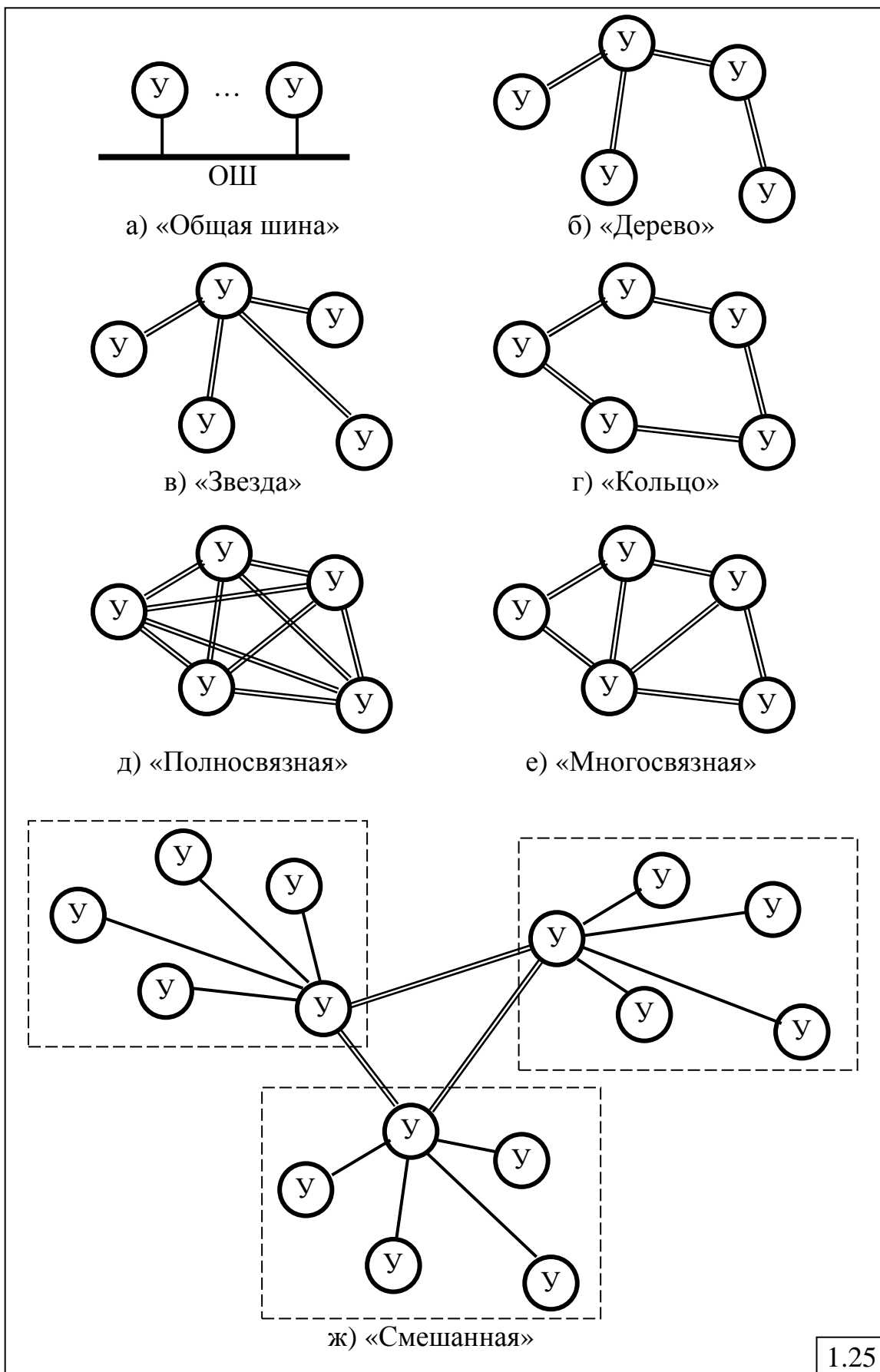
**Физическая (структурная) топология** отображает структурную взаимосвязь узлов сети.

**Логическая (функциональная) топология** определяется функциональной взаимосвязью узлов сети, то есть отображает последовательность передачи данных между узлами сети.

Физическая и логическая топологии сети, как мы увидим ниже, могут различаться.

**Топология «общая шина»** (рис. 1.25,а), представляет собой кабель, называемый *шиной* или *магистралью*, к которому подсоединены компьютеры сети. Данные, передаваемые любым компьютером, занимают шину на всё время передачи, при этом остальные компьютеры, имеющие данные для передачи, должны ждать освобождения общей шины. Таким образом, в каждый момент времени передавать данные может только один компьютер сети, и пропускная способность общей шины некоторым образом распределяется между всеми компьютерами. Основным *достоинством* топологии «общая шина» является простота структурной и функциональной организации и, как следствие, дешевизна, что делает её

наиболее привлекательной для локальных сетей. Недостаток этой топологии заключается в низкой надёжности сети – выход из строя общей шины приводит к полной остановке сети.



**Топология «дерево»** (рис. 1.25,б) формируется по принципу «минимума суммарной длины связей между узлами сети» и является основой для построения иерархических сетей. В таких сетях для передачи данных существует только один путь между двумя любыми узлами, что делает процедуру маршрутизации тривиальной.

**Топология «звезда»** (рис. 1.25,в) содержит один центральный узел, к которому присоединяются все остальные узлы сети. В качестве центрального узла может выступать мощный компьютер, к которому присоединены менее мощные периферийные компьютеры. В этом случае центральный компьютер может предоставлять свои ресурсы (файлы, дисковое пространство, ресурсы процессора) периферийным компьютерам, либо выполнять функции маршрутизатора при обмене данными между компьютерами сети. Возможна и другая организация топологии «звезда», когда в качестве центрального узла используется сетевое устройство (например, концентратор или коммутатор), с помощью которого все компьютеры связаны в единую сеть и которое обеспечивает только обмен данными между компьютерами. Если в качестве центрального узла сети используется концентратор, то логическая топология сети может быть как «звезда», так и «общая шина».

В топологии «кольцо» каждый узел связан с двумя другими узлами так, как это показано на рис.1.25,г), при этом данные, переданные каким-либо узлом, пройдя через все другие узлы сети, могут вернуться в исходный узел. Основным *достоинством* этой топологии по сравнению с рассмотренными выше топологиями является возможность передачи данных по двум направлениям, то есть наличие в каждом узле альтернативного пути, по которому могут быть переданы данные при отказе основного пути. При этом стоимость сети при небольшом количестве узлов соизмерима со стоимостью сетей с топологиями «звезда» и «дерево». Однако с увеличением количества узлов в сети стоимость может оказаться значительной.

**Топология «полносвязная»** (рис. 1.25,д) формируется по принципу «каждый с каждым», то есть каждый узел сети имеет связь со всеми другими узлами. Такая топология является наиболее эффективной по всем основным показателям качества функционирования: надёжности, производительности и т.д., но из-за большой стоимости практически не используется.

**Топология «многосвязная»** или «ячеистая» (рис. 1.25,е) представляет собой топологию произвольного вида, которая формируется по принципу «каждый узел сети связан с не менее чем двумя другими узлами», то есть для каждого узла сети всегда должен быть хотя бы один альтернативный путь. Такая топология может быть получена путем удаления из полносвязной топологии некоторых каналов связи (например, не используемых для передачи данных или мало загруженных), что во многих случаях существенно снижает стоимость сети.

**Топология «смешанная»** представляет собой любую комбинацию рассмотренных выше топологий и образуется обычно при объединении нескольких локальных сетей, например так, как это показано на рис.1.25,ж), где 3 сети с топологией «звезда» связаны в сеть с топологией «кольцо».

### 1.4.2. Сравнительный анализ топологий

Сравнительный анализ топологий компьютерных сетей будем проводить на основе следующих признаков:

1) *простота структурной организации*, измеряемая количеством каналов связи между узлами сети;

2) *надёжность*, определяемая наличием «узких мест», при отказе которых сеть перестаёт функционировать или же резко падает её эффективность, а также наличием альтернативных путей, благодаря которым, при отказах отдельных каналов и узлов, передача данных может осуществляться в обход отказавших элементов;

3) *производительность сети*, измеряемая количеством блоков данных (сообщений или пакетов), передаваемых в сети за единицу времени с учётом возможного снижения эффективной скорости передачи данных из-за конфликтов в сети;

4) *время доставки* сообщений (пакетов), измеряемое, например, в *хонах (hop)*, представляющих собой число промежуточных каналов или узлов на пути передачи данных;

5) *стоимость* топологии, зависящая как от состава и количества оборудования (например, каналов при заданном количестве узлов), так и от сложности реализации.

Перечисленные признаки взаимосвязаны. Естественно, что более эффективные топологии с позиций надёжности, производительности и времени доставки являются более сложными в реализации и, как следствие, более дорогими. Сравнение рассмотренных выше топологий будем проводить на качественном уровне, результаты которого представлены в виде табл.1.1. В таблице наилучшему показателю соответствует значение 1, заключённое в фигурные скобки, а наихудшему показателю – значение 5.

Таблица 1.1

Показатель	Топология						
	ОШ	Звезда	Дерево	Кольцо	Полно-связная	Много-связная	Смешанная
Простота	{1}	2	2	3	5	4	4
Стоимость	{1}	2	2	3	5	4	4
Надёжность	5	4	4	3	{1}	2	2
Производит.	5	4	4	3	{1}	2	2
Время дост.	3	2	4	5	{1}	3	3

**Простота структурной организации и стоимость.** По количеству каналов связи *наиболее простой* топологией компьютерной сети является топология «общая шина», которая содержит один канал связи, объединяющий все компьютеры сети. Простота такой сети обусловлена также отсутствием каких-либо специальных сетевых устройств, таких как маршрутизаторы, коммутаторы и т.п. Единственным необходимым устройством для подключения к общей шине служит сравнительно простое устройство – сетевой адаптер (сетевая карта). Еще одним фактором, обуславливающим простоту этой топологии, является простота подключения новых компьютеров к общей шине. Естественно, что простота структурной организации топологии «общая шина» определяет и её низкую стоимость.

К сравнительно простым и дешёвым топологиям можно отнести топологии «дерево» и «звезда», что обусловлено небольшим количеством связей (каналов)  $N_K$  между узлами сети, которое на единицу меньше количества узлов  $N_Y$ :  $N_K = N_Y - 1$ .

Топология «кольцо» по показателю «простота» занимает следующую позицию после рассмотренных топологий. Легко убедиться, что для этой топологии количество связей (каналов) между узлами сети равно количеству узлов:  $N_K = N_Y$ .

Полносвязная топология является наиболее сложной, поскольку имеет максимально возможное количество связей (каналов) в сети, равное  $N_K = \frac{N_Y(N_Y - 1)}{2}$ . Следствием этого является высокая стоимость сети, что делает нецелесообразным применение такой топологии при построении компьютерных сетей, особенно с большим числом узлов.

При построении глобальных сетей наибольшее распространение получили топологии многосвязные (ячеистые) и смешанные, занимающие промежуточное положение между простыми и дешёвыми топологиями «звезда» и «кольцо» и полносвязной топологией.

**Надёжность.** По показателю надёжности *наилучшей*, естественно, является *полносвязная* топология, которая характеризуется отсутствием «узких мест» с точки зрения надёжности и наличием максимально возможного количества альтернативных путей для передачи данных, которые могут быть задействованы при отказах одного или даже нескольких каналов и узлов сети. При этом сеть продолжает функционировать и передавать данные, правда, с более низким качеством.

Наименее надёжными топологиями являются топологии «общая шина», «звезда» и «дерево», имеющие «узкие места» соответственно в виде общей шины, центрального и корневого узла сети, при отказе которых сеть перестает функционировать.

Несколько выше надёжность топологии «кольцо» за счёт наличия альтернативного пути, обратного по отношению к основному пути



передачи данных, что позволяет при отказах канала или узла сети передавать сообщения в противоположном направлении.

Многосвязные (ячеистые) и смешанные топологии за счёт наличия, в общем случае, нескольких альтернативных путей для передачи данных, обладают более высокой надёжностью, чем топология «кольцо», приближаясь по этому показателю к полносвязной топологии.

**Производительность сети.** Под производительностью сети передачи данных будем понимать количество пакетов, передаваемых в сети за единицу времени. Очевидно, что производительность сети зависит от количества пакетов, одновременно находящихся в сети: чем больше пакетов в сети, тем выше её производительность. Производительность сети растёт до некоторого предельного значения, называемого *пропускной способностью сети передачи данных* (СПД). Значение пропускной способности СПД определяется узким местом сети – наиболее загруженным узлом или каналом связи, загрузка которого близка к единице. Ясно, что пропускная способность СПД в значительной степени определяется пропускными способностями каналов связи, измеряемыми количеством бит, передаваемых по каналу за единицу времени, и количеством каналов связи в СПД, по которым одновременно могут передаваться пакеты, причем, чем больше каналов в СПД, тем выше производительность и пропускная способность сети. Таким образом, при условии, что все каналы связи сравниваемых топологий имеют одинаковые пропускные способности, можно сделать следующий вывод: наибольшей производительностью обладает полносвязная топология, а наименьшей – «общая шина», имеющая только один канал для передачи данных всех компьютеров. Следует также иметь в виду, что в общей шине могут возникать коллизии в результате столкновения данных, одновременно передаваемых от нескольких компьютеров, что ещё больше снижает пропускную способность общей шины. Остальные топологии занимают промежуточное положение между полносвязной топологией и топологией «общая шина».

**Время доставки.** Как и ранее, положим, что все каналы связи сравниваемых топологий имеют одинаковые пропускные способности. В этом случае время доставки пакетов в сети удобно оценивать в *хопах* (hop) – количестве каналов на пути передачи пакетов между узлами сети. Очевидно, что наименьшее время доставки пакетов, равное одному хопу между любыми двумя узлами сети, обеспечивает полносвязная топология. В топологии «звезда» время доставки пакетов не более двух хопов – двух каналов связи между любыми двумя периферийными узлами, путь между которыми пролегает через центральный узел. В многосвязных и смешанных топологиях время доставки несколько больше, чем в топологии «звезда», и зависит от степени связности – количества каналов связи и, соответственно, количества альтернативных путей. Время доставки пакетов в сети с топологией «дерево» зависит от конфигурации связей и, при одном и том же количестве узлов  $N_y$ , может принимать

различные максимальные значения: 2 – в случае конфигурации, совпадающей с топологией «звезда», и  $(N_y - 1)$  – в случае линейной конфигурации, когда все узлы сети, связанные последовательно друг с другом, образуют цепочку:  $У_1-У_2-У_3-...-У_N$ . При достаточно большом количестве узлов наибольшее время доставки может оказаться у сети с топологией «кольцо». Поскольку в реальных сетях с кольцевой топологией пакеты обычно передаются в одном направлении, среднее время доставки, измеренное в хопах, будет равно  $N_y/2$ , где  $N_y$  – количество узлов и, соответственно, каналов связи в сети.

Несколько сложнее оценить время доставки пакетов для сети с топологией «общая шина». Действительно, поскольку канал один – шина, то время доставки равно одному хопу. Однако следует учитывать, что пропускная способность общей шины делится между всеми компьютерами сети, вследствие чего реальное время доставки, измеренное в секундах, может оказаться во много раз больше, чем в канале полносвязной сети с такой же пропускной способностью. Кроме того, возникающие в общей шине коллизии в результате столкновений пакетов от разных компьютеров и необходимость их повторной передачи ещё больше увеличивают время доставки.

Выполненный качественный анализ различных топологий позволяет сделать следующие **выводы**.

Основным требованием, предъявляемым к локальным вычислительным сетям, объединяющим обычно недорогие персональные компьютеры, является *низкая стоимость сетевого оборудования*, что достигается использованием наиболее простых и, следовательно, дешевых топологий: «общая шина», «звезда» и «кольцо». Глобальные вычислительные сети строятся обычно на основе многосвязной или смешанной топологии.

Представленные результаты сравнительного анализа различных сетевых топологий носят относительный характер, то есть показывают уровень того или иного показателя некоторой топологии относительно других топологий, и не могут служить количественной оценкой. Более того, при оценке этих показателей не учитывались значения количественных параметров структурной организации компьютерной сети, таких как пропускные способности и надёжность каналов связи, производительность узлов связи, стоимость компонент сети, эксплуатационные расходы и т.п. Учёт этих параметров в каждом конкретном случае может привести к ситуации, когда более простые топологии оказываются более производительными, а сложные топологии – более дешёвыми, чем простые топологии, например, потому, что в них используются каналы связи с небольшой пропускной способностью.

Существенное влияние на рассмотренные выше характеристики оказывает также функциональная организация компьютерной сети.

## 1.5. Принципы функциональной организации компьютерных сетей

Функциональная организация компьютерной сети складывается из функциональной организации вычислительного процесса (обработки данных) и процесса передачи данных.

Функциональная организация вычислительного процесса определяется режимами функционирования отдельных компьютеров сети и способом реализации обработки данных в компьютерной сети.

Обработка данных в компьютерных сетях может быть реализована двумя способами:

- *распределённая обработка*, при которой обработка данных распределяется между несколькими узлами (компьютерами) и выполняется параллельно;

- *централизованная обработка*, при которой данные обрабатываются в одном центральном узле (компьютере), в качестве которого обычно выступает сервер, при этом другие компьютеры рассматриваются как клиенты (удалённые терминалы), формирующие запросы к центральному узлу.

Функциональная организация процесса передачи данных в значительной степени определяется:

- *способом организации взаимодействия* между абонентами сети – *способом коммутации*;

- *методами управления трафиком* (потоками данных), реализуемыми на разных уровнях OSI-модели.

### 1.5.1. Коммутация

Передача данных в компьютерной сети предполагает организацию физического или логического соединения между взаимодействующими пользователями сети (конечными узлами).

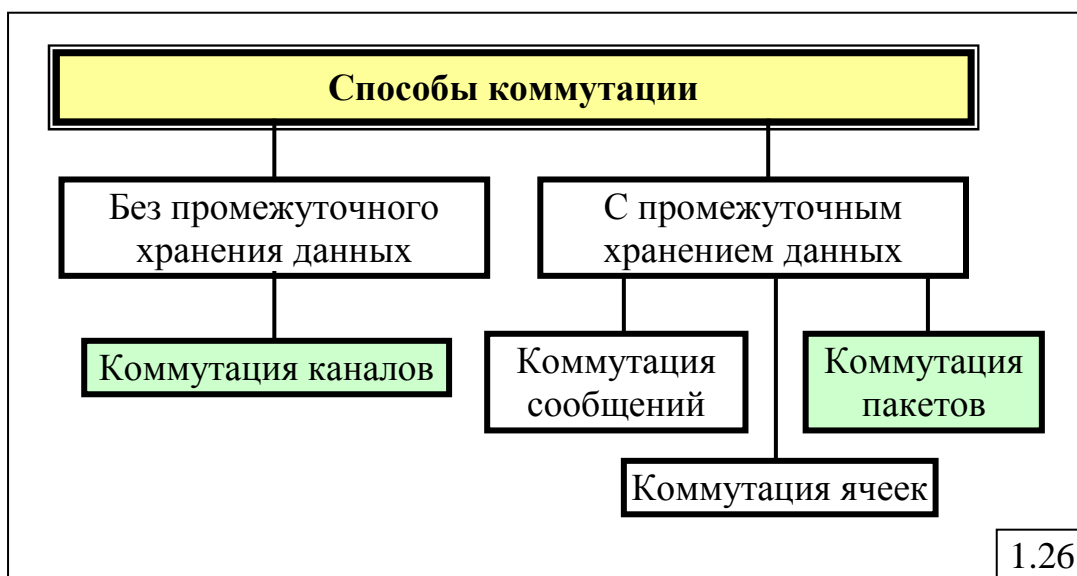
*Организация взаимодействия* между абонентами компьютерной сети называется **коммутацией**. Коммутация в сети может быть реализована разными способами (рис.1.26), которые можно разбить на две группы:

- способы коммутации без промежуточного хранения данных;
- способы коммутации с хранением данных в промежуточных узлах.

В качестве способа коммутации без промежуточного хранения данных в компьютерных сетях применяется коммутация каналов, используемая в традиционных *телефонных* сетях связи.

Для передачи данных в *компьютерных* сетях был разработан новый способ коммутации – коммутация сообщений, предполагающая использование в качестве узлов связи специализированных средств вычислительной техники, что позволяло реализовать в промежуточных узлах хранение передаваемых данных, обеспечивающее ряд преимуществ по сравнению с коммутацией каналов. Дальнейшее развитие способов

коммутиций было направлено на усовершенствование коммутации сообщений для обеспечения определенного качества передачи данных.

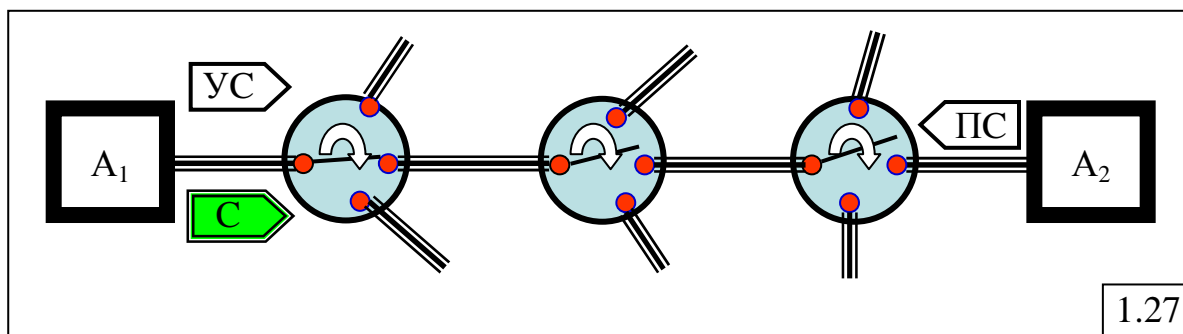


1.26

Рассмотрим перечисленные способы коммутации и в процессе сравнительного анализа выявим присущие им достоинства и недостатки.

#### 1.5.1.1. Коммутация каналов

**Коммутация каналов** основана на формировании единого физического соединения (канала) между взаимодействующими абонентами для непосредственной передачи данных из конца в конец так же, как это реализуется в традиционных телефонных сетях (рис.1.27).



1.27

Если абонент  $A_1$  хочет передать данные абоненту  $A_2$ , то перед началом передачи он предварительно должен установить соединение с абонентом  $A_2$  путем посылки специального служебного сообщения «УС – установить соединение», которое «прокладывает» путь, формируя в каждом из промежуточных узлов непосредственное физическое (электрическое) соединение между входным и выходным портами узла. После того, как служебное сообщение достигнет абонента  $A_2$ , последний формирует и посылает по созданному пути (маршруту) абоненту  $A_1$  новое служебное сообщение «ПС – подтвердить соединение», подтверждающее установление соединения между абонентами сети. Только после получения такого сообщения абонент  $A_1$  может начать передачу сообщения  $C$  абоненту  $A_2$  по установленному маршруту. Созданное физическое

соединение обычно существует в течение времени передачи данных, называемого *сеансом* или *сессией* (*session*), по завершению которого это соединение может быть разрушено. Такой канал между двумя абонентами сети называется *временным* или *коммутируемым*, в отличие от *некоммутируемого* (*выделенного*) канала, который формируется единожды и существует постоянно или, по крайней мере, в течение длительного времени, независимо от того, передаются данные или же канал простаивает.

В простейшем случае узел сети с коммутацией каналов можно рассматривать как переключатель, обеспечивающий в каждый момент времени электрическое соединение между двумя портами (точками входа и выхода) узла. В телефонной сети такими «переключателями» являются автоматические телефонные станции (АТС).

К основным **достоинствам** коммутации каналов относятся:

- *возможность использования* существующих и достаточно хорошо развитых *телефонных сетей связи*;
- *отсутствие необходимости в хранении передаваемых данных* в промежуточных узлах сети;
- *высокая эффективность при передаче больших объемов данных*, поскольку в этом случае относительное значение накладных расходов на установление соединения оказывается незначительным.

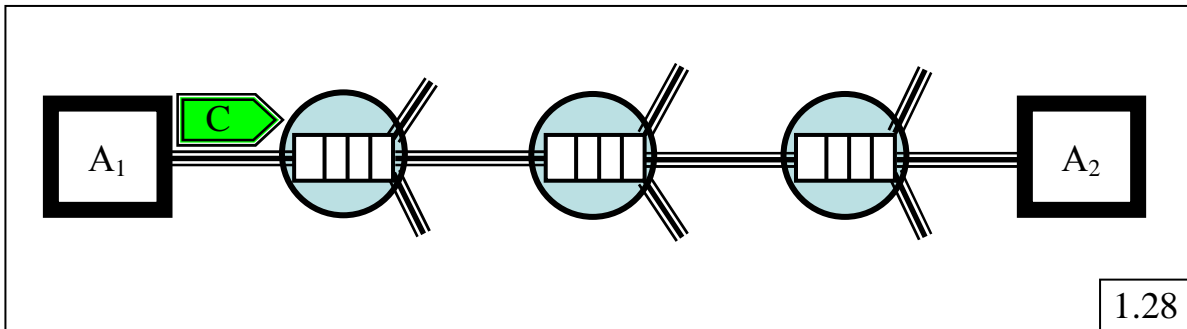
В то же время коммутация каналов обладает следующими серьёзными **недостатками**:

- каналы связи на всем пути передачи данных *должны иметь одинаковые пропускные способности* и обеспечивать одинаковую скорость передачи данных, в противном случае, если пропускная способность некоторого канала связи окажется меньше пропускной способности предыдущего канала, произойдёт потеря передаваемых данных, поскольку в промежуточных узлах отсутствует возможность буферирования (временного хранения) данных;
- *большие накладные расходы на установление соединения* на начальном этапе, что негативно сказывается при передаче небольших объёмов данных, поскольку в этом случае относительное значение накладных расходов на установление соединения оказывается существенным, что приводит к неэффективному использованию ресурсов (пропускной способности) каналов связи, что проявляется в значительном уменьшении реальной скорости передачи данных по отношению к максимально возможной скорости канала, называемой *пропускной способностью*;
- телефонные каналы связи, ориентированные на передачу голоса, имеют сравнительно низкое качество и обеспечивают передачу компьютерных данных *с невысокой скоростью*, что не позволяет их использовать в высокоскоростных магистральных сетях.

Альтернативой коммутации каналов, устраняющей присущие ей недостатки, является коммутация сообщений.

### 1.5.1.2. Коммутация сообщений

**Коммутация сообщений**, в отличие от коммутации каналов, предполагает хранение передаваемых сообщений в буферной памяти промежуточных узлов, находящихся на пути передачи, который прокладывается в каждом узле в соответствии с заданным алгоритмом маршрутизации (рис.1.28). При этом не требуется предварительно устанавливать соединение между взаимодействующими абонентами.



Если абонент  $A_1$  желает передать сообщение  $C$  абоненту  $A_2$ , то он, не устанавливая непосредственное соединение с  $A_2$ , посылает сообщение к узлу связи, к которому он подключён. Там сообщение хранится в буфере узла в течение некоторого времени, необходимого для анализа заголовка, определения в соответствии с заданным алгоритмом маршрутизации следующего узла и, возможно, ожидания освобождения канала связи с этим узлом, если канал занят передачей ранее обработанного сообщения. Проходя таким образом через все узлы, находящиеся на пути передачи, сообщение достигает конечного абонента  $A_2$ . Отметим ещё раз, что направление передачи сообщения, то есть его маршрут в сети, определяется только после поступления сообщения в тот или иной узел сети, а не устанавливается заранее, как это происходит при коммутации каналов.

Благодаря такой организации передачи данных между взаимодействующими абонентами, коммутация сообщений обладает следующими *достоинствами* по сравнению с коммутацией каналов:

- *не требуется предварительное установление соединения*, что существенно снижает накладные расходы, но не делает их нулевыми, поскольку имеются непроизводительные затраты времени в каждом узле на обработку заголовка и реализацию алгоритма маршрутизации; однако в целом эти затраты существенно меньше по сравнению с затратами на установление соединения при коммутации каналов;

- каналы связи на всем пути передачи могут иметь *разные пропускные способности*, поскольку буферирование сообщений в узлах сети позволяет сгладить различие в пропускных способностях входного и выходного каналов узла.

*Недостатками* коммутации сообщений являются:

- *необходимость хранения передаваемых сообщений* в промежуточных узлах, что требует значительной ёмкости буферной

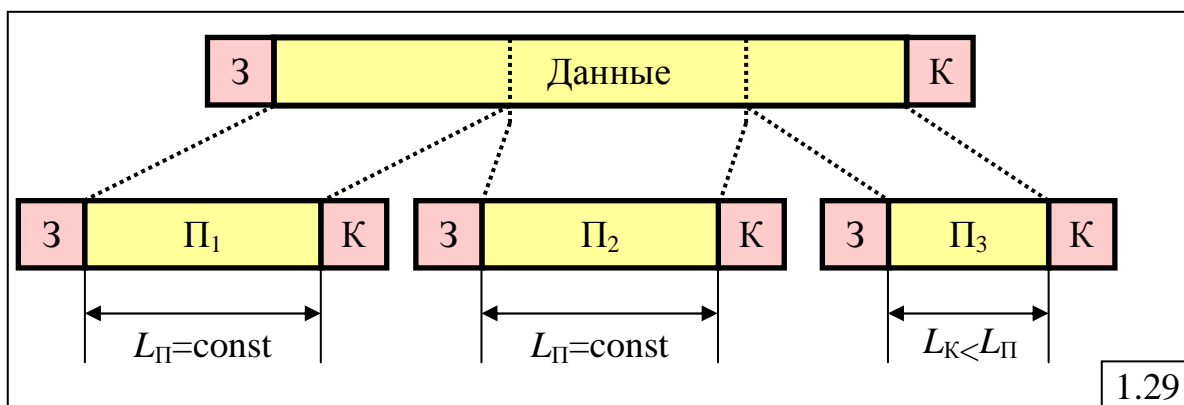
памяти, которая рассчитывается как произведение ёмкости одного буфера на максимально возможное количество сообщений, которые одновременно могут находиться в узле; ёмкость одного буфера должна быть рассчитана на сообщения максимальной длины, которая, например, для видео файлов может составлять несколько гигабайт, что делает ёмкость буферной памяти узла неоправданно большой; при этом коэффициент использования (загрузки) буферной памяти оказывается незначительным, поскольку большинство сообщений, занимая один буфер, будут иметь длину много меньше, чем ёмкость буфера;

- *задержка в промежуточных узлах может оказаться значительной*, особенно из-за большого времени ожидания освобождения выходного канала связи при большой загрузке сети, что приводит к увеличению времени доставки сообщений;

- *монополизация среды передачи (канала связи) на длительный промежуток времени при передаче длинных сообщений приводит к неоправданно большим задержкам коротких сообщений в связи с ожиданием освобождения канала, длительность которого может многократно превышать время непосредственной передачи этих сообщений.*

### 1.5.1.3. Коммутация пакетов

**Коммутация пакетов** отличается от коммутации сообщений лишь тем, что каждое сообщение в сети разбивается на блоки фиксированной длины  $L_{\Pi} = const$  (кроме последнего блока:  $L_{\kappa} \leq L_{\Pi}$ ), называемых **пакетами** (рис.1.29), каждый из которых имеет структуру аналогичную структуре сообщений: заголовок, текст и, возможно, концевик. При этом, заголовки всех пакетов одного и того же сообщения содержат одни и те же адреса назначения и источника. Каждый пакет сообщения передаётся в сети как независимый блок данных в соответствии с адресом назначения, указанным в заголовке.



Коммутация пакетов по сравнению с коммутацией сообщений позволяет реализовать более эффективную передачу данных за счёт следующих присущих ей *достоинств*:

- меньшее время доставки сообщения в сети;
- более эффективное использование буферной памяти в узлах;

- более эффективная организация надёжной передачи данных;
- среда передачи не монополизируется одним сообщением на длительное время;
- задержка пакетов в узлах меньше, чем задержка сообщений.

Рассмотрим каждое из перечисленных достоинств более подробно.

**Уменьшение времени доставки сообщений** при коммутации пакетов достигается за счёт параллельной передачи пакетов по каналам связи. Покажем это на следующем примере.

Положим, что сообщение длиной  $L$  передаётся от абонента  $A_1$  к абоненту  $A_2$  в сети с коммутацией сообщений так, как это показано на рис.1.28. В процессе передачи сообщение проходит через  $K$  каналов связи с одинаковыми пропускными способностями  $C_{KC}$  и  $(K-1)$  промежуточных узлов. Время передачи сообщения длиной  $L$  в одном канале с пропускной способностью  $C_{KC}$  будет равно:  $t = \frac{L}{C_{KC}}$ . Пренебрегая

временем распространения сигнала в канале связи и задержкой сообщения в узлах, определим время доставки сообщения от абонента  $A_1$  к абоненту

$$A_2: T_C = Kt = \frac{KL}{C_{KC}}.$$

Положим теперь, что в рассматриваемой сети реализован принцип коммутации пакетов, и передаваемое от абонента  $A_1$  к абоненту  $A_2$  сообщение длиной  $L$  разбивается на  $n$  пакетов, длина каждого из которых равна  $l = \frac{L}{n}$ . Тогда время передачи пакета в канале с пропускной

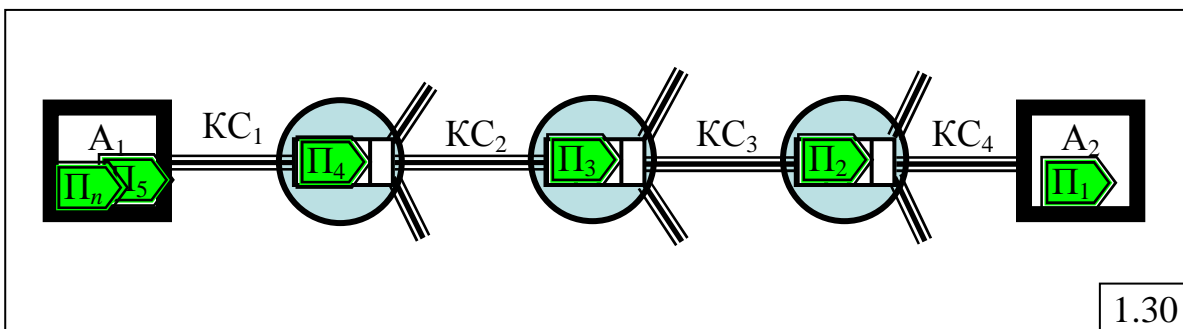
способностью  $C_{KC}$  будет равно:  $t_n = \frac{l}{C_{KC}} = \frac{L}{nC_{KC}}$ . Как и ранее, пренебрегая

временем распространения сигнала в канале связи и задержкой сообщения в узлах, определим время доставки сообщения от абонента  $A_1$  к абоненту  $A_2$ . Очевидно, что первый пакет будет доставлен к абоненту  $A_2$  за время

$$t_1 = Kt_n = \frac{KL}{nC_{KC}}.$$

На момент доставки к абоненту  $A_2$  первого пакета  $\Pi_1$

остальные пакеты  $\Pi_2, \Pi_3, \Pi_4$  сообщения, двигаясь по тому же маршруту, окажутся в промежуточных узлах, как это показано на рис.1.30, а пакеты  $\Pi_5, \dots, \Pi_n$  будут находиться в исходном узле у абонента  $A_1$ .



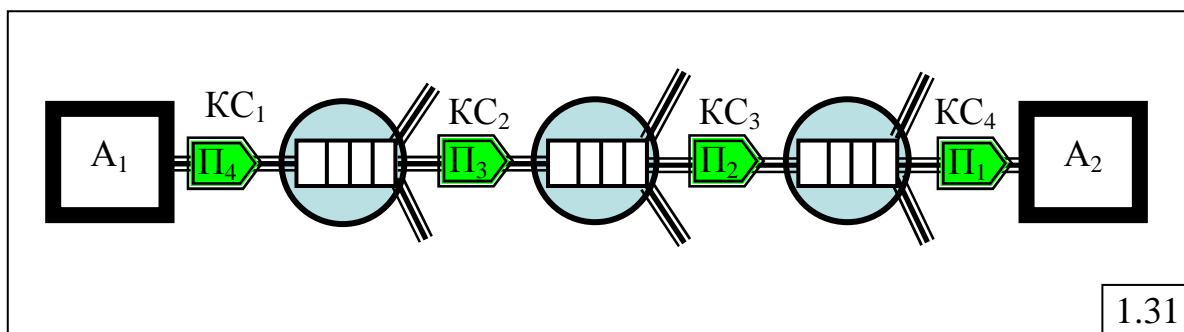
1.30



Дальнейшее перемещение пакетов приведёт к тому, что пакет  $\Pi_2$  окажется у абонента  $A_2$  через время  $t_2 = t_1 + t_{\Pi} = \frac{(K+1)L}{nC_{KC}}$ . Аналогично, пакет  $\Pi_3$  окажется у абонента  $A_2$  через время  $t_3 = t_2 + t_{\Pi} = \frac{(K+2)L}{nC_{KC}}$ , и т.д.. Последним к абоненту  $A_2$  придёт пакет  $\Pi_n$  через время  $t_n = t_{n-1} + t_{\Pi} = \frac{(K+n-1)L}{nC_{KC}}$ . Таким образом, все  $n$  пакетов, а, следовательно, всё сообщение будут доставлены к абоненту  $A_2$  за время  $T_{\Pi} = t_n = \frac{(K+n-1)L}{nC_{KC}}$ .

Сравнивая времена доставки сообщения при использовании коммутации пакетов  $T_{\Pi} = \frac{(K+n-1)L}{nC_{KC}}$  и коммутации сообщений  $T_C = \frac{KL}{C_{KC}}$ , можно убедиться, что при  $n > 1$ :  $T_{\Pi} < T_C$ , т.е. время доставки сообщения при коммутации сообщений больше времени доставки сообщения при использовании коммутации каналов в  $k = \frac{T_C}{T_{\Pi}} = \frac{Kn}{K+n-1}$  раз.

Для значений  $K = 4$  и  $n = 5$  (четыре канала связи, как на рис.1.30, и пять пакетов) получим, что время доставки сообщения при коммутации пакетов уменьшится в  $k = 2,5$  раза по сравнению с коммутацией сообщений, а при разбиении исходного сообщения на 17 пакетов – в  $k = 3,4$  раз. Легко убедиться, что при  $n \rightarrow \infty$  выигрыш  $k$  во времени доставки стремится к  $K$ :  $k \rightarrow K$ , то есть максимально возможный выигрыш при коммутации пакетов определяется количеством каналов связи, через которые проходят пакеты. Этот вывод очевиден, если учесть, что выигрыш во времени доставки обусловлен тем, что разные пакеты сообщения одновременно (параллельно) друг за другом перемещаются в последовательных каналах (рис.1.31): когда пакет  $\Pi_1$  находится в канале  $KC_4$ , пакет  $\Pi_2$  передаётся по каналу  $KC_3$ , пакет  $\Pi_3$  – по каналу  $KC_2$  и пакет  $\Pi_4$  – по каналу  $KC_1$ , что обеспечивает в процессе передачи пакетов уровень параллелизма, равный четырём. Ясно, что чем больше каналов связи на пути пакетов, тем выше уровень параллелизма и, следовательно, тем больше выигрыш.



1.31

Еще больший выигрыш может быть получен, если передача пакетов одного того же сообщения осуществляется параллельно по разным маршрутам.

Представленные выше расчёты выигрыша во времени доставки сообщений при использовании коммутации пакетов по сравнению с коммутацией сообщений естественно являются упрощёнными, поскольку не учитывают задержки пакетов в узлах сети, а также дополнительные накладные расходы на передачу обрамления (заголовков и концевиков) пакетов. Несмотря на это, они достаточно убедительно показывают наличие такого выигрыша.

**Более эффективное использование буферной памяти** при коммутации пакетов по сравнению с коммутацией сообщений обусловлено тем, что размер буфера строго фиксирован и определяется максимально допустимой (фиксированной) длиной передаваемых пакетов, которая может составлять от нескольких десятков байт до нескольких килобайт. За счёт этого достигается более высокая загрузка одного буфера, которая при передаче длинных сообщений близка к единице и, как следствие, более высокая загрузка всей буферной памяти узла.

**Более эффективная организация надежной передачи данных**, по сравнению с коммутацией сообщений, обусловлена тем, что контроль передаваемых данных осуществляется для каждого пакета и в случае обнаружения ошибки повторно передается только один пакет, а не всё сообщение.

**Среда передачи не монополизирована** одним сообщением на длительное время, поскольку длинное сообщение разбивается на пакеты ограниченной длины, которые передаются как независимые единицы данных. При этом механизм управления трафиком организуется таким образом, что после пакета одного сообщения по тому же каналу связи могут быть переданы пакеты других сообщений, а затем снова пакет первого сообщения. Это позволяет уменьшить среднее время ожидания пакетами освобождения канала связи и за счёт этого увеличить оперативность передачи данных. При этом, чем меньше предельно допустимая длина пакетов, тем выше указанный эффект.

**Задержка пакетов в узлах меньше, чем задержка сообщений**, которая складывается из следующих составляющих:

- приём (запись) поступающего блока данных (пакета или сообщения) во входной буфер узла;
- подсчёт и проверка контрольной суммы блока данных;
- передача блока данных из входного буфера в выходной буфер;
- ожидание освобождения выходного канала, занятого передачей ранее поступивших блоков данных;
- передача данных в выходной канал связи и освобождение выходного буфера узла.

Очевидно, что все эти задержки пропорциональны длине блока данных.

Несмотря на очевидные достоинства, коммутации пакетов присущи *недостатки*, которые состоят в следующем:

- *большие накладные расходы* на передачу и анализ заголовков всех пакетов сообщения, что снижает эффективную (реальную) пропускную способность канала связи, используемую непосредственно для передачи данных, и, следовательно, увеличивает время доставки сообщения в сети, в том числе и за счёт дополнительных затрат времени на обработку заголовков пакетов в узлах сети;

- *необходимость сборки из пакетов сообщения* в узле назначения может существенно увеличить время доставки сообщения конечному абоненту за счёт ожидания прихода всех пакетов сообщения, поскольку в случае потери хотя бы одного пакета, сообщение не сможет быть собрано в конечном узле сети; при этом возникает серьёзная проблема, связанная с определением предельно допустимого времени ожидания пакетов для сборки сообщения в конечном узле; при большом значении этого времени в конечном узле может скопиться большое число пакетов разных сообщений, что приведёт к переполнению буферной памяти узла и, как следствие, к потере передаваемых пакетов или к отказу в приёме новых пакетов, что, в свою очередь, не позволит собрать сообщения; маленькое значение предельно допустимого времени ожидания пакетов для сборки сообщения в конечном узле может создавать такую ситуацию, при которой большое количество сообщений не смогут дожидаться прихода последнего пакета и, поскольку по истечении этого времени все пакеты таких сообщений будут удалены из буферной памяти, потребуются повторная передача всех пакетов этих сообщений, что приведёт к значительной загрузке оборудования (узлов и каналов) сети и, в пределе, может вызвать перегрузку сети.

#### 1.5.1.4. Коммутация ячеек

*Коммутация ячеек* – способ коммутации, который можно рассматривать как частный случай коммутации пакетов со строго фиксированной длиной передаваемых блоков данных в 53 байта, называемых *ячейками* (рис.1.32).

Первые компьютерные сети строились для передачи цифровых (компьютерных) данных с единственным требованием – обеспечить надёжную (без ошибок) доставку данных, при этом время доставки не являлось критичным. Развитие компьютерных технологий и появление необходимости передачи мультимедийных данных, таких как речь и видео, выдвинуло, наряду с надёжной доставкой, новое требование к передаче данных в компьютерных сетях: минимизация времени доставки сообщений. Для реализации этой концепции в начале девяностых годов прошлого столетия была разработана сетевая



технология, получившая название **Asynchronous Transfer Mode (АТМ) - режим асинхронной передачи**, назначение которой – передача мультимедийных данных в компьютерной сети с минимальной задержкой.

Как было показано выше при рассмотрении коммутации пакетов, чем короче пакеты, тем меньше время доставки всего сообщения. Исходя из этого, в АТМ-сетях в качестве единицы передачи данных был выбран блок размером в 53 байта (5 байт – заголовок и 48 байт – данные), названный *ячейкой* (рис.1.32). Столь странный размер ячейки появился в результате компромисса двух противодействующих групп, из которых одна группа (по одной из версий: традиционные связисты – телефонисты) настаивала на меньшем значении поля данных в 32 байта, а другая (компьютерщики) – на значении в 64 байта. Действительно, меньшее значение размера ячейки обеспечило бы меньшие задержки при доставке данных, однако не следует забывать, что при этом возрастают накладные расходы на передачу заголовков ячеек, что снижает полезную (эффективную) пропускную способность среды передачи. В АТМ-сетях это снижение составляет около 10%. Если же размер поля данных будет 32 байта, то при том же заголовке в 5 байт снижение полезной пропускной способности составит 13,5%. Принимая во внимание, что в мультимедийных сетях обычно используются высокоскоростные каналы, потери пропускной способности могут оказаться значительными, что отрицательно скажется на экономической эффективности компьютерной сети.

Подводя итог сказанному, можно отметить следующие *достоинства* коммутации ячеек:

- маленькие задержки ячеек (не монополизирован канал связи);
- быстрая обработка заголовка ячейки в узлах, поскольку местоположение заголовка строго фиксировано;
- более эффективная, по сравнению с коммутацией пакетов, организация буферной памяти и надежной передачи данных.

Основным *недостатком* коммутации ячеек является:

- наличие сравнительно больших накладных расходов на передачу заголовка (почти 10%) и, как следствие, значительная потеря пропускной способности, особенно в случае высокоскоростных каналов связи.

*Коммутация пакетов и коммутация каналов* – основные способы передачи данных в компьютерных сетях, поскольку коммутация пакетов обеспечивает более эффективную передачу данных через СПД по сравнению с коммутацией сообщений (в первую очередь, значительно меньшие задержки), а коммутация каналов может быть достаточно легко реализована на основе существующей телефонной сети.

### 1.5.2. Способы передачи пакетов

Пакеты в сети могут передаваться двумя способами (рис.1.33):

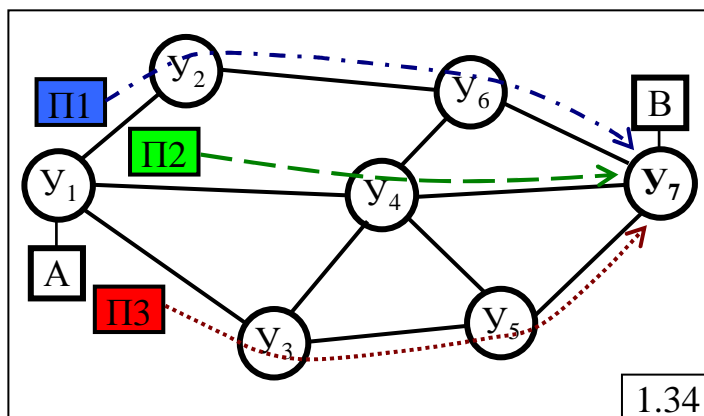
- *дейтаграммным*;

- путем формирования «виртуального канала».



### 1.5.2.1. Дейтаграммная передача

При дейтаграммном способе пакеты одного и того же сообщения могут передаваться между двумя взаимодействующими пользователями **A** и **B** по разным маршрутам, как это показано на рис.1.34, где пакет **П1** передаётся по маршруту  $Y_1-Y_2-Y_6-Y_7$ , пакет **П2** – по маршруту  $Y_1-Y_4-Y_7$  и пакет **П3** – по маршруту  $Y_1-Y_3-Y_5-Y_7$ . В результате такого способа передачи все пакеты приходят в конечный узел сети в разное время и в произвольной последовательности. Пакеты одного и того же сообщения, рассматриваемые в каждом узле сети как самостоятельные независимые единицы данных и передаваемые разными маршрутами, называются **дейтаграммами** (datagram). В узлах сети для каждой дейтаграммы всякий раз определяется наилучший путь передачи в соответствии с выбранной метрикой маршрутизации, независимо от того, по какому пути переданы были предыдущие дейтаграммы с такими же адресами назначения (получателя) и источника (отправителя).



Дейтаграммный способ передачи пакетов может быть реализован:

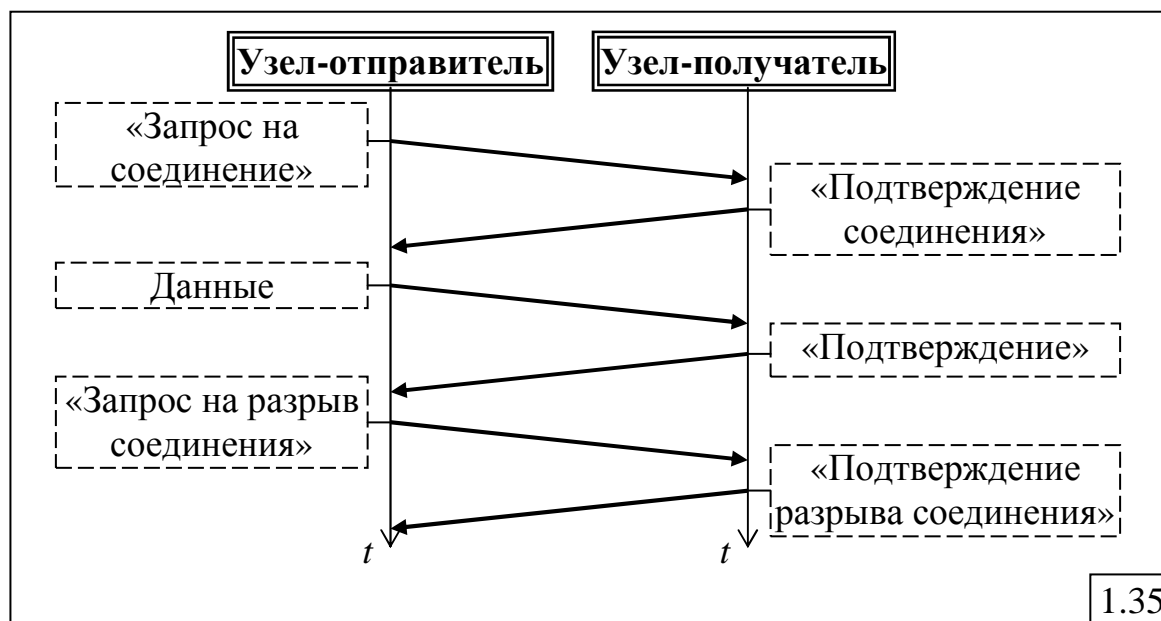
- без установления соединения между абонентами сети;
- с установлением соединения между взаимодействующими абонентами сети.

В последнем случае между взаимодействующими абонентами предварительно устанавливается соединение путём обмена служебными пакетами: «запрос на соединение» и «подтверждение соединения», означающее готовность принять передаваемые данные. В процессе установления соединения могут «оговариваться» значения параметров передачи данных, которые должны выполняться в течение сеанса связи.

После установления соединения отправитель начинает передачу, причём пакеты одного и того же сообщения могут передаваться разными маршрутами, то есть дейтаграммным способом. По завершении сеанса передачи данных выполняется процедура разрыва соединения путём обмена служебными пакетами: «запрос на разрыв соединения» и «подтверждение разрыва соединения». Описанная процедура передачи пакетов с установлением соединения иллюстрируется на диаграмме (рис.1.35).

Достоинствами дейтаграммного способа передачи пакетов в компьютерных сетях являются:

- простота организации и реализации передачи данных – каждый пакет (дейтаграмма) сообщения передаётся независимо от других пакетов;
- в узлах сети для каждого пакета выбирается наилучший путь (маршрут);
- передача данных может выполняться как без установления соединения между взаимодействующими абонентами, так и при необходимости с установлением соединения.



К недостаткам дейтаграммного способа передачи пакетов следует отнести:

- необходимость сборки сообщения в конечном узле: сообщение не может быть передано получателю, пока в конечном узле сети не соберутся все пакеты данного сообщения, поэтому в случае потери хотя бы одного пакета сообщение не сможет быть сформировано и передано получателю;
- при длительном ожидании пакетов одного и того же сообщения в конечном узле может скопиться достаточно большое количество пакетов сообщений, собранных не полностью, что требует значительных затрат на организацию в узле буферной памяти большой ёмкости;
- для предотвращения переполнения буферной памяти узла время нахождения (ожидания) пакетов одного и того же сообщения в конечном узле ограничивается, и по истечении этого времени все поступившие

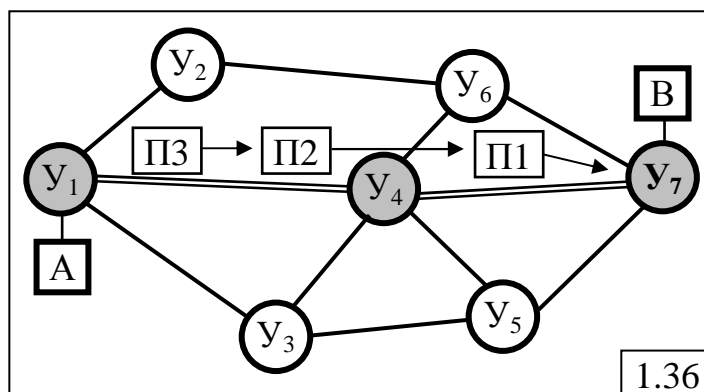
пакеты не полностью собранного сообщения уничтожаются, после чего выполняется запрос на повторную передачу данного сообщения; это приводит к *увеличению нагрузки* на сеть и, как следствие, к *снижению её производительности*, измеряемой количеством сообщений, передаваемых в сети за единицу времени.

#### 1.5.2.2. Виртуальный канал

Способ передачи пакетов «**виртуальный канал**» заключается в формировании единого «виртуального» канала на время взаимодействия абонентов для передачи всех пакетов сообщения. Этот способ реализуется с использованием предварительного установления соединения между взаимодействующими абонентами, в процессе которого формируется наиболее рациональный единый для всех пакетов маршрут, по которому, в отличие от дейтаграммного способа, все пакеты сообщения передаются в *естественной последовательности*, как это показано на рис.1.36.

Пакеты П1, П2 и П3 сообщения передаются в естественной последовательности от пользователя А к пользователю В по предварительно созданному виртуальному каналу через узлы  $Y_1$ - $Y_4$ - $Y_7$ .

Виртуальный канал, как и реальный физический канал в случае коммутации каналов, существует только в течение сеанса связи, при этом ресурсы реальных каналов связи (пропускная способность) и узлов сети (буферная память), находящихся на маршруте, резервируются на всё время сеанса.



Не следует путать и смешивать *коммутацию каналов* и *способ передачи пакетов «виртуальный канал»*. Основное их отличие состоит в том, что «виртуальный канал» реализуется с промежуточным хранением пакетов в узлах сети, в то время как коммутация каналов реализуется без промежуточного хранения передаваемых пакетов за счёт создания реального (а не виртуального) физического канала между абонентами сети.

К *достоинствам* способа передачи пакетов «виртуальный канал» по сравнению с дейтаграммной передачей пакетов можно отнести:

- *меньшие задержки в узлах* сети, обусловленные резервированием ресурсов, и прежде всего пропускной способности каналов связи, в процессе установления соединения;
- *небольшое время ожидания* в конечном узле для сборки всего сообщения, поскольку пакеты передаются последовательно друг за другом по одному и тому же маршруту (виртуальному каналу), и вероятность того, что какой-либо пакет «заблудится» в результате неудачно выбранного

маршрута или его время доставки окажется слишком большим, как это может произойти при дейтаграммном способе, близка к нулю;

- более *эффективное использование буферной* памяти промежуточных узлов за счёт её предварительного резервирования, а также буферной памяти в конечном узле в связи с небольшим временем ожидания прихода всех пакетов сообщения.

К *недостаткам* способа передачи пакетов «виртуальный канал» можно отнести:

- *наличие накладных расходов* (издержек) на установление соединения;

- *неэффективное использование ресурсов* сети, поскольку они резервируются на всё время взаимодействия абонентов (сеанса) и не могут быть предоставлены другому соединению, даже если они в данный момент не используются.

### 1.5.3. Маршрутизация

*Маршрутизация* – одна из основных функций компьютерной сети, определяющая эффективность передачи данных. Проблема маршрутизации в компьютерных сетях аналогична проблеме организации автомобильного движения по улицам города и состоит в выборе в каждом узле сети направления передачи данных (выходного канала) из множества возможных направлений в соответствии с адресом назначения и с учётом требований, предъявляемых к качеству передачи. Очевидно, что наиболее просто маршрутизация реализуется в узлах с двумя каналами: данные, поступившие по одному каналу, автоматически направляются в другой канал узла.

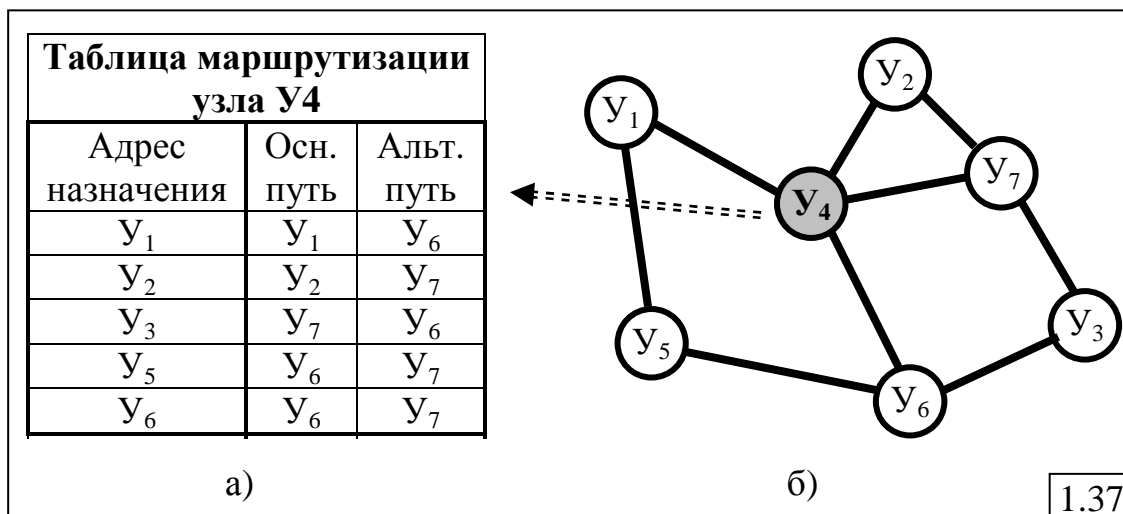
#### 1.5.3.1. Таблица маршрутизации

При наличии нескольких выходных каналов, по которым могут быть переданы данные, маршрутизация реализуется на основе **таблицы маршрутизации**, вид которой зависит от используемого в сети алгоритма маршрутизации. В простейшем случае каждому адресу назначения ставится в соответствие адрес следующего соседнего узла, к которому должен быть направлен пакет с указанным в заголовке адресом назначения. При наличии *альтернативных маршрутов*, например в многосвязных сетях, дополнительно могут быть указаны адреса других соседних узлов, через которые проходят альтернативные маршруты. При этом для каждого маршрута задаётся значение некоторой *метрики*, на основе которой выбирается тот или иной маршрут. В качестве метрики может использоваться расстояние до узла назначения, измеряемое, например, в хопах, пропускная способность соответствующего выходного канала связи и т.д.

На рис. 1.37,а) показан пример простейшей *таблицы маршрутизации* узла 4 сети с топологией, представленной на рис.1.37,б). В таблице для каждого адреса назначения указывается направление передачи

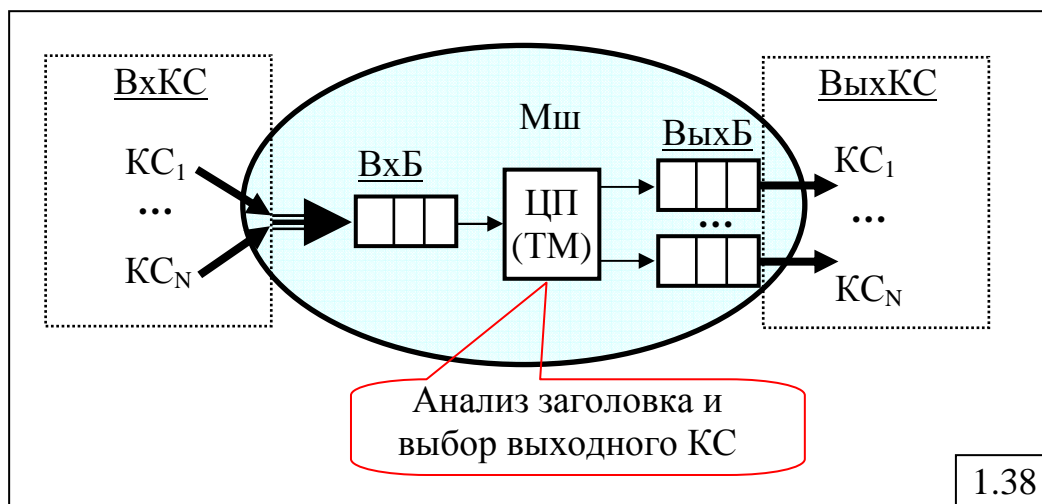


данных по основному пути и альтернативному пути в случае невозможности передачи по основному пути, например в случае отказа основного пути.



### 1.5.3.2. Модель маршрутизатора

На рис.1.38 показана концептуальная модель функционирования маршрутизатора. По входным каналам связи (ВхКС) пакеты с данными поступают во входной буфер (ВхБ) маршрутизатора. Центральный процессор (ЦП) последовательно анализирует заголовки пакетов и в соответствии с таблицей маршрутизации (ТМ) определяет направление передачи пакета и соответствующий выходной канал связи (ВыхКС). Затем пакет направляется в выходной буфер (ВыхБ) этого канала, где он ожидает освобождения канала, если последний занят передачей предыдущих пакетов. Пакеты, находящиеся в выходном буфере, образуют очередь перед каналом связи. Для дифференцированного обслуживания пакетов разных типов, имеющих разные требования к качеству обслуживания, выбор очередного пакета для передачи по каналу связи может осуществляться в соответствии с некоторой, например приоритетной, дисциплиной обслуживания.

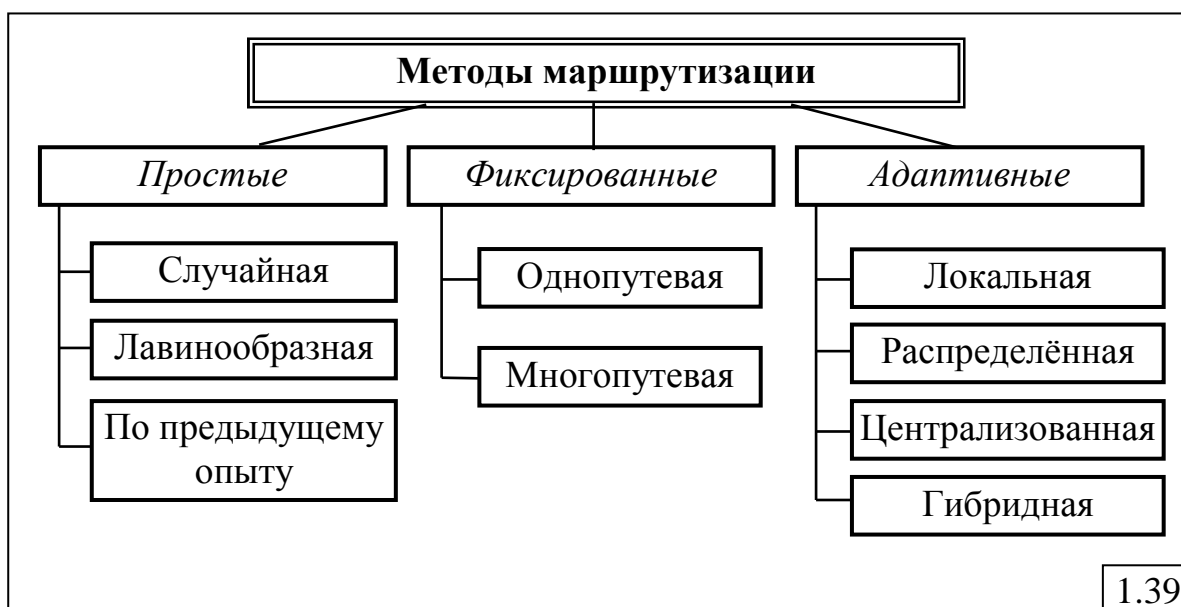


Отметим, что конкретные реализации реальных маршрутизаторов могут существенно различаться. Например, буферная память для хранения пакетов может быть разделена на входную и выходную, как это показано на рис.1.38, либо выполнена как единая память, где хранятся все пакеты.

### 1.5.3.3. Классификация методов маршрутизации

В компьютерных сетях теоретически могут использоваться самые разнообразные методы (алгоритмы) маршрутизации, обеспечивающие разные эффекты и зависящие от структурно-функциональных особенностей сети и требований, предъявляемых к качеству функционирования сети. На рис.1.39 представлена одна из возможных классификаций различных методов маршрутизации, которые можно разбить на 3 группы:

- простые;
- фиксированные;
- адаптивные.



1.39

### 1.5.3.4. Простые методы маршрутизации

К *простым* относятся следующие методы маршрутизации.

1. **Случайная маршрутизация**, при которой выбор направления передачи данных осуществляется случайным образом между всеми каналами узла (портами маршрутизатора) за исключением канала, по которому эти данные поступили в узел. Например, если маршрутизатор имеет 4 порта (канала) и по одному из них поступили данные, то вероятность передачи по любому из трёх других каналов будет равна  $1/3$ , при этом адрес назначения не используется для выбора выходного канала. Очевидно, что применение такого алгоритма маршрутизации оправдано только в том случае, если отсутствует информация о топологии сети и сетевых адресах, что делает невозможным построение таблицы маршрутизации. Несмотря на то, что направление передачи пакетов

осуществляется случайным образом, вероятность доставки пакета конечному адресату отлична от нуля, но меньше единицы, поскольку существует вероятность заикливания пакета в сети.

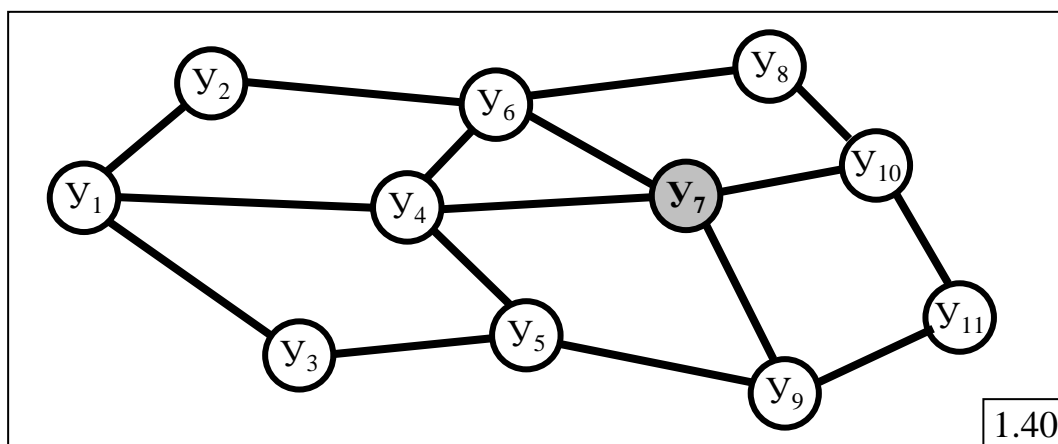
2. **Лавинообразная маршрутизация**, заключающаяся в размножении поступившего пакета и рассылке его копии по всем направлениям, кроме того, по которому поступил пакет. Такое размножение приводит к большой загрузке каналов и узлов сети и, в то же время, обеспечивает доставку пакета конечному адресату с вероятностью единица. Если в некоторый узел поступает несколько копий одного и того же пакета, то все они, кроме одного пакета, уничтожаются. Как и в предыдущем случае, применение лавинообразного алгоритма оправдано, если невозможно построить таблицу маршрутизации из-за отсутствия информации о топологии сети и сетевых адресах.

Случайная и, особенно, лавинообразная маршрутизации являются скорее экзотическими методами, которые могут использоваться в условиях неопределённости, возникших, например, в результате боевых действий, приведших к выходу из строя значительной части сетевых ресурсов и, как следствие, к отсутствию сведений о топологии сети и сетевых адресах.

3. **Маршрутизация по предыдущему опыту** является наиболее интересной среди простых алгоритмов и заключается в автоматическом построении таблицы маршрутизации.

Заголовки пакетов, передаваемых в сети с маршрутизацией от источника, кроме адреса получателя АП и адреса отправителя АО содержат специальное поле «расстояние», значение которого в узле-отправителе устанавливается равным 0. В процессе передачи пакета в сети в каждом промежуточном узле значение поля «расстояние» увеличивается на единицу. Таким образом, в каждом узле значение этого поля показывает расстояние, которое прошел пакет от узла-отправителя до этого узла и которое измеряется в количестве промежуточных узлов на пути передачи пакета.

Рассмотрим принцип реализации метода маршрутизации по предыдущему опыту на примере построения таблицы маршрутизации узла  $Y_7$  сети, показанной на рис. 1.40.



Положим, что в некоторый момент времени в узел  $Y_7$  поступает пакет от узла  $Y_1$  (адрес отправителя АО =  $Y_1$ ) по маршруту, пролегающему через узлы  $Y_3, Y_5, Y_9$ , в каждом из которых значение поля «расстояние» было увеличено на 1:

$$Y_1 \rightarrow Y_3 \rightarrow Y_5 \rightarrow Y_9 \rightarrow Y_7.$$

В узле  $Y_7$  анализируется адрес отправителя АО и, если такой адрес отсутствует в таблице маршрутизации, то он заносится в таблицу. Для этого же адреса одновременно указывается адрес соседнего узла  $Y_9$ , от которого поступил этот пакет, и расстояние до узла-отправителя, содержащееся в поле «расстояние» заголовка пакета и равное 3 (рис.1.41).

Таблица маршрутизации узла $Y_7$ (1)		
Адрес назначения	Адрес соседнего узла	Расстояние
$Y_1$	$Y_9$	3
...	...	...

1.41

Положим теперь, что через некоторое время в узел  $Y_7$  поступит новый пакет от узла  $Y_1$  с адресом отправителя АО= $Y_1$ , маршрут которого пролегал через узлы  $Y_2, Y_6$ :

$$Y_1 \rightarrow Y_2 \rightarrow Y_6 \rightarrow Y_7.$$

Поскольку в таблице маршрутизации узла  $Y_7$  уже есть строка с адресом  $Y_1$ , то для этого адреса сравнивается значение расстояния от узла отправителя, содержащееся в заголовке поступившего пакета в поле «расстояние», со значением расстояния, записанного в таблице маршрутизации. Если расстояние, указанное в заголовке поступившего пакета, меньше расстояния, записанного в таблице маршрутизации, то новый маршрут считается более коротким и в таблице маршрутизации корректируется строка, соответствующая адресу назначения  $Y_1$ : адрес соседнего узла заменяется на  $Y_6$ , а расстояние до адреса назначения становится равным 2 (рис.1.42).

Таблица маршрутизации узла $Y_7$ (2)		
Адрес назначения	Адрес соседнего узла	Расстояние
$Y_1$	$Y_6$	2
...	...	...

1.42

И наконец, если через некоторое время в узел  $U_7$  поступит пакет от узла  $U_1$ , маршрут которого пролегал через узел  $U_4$ :  $U_1 \rightarrow U_4 \rightarrow U_7$ , то поле «расстояние» в заголовке пакета будет иметь значение, равное 1, что меньше значения в таблице маршрутизации. Тогда после корректировки таблицы маршрутизации, соответствующая строка примет вид, показанный на рис.1.43.

Таблица маршрутизации узла $U_7$ (3)		
Адрес назначения	Адрес соседнего узла	Расстояние
$U_1$	$U_4$	1
...	...	...

1.43

Таким образом, анализируя всякий раз адрес отправителя и расстояние от узла-отправителя всех проходящих через узел пакетов и корректируя таблицу маршрутизации, после некоторого времени получим в таблице маршрутизации наилучший маршрут, по которому будут передаваться пакеты с соответствующим адресом назначения. В нашем примере поступающие в узел  $U_7$  пакеты с адресом назначения с  $АН=U_1$  будут направляться в узел  $U_4$ .

#### 1.5.3.5. Методы фиксированной маршрутизации

Вторую группу методов маршрутизации образуют методы *фиксированной маршрутизации* (см. рис.1.39), предполагающие наличие таблицы маршрутизации, которая формируется в узле администратором сети и не изменяется, по крайней мере, в течение длительного периода функционирования сети.

Фиксированная маршрутизация может быть:

- **однопутевая**, когда таблица маршрутизации содержит для каждого адреса назначения только один маршрут, и пакеты с одним и тем же адресом назначения направляются всегда к одному и тому же узлу;
- **многопутевая**, когда таблица маршрутизации содержит для каждого адреса назначения несколько маршрутов (адресов соседних узлов), по которым могут быть направлены пакеты с одним и тем же адресом назначения.

*Достоинством* фиксированной маршрутизации, несомненно, следует считать простоту реализации.

В то же время существенным *недостатком* фиксированной маршрутизации является отсутствие гибкости, что проявляется в невозможности изменения маршрутов при изменении состава и топологии сети, а также при отказах узлов и каналов связи. В связи с этим, такие

методы маршрутизации могут применяться только в небольших и не изменяющихся в течение длительного промежутка времени сетях.

#### **1.5.3.6. Методы адаптивной маршрутизации**

*Адаптивная* или *динамическая маршрутизация* (рис.1.39) предполагает оперативное изменение таблиц маршрутизации при изменении состава и топологии сети, а также при отказах узлов и каналов связи. Адаптивная маршрутизация может быть реализована как:

- локальная;
- распределённая;
- централизованная;
- гибридная.

**Локальная маршрутизация** означает, что таблица маршрутизации изменяется (корректируется) на основе локальной информации о состоянии соответствующего узла, например о загрузке выходных каналов узла или о количестве пакетов, ожидающих в очереди освобождения выходного канала. При этом, если загрузка некоторого канала оказывается значительной, то таблица маршрутизации корректируется таким образом, чтобы выровнять загрузки всех выходных каналов.

*Недостаток* локальной маршрутизации состоит в том, что выбранный на основе локальной информации о состоянии узла маршрут может оказаться плохим, если соседний узел, к которому направляются пакеты, перегружен.

При **распределённой маршрутизации** корректировка таблицы маршрутизации осуществляется на основе не только локальной информации о состоянии соответствующего узла, но и с учётом состояний соседних узлов сети. Для этого узлы могут обмениваться специальными служебными пакетами, содержащими информацию о состоянии соседних узлов.

Недостаток распределённой маршрутизации очевиден – служебные пакеты создают дополнительную нагрузку в каналах и узлах сети, что при неудачной организации может существенно снизить производительность среды передачи данных, измеряемую количеством пакетов, передаваемых с сети за единицу времени.

**Централизованная маршрутизация** предполагает наличие в сети специально выделенного узла, собирающего и анализирующего информацию о состоянии всех узлов сети. Результаты анализа рассылаются в виде служебных пакетов всем узлам, которые на их основе корректируют свои таблицы маршрутизации. Несмотря на кажущуюся эффективность такой маршрутизации, результирующий эффект может оказаться незначительным и даже привести к снижению эффективности передачи данных по сравнению с распределённой маршрутизацией в связи со значительным ростом числа передаваемых служебных пакетов, существенно загружающих каналы связи и сеть передачи данных в целом.

*Гибридная маршрутизация* представляет собой любую комбинацию рассмотренных выше методов маршрутизации.

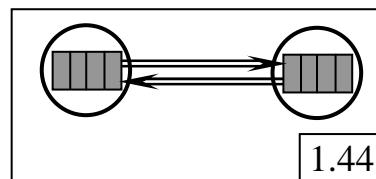
На практике в современных сетях передачи данных реализованы только некоторые из рассмотренных выше методов маршрутизации, причём конкретная реализация в маршрутизаторах разных фирм может быть различной и часто является секретом фирмы-разработчика.

#### 1.5.4. Задачи управления трафиком

Необходимость управления трафиком в сети обусловлена следующими **особенностями**, присущими сетевому трафику современных компьютерных сетей:

- *неоднородность трафика*, характеризующаяся наличием в сети нескольких типов данных, которые можно разделить на две большие группы: мультимедийные (речь, аудио и видео) и компьютерные (электронные письма, файлы и т.п.)
- наличие *различных (дифференцированных) требований* к качеству передачи данных разных типов;
- *случайный характер и нестационарность* сетевого трафика, обусловленные изменением интенсивностей потоков данных в различное время суток и непредсказуемостью характера и темпа работы пользователей в компьютерной сети;
- в свою очередь, нестационарность сетевого трафика может привести к возникновению в компьютерной сети периодов *перегрузок* и даже к блокировкам.

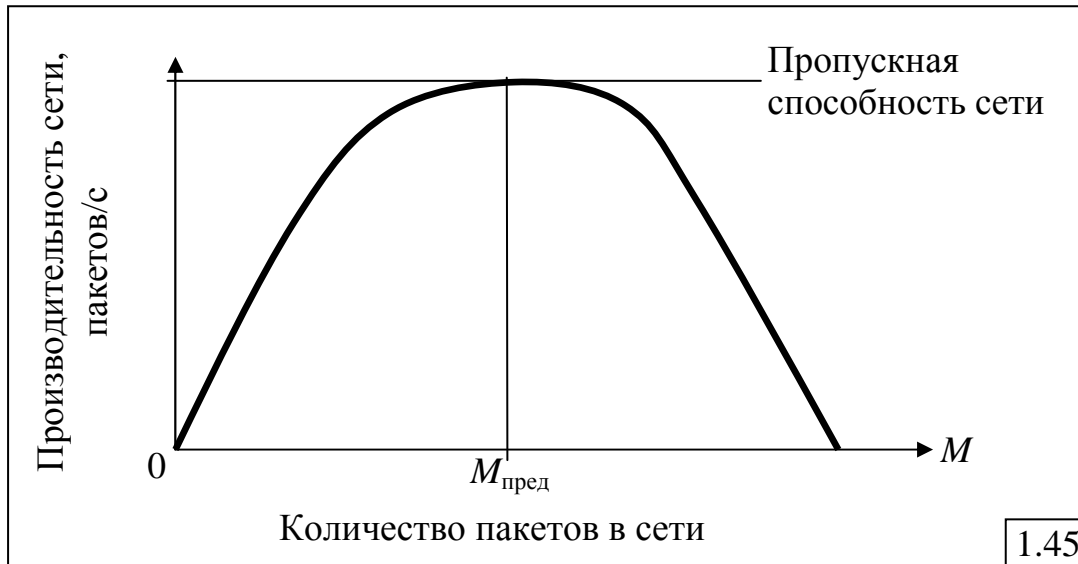
Блокировки в сети могут возникнуть в результате заполнения буферной памяти узлов. Простейший пример блокировок показан на рис.1.44, где буферы двух соседних узлов, желающих обменяться пакетами, заполнены до конца. Это приводит к ситуации, когда обмен пакетами невозможен, несмотря на то, что в принципе буферной памяти достаточно для хранения имеющихся пакетов. Однако, для того чтобы принять пакет от соседнего узла, необходимо иметь хотя бы один свободный буфер. Таким образом, узлы оказываются заблокированными, что может, в конечном счете, привести к остановке (блокировке) всей сети.



На рис.1.45 показана зависимость производительности сети передачи данных, измеряемая количеством пакетов, передаваемых в сети за единицу времени, от количества пакетов, находящихся в сети.

Вначале производительность сети передачи данных, как и следовало ожидать, растёт с увеличением количества находящихся в сети пакетов  $M$ , достигая при  $M_{\text{пред}}$  некоторого предельного значения, представляющего собой *пропускную способность сети*. При этом загрузка, по крайней мере, одного из узлов или каналов связи, называемого **узким местом**, достигает при  $M = M_{\text{пред}}$  значения 1, что приводит к *перегрузке* сети. Дальнейшее

увеличение количества пакетов в сети не приводит к росту производительности, значение которой будет определяться производительностью узкого места. Более того, дальнейшее увеличение количества пакетов в сети ведёт к снижению производительности и даже к прекращению передачи пакетов, то есть к остановке сети, что, в частности, связано с возникающими в сети блокировками.



Перечисленные выше особенности компьютерных сетей обуславливают необходимость управления неоднородным трафиком в сети для решения следующих задач:

- 1) *обеспечение надежной передачи* данных, предполагающей доставку данных абоненту без потерь и без искажения данных (за счет применения механизмов квитирования и тайм-аута);
- 2) *обеспечение эффективной загрузки* дорогостоящего сетевого оборудования (каналов и узлов) сети (за счет реализации механизма скользящего окна и перераспределения потоков данных в процессе адаптивной маршрутизации);
- 3) *малые задержки* при передаче по сети сообщений и, прежде всего, мультимедийных (за счет маршрутизации и приоритетов);
- 4) *предотвращение перегрузок и блокировок* при передаче данных (за счет приоритетов и ограничения входящего в сеть трафика).

Управление потоком данных реализуется на различных уровнях OSI-модели. Ниже рассматриваются некоторые наиболее типичные методы управления трафиком на первых трёх уровнях, а именно:

- процедура бит-стаффинга, используемая на физическом уровне;
- механизм «скользящего окна», используемый на канальном уровне;
- методы маршрутизации, используемые на сетевом уровне.



## 1.5.5. Методы управления трафиком на физическом уровне

### 1.5.5.1. Способы разделения кадров

На физическом уровне для разделения потока битов, соответствующих разным блокам данных 2-го уровня – кадрам, могут использоваться различные способы:

1) указание в заголовке кадра его длины и подсчет количества символов в процессе приема потока данных (основной недостаток – неустойчивость к помехам);

2) использование в качестве границы кадров запрещенных сигналов физического уровня;

3) использование в качестве границы кадров специальных стартовых и стоповых символов (байтов);

4) использование в качестве границы кадров специальных последовательностей битов.

### 1.5.5.2. Бит-стаффинг

При использовании в качестве границы кадров специальных последовательностей битов (например, в протоколах семейства HDLC используется специальная последовательность битов: 01111110, называемая *флагом*) возникает проблема, связанная с тем, что такая последовательность битов может встретиться внутри кадра и будет ошибочно воспринята аппаратурой передачи данных как обрамление, т.е. как начало следующего кадра. Для исключения этого используется процедура обеспечения прозрачности канала – бит-стаффинг.

**Бит-стаффинг** (*bit stuffing – вставка битов*) – техника вставки и стирания битов, используемая в высокоскоростных цифровых каналах связи с большим числом линий связи, не имеющих взаимной синхронизации, а также средство синхронизации в протоколах управления каналом связи типа HDLC.

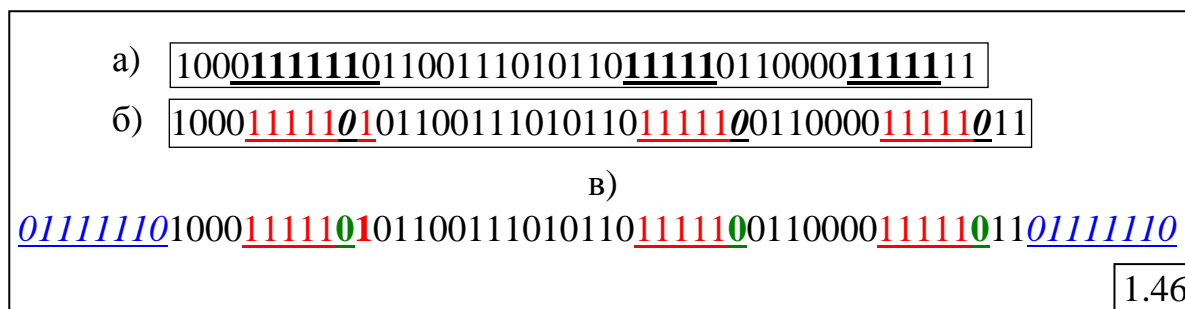
Бит-стаффинг реализуется следующим образом. На передающем узле после пяти подряд следующих единиц внутри кадра принудительно вставляется 0, который автоматически изымается на приемном узле. Таким образом, исключается возможность появления внутри кадра последовательности битов 01111110, используемой для разделения кадров.

На рис.1.46 иллюстрируется процедура бит-стаффинга. Положим, что необходимо передать кадр, показанный на рис.1.46,а), в котором встречаются:

- шесть подряд идущих единиц (которые находятся между двумя нулевыми битами и могут быть восприняты как граница кадра);
- ровно пять единиц;
- более шести единиц.

В соответствии с рассмотренным выше принципом реализации бит-стаффинга в передающем узле после пяти любых подряд идущих единиц принудительно будут вставлены нулевые биты, как это показано на

рис.1.46,б). Отметим, что нули вставляются не зависимо от того, совпадает или не совпадает внутрикадровая битовая последовательность с флагом 01111110, используемым для разделения кадров. В результате такой процедуры по каналу связи будет передана последовательность битов, показанная на рис.1.46,в), которая анализируется в принимающем узле. Если после пяти подряд поступивших в узел единиц два последующих бита имеют значения 1 и 0, то такая комбинация рассматривается как граница кадра. Если же после пяти единиц следующий бит равен 0, то он изымается, и текст кадра принимает исходный вид рис.1.46,а).



При использовании в качестве границы кадров специальных стартовых и стоповых символов (байтов) реализуется «байт-стаффинг» – техника вставки, а точнее замены байтов, совпадающих с граничными в тексте кадра, на определённые последовательности других символов (см. протокол SLIP).

### 1.5.6. Управление трафиком на канальном уровне

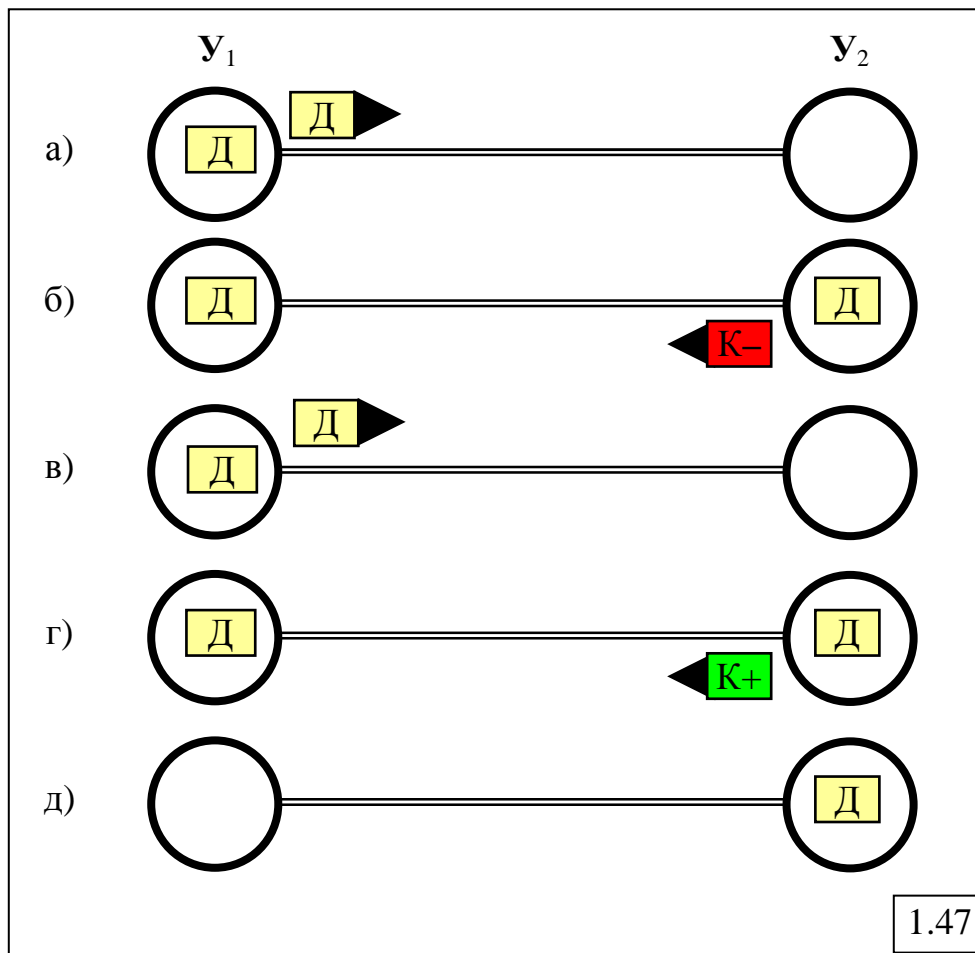
На канальном уровне управление потоком в канале связи между двумя узлами реализуется за счет применения:

- механизма квитирования;
- механизма тайм-аута;
- механизма скользящего окна.

#### 1.5.6.1. Квитирование

Механизм квитирования предназначен для обеспечения надёжной передачи данных (кадров или пакетов) и может быть реализован как на канальном, так и на более высоком уровне (например, сетевом или транспортном) OSI-модели. Реализация механизма квитирования на канальном уровне (в звене передачи данных) показана на рис.1.47. Положим, что в некоторый момент времени узел  $У_1$  отправляет в узел  $У_2$  кадр с данными (Д), причём копия отправленного кадра сохраняется в буферной памяти узла (рис.1.47,а). Узел  $У_2$  после получения кадра от узла  $У_1$  подсчитывает контрольную сумму и сравнивает её со значением, содержащимся в концеике. Если эти значения не совпадают, то узел  $У_2$  формирует и отправляет узлу  $У_1$  специальный служебный кадр (К–), называемый *отрицательной квитанцией*, свидетельствующей о том, что кадр был передан с ошибкой (рис.1.47,б). Узел  $У_1$  анализирует квитанцию и, если квитанция была отрицательной, повторно посылает тот же самый

кадр Д (рис.1.47,в). Если подсчитанная в узле  $У_2$  контрольная сумма совпадает со значением, содержащимся в концеике, то узел  $У_2$  формирует и отправляет **положительную квитанцию (К+)**, свидетельствующую о том, что кадр был передан без ошибок (рис.1.47,г). Узел  $У_1$  анализирует квитанцию и, если квитанция была положительной, удаляет сохранённую копию этого кадра из буферной памяти (рис.1.47,д).



### 1.5.6.2. Тайм-аут

Недостаток рассмотренного механизма квитирования состоит в том, что в случае потери кадра данных или квитанции в процессе передачи между узлами  $У_1$  и  $У_2$  узел-отправитель  $У_1$  может ожидать прихода квитанции бесконечно долго. При этом становится невозможной передача других данных к узлу  $У_2$ , что может привести в конечном счёте к прекращению передачи данных в сети. Для исключения подобной ситуации был реализован **механизм тайм-аута**, заключающийся в следующем. Узел-отправитель  $У_1$  после завершения передачи данных (кадра) к узлу  $У_2$  запускает таймер и ожидает поступления квитанции (положительной или отрицательной) в течение ограниченного промежутка времени  $\Delta t$ , называемого **тайм-аутом**.

Величина тайм-аута выбирается из следующего условия:  $\Delta t$  должно быть больше, чем удвоенное время передачи кадра между узлами, то есть  $\Delta t > 2\tau_k$ . Время передачи кадра между узлами  $\tau_k$  складывается из времени

распространения сигнала по каналу связи  $\tau_c$  и времени передачи кадра максимальной длины  $\tau_{\max}$ :  $\tau_k = \tau_c + \tau_{\max}$ .

Время распространения сигнала по каналу связи определяется как  $\tau_c = \frac{L}{v}$ , где  $L$  – длина канала и  $v$  – скорость распространения сигнала в среде передачи. Время передачи кадра максимальной длины зависит от длины кадра  $l_{\max}$  и пропускной способности канала  $C_{KC}$ :  $\tau_{\max} = \frac{l_{\max}}{C_{KC}}$ .

Тогда:  $\tau_k = \frac{L}{v} + \frac{l_{\max}}{C_{KC}}$  и условие для выбора величины тайм-аута примет

$$\text{вид: } \Delta t > 2 \left( \frac{L}{v} + \frac{l_{\max}}{C_{KC}} \right).$$

Если по истечении тайм-аута  $\Delta t$  узел-отправитель  $U_1$  не получает квитанцию, то он повторно передаёт тот же кадр. Для исключения бесконечного числа передач одного и того же кадра обычно устанавливается некоторое предельное количество попыток передать кадр, после которого передача этого кадра прекращается, и данное направление передачи (маршрут) исключается из рассмотрения и в дальнейшем не используется, поскольку предполагается, что канал или узел данного маршрута находится в неисправном состоянии. Для рассматриваемого кадра выбирается новое направление передачи в соответствии с используемым методом маршрутизации.

*Недостатком* рассмотренного способа передачи данных является низкий коэффициент полезной загрузки канала, обусловленный большими накладными расходами на ожидание и передачу служебных квитанций. Коэффициент полезной загрузки канала, измеряемый как доля времени, используемого для передачи непосредственно пользовательских данных, составляет менее 30%, в чём несложно убедиться, используя следующие рассуждения.

Пусть время передачи кадра данных равно  $\tau_k$ . Как показано выше, величина тайм-аута выбирается из условия:  $\Delta t > 2(\tau_c + \tau_{\max})$ , где  $\tau_c$  – время распространения сигнала в канале связи,  $\tau_{\max}$  – время передачи кадра максимальной длины. С учётом того, что  $\tau_k < \tau_{\max}$  и  $\tau_c > 0$ , положим

$$\Delta t = 3\tau_k. \text{ Тогда коэффициент полезной загрузки канала: } \rho_{KC} = \frac{\tau_k}{\Delta t} = 0,33.$$

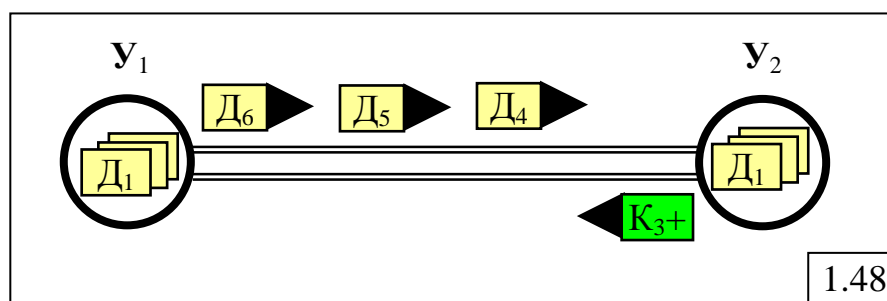
Следует иметь в виду, что этот результат справедлив при условии, что положительная квитанция поступила с первого раза. Если же первая передача кадра окажется неудачной, и в узел-отправитель поступит отрицательная квитанция, что потребует повторной передачи кадра, коэффициент полезной загрузки канала окажется гораздо ниже и составит:

$\rho_{КС} = \frac{\tau_k}{2\Delta t} = 0,165$ , то есть менее 20%. Это означает, что реальная скорость

передачи данных по каналу с пропускной способностью 10 Мбит/с будет составлять менее 2 Мбит/с. Очевидно, что такая ситуация является экономически неприемлемой, особенно для высокоскоростных каналов.

### 1.5.6.3. Скользящее окно

Для увеличения коэффициента полезной загрузки канала используется механизм «скользящего окна». Предварительно отметим, что если рассмотренные выше механизмы квитирования и тайм-аута предполагали наличие между взаимодействующими узлами полудуплексного канала, то механизм скользящего окна может быть реализован только при наличии дуплексного канала. При этом, кадры данных и квитанции могут передаваться одновременно по разным каналам (рис.1.48).



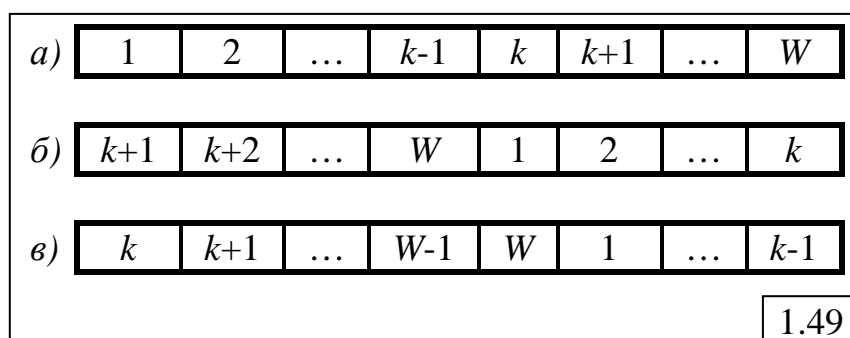
Суть механизма «скользящего окна» заключается в следующем. Узел-отправитель может послать подряд несколько кадров данных без получения на эти кадры квитанций. При этом кадры циклически нумеруются от 1 до  $W$ , где  $W$  – *размер (ширина) окна* – максимальное количество кадров, которые могут быть переданы без подтверждения. Номер кадра указывается в заголовке. Ширина окна может быть выбрана из условия максимальной загрузки прямого канала связи от узла-отправителя к узлу-получателю, которая может быть достигнута за счёт передачи ещё нескольких кадров за время ожидания квитанции на первый кадр:

$$W > \frac{T}{\tau_k} = \frac{2(\tau_c + \tau_k) + \tau_o}{\tau_k} = 2 + \frac{2\tau_c + \tau_o}{\tau_k},$$

где  $T = 2(\tau_c + \tau_k) + \tau_o$  – минимальное время ожидания квитанции;  $\tau_k$  – время передачи кадра,  $\tau_c$  – время распространения сигнала по каналу связи,  $\tau_o$  – время, затрачиваемое в узле-получателе на обработку кадра и формирование квитанции.

Как следует из представленного выражения, если пренебречь временем распространения сигнала по каналу связи и временем обработки кадра в узле-получателе  $U_2$ , то минимальная ширина окна должна быть не менее 2.

Положим, что в начальный момент времени окно узла-отправителя  $U_1$  выглядит так, как это показано на рис.1.49,а), что означает возможность передачи  $W$  кадров без подтверждения. Для того чтобы простой канала связи свести к минимуму, квитанция в узле-получателе может быть сформирована раньше, чем закончится передача всех  $W$  кадров, то есть узел-получатель может отправить квитанцию узлу-отправителю в любой удобный для него момент времени. Такой момент обычно связан с формированием кадра данных, посылаемого по обратному каналу от узла  $U_2$  к узлу  $U_1$ . При этом в заголовок этого кадра вставляется квитанция, указывающая номер последнего кадра, который был принят без ошибок (положительная квитанция) или с ошибкой (отрицательная квитанция). Если квитанция на кадр с номером  $k$  ( $1 < k < W$ ) – положительная, то окно в узле  $U_1$  сдвигается так, как это показано на рис.1.49,б), что означает возможность передачи ещё  $W$  кадров с номерами  $k+1, \dots, W, 1, \dots, k$  без квитанции. Если квитанция на кадр с номером  $k$  ( $1 < k < W$ ) – отрицательная, то это означает, что кадры с номерами до  $(k-1)$  приняты правильно, а кадры, начиная с номера  $k$ , должны быть переданы повторно. При этом окно в узле  $U_1$  сдвигается так, как это показано на рис.1.49,в) что означает возможность передачи ещё  $W$  кадров с номерами  $k, \dots, W, 1, \dots, k-1$  без квитанции. Таким образом, квитанция может формироваться не на все передаваемые кадры, а только на некоторые из них, причём, если положительная квитанция пришла на кадр с номером  $k$ , то считается, что этот кадр и все предыдущие кадры с номерами от 1 до  $(k-1)$  приняты без ошибок. Аналогично, отрицательная квитанция на кадр с номером  $k$  означает, что все предыдущие кадры приняты без ошибок, и повторной передаче подлежат все ранее переданные кадры, начиная с номера  $k$ .



### 1.5.7. Управление трафиком на высших уровнях OSI-модели

На сетевом уровне управление потоком в сети передачи данных реализуется за счет:

- применения различных методов маршрутизации;
- установления приоритетов между различными типами трафика.

На транспортном уровне управление потоком между конечными узлами сети реализуется за счет:

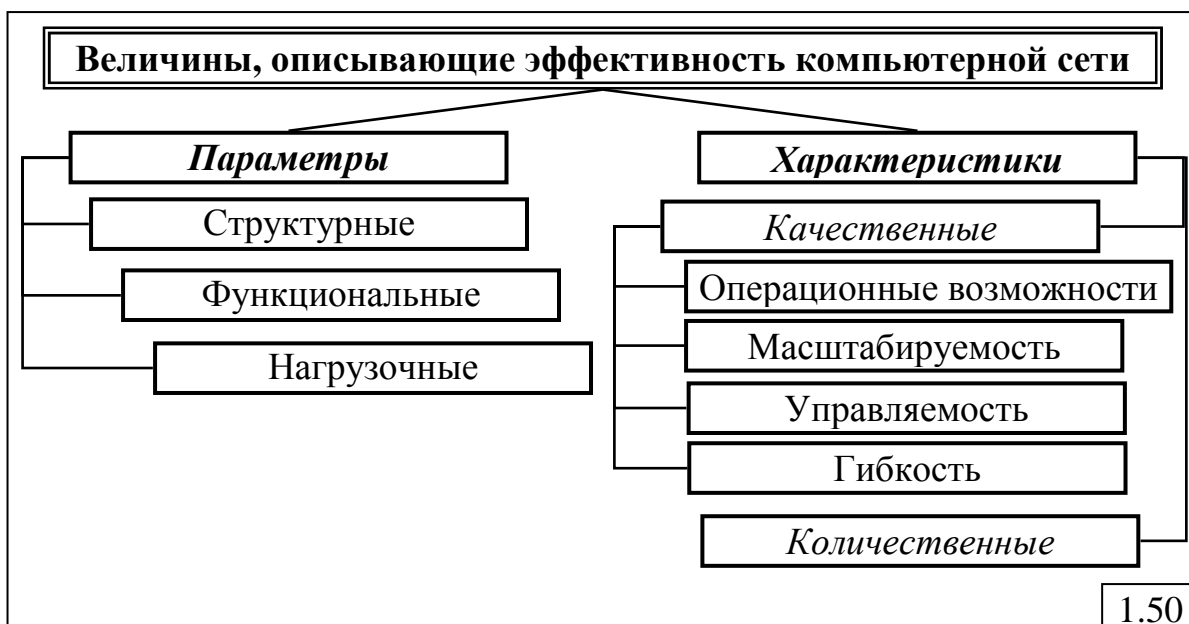
- установления приоритетов между различными типами трафика;
- ограничения поступающего от абонента трафика (например, когда скорость работы отправителя выше скорости получателя);
- ограничения доступа в сеть передачи данных.

**На сеансовом уровне** управление в коммутируемых сетях сеансом связи реализуется за счет применения различных способов установления соединения между абонентами.

### 1.6. Параметры и характеристики компьютерных сетей

Эффективность компьютерной сети может быть охарактеризована совокупностью величин, которые можно разделить на два класса (рис.1.50):

- параметры;
- характеристики.



**Параметры** компьютерной сети представляют собой величины, описывающие структурно-функциональную организацию сети и ее взаимодействие с внешней средой, в том числе, создаваемую в сети нагрузку.

**Характеристики** компьютерной сети описывают её эффективность и зависят от параметров.

Характеристики определяются в процессе эксплуатации сети путем измерений с помощью специальных измерительных средств – *сетевых мониторов* и в процессе решения задач системного анализа как функции параметров, т.е. являются *вторичными* по отношению к параметрам.

#### 1.6.1. Параметры компьютерных сетей

Все параметры компьютерной сети можно разделить на три группы (см. рис.1.50):

- 1) **структурные параметры**, описывающие состав и структуру сети;

2) **функциональные параметры**, описывающие стратегию управления передачей данных в компьютерной сети и стратегию управления обработкой данных в узлах;

3) **нагрузочные параметры**, описывающие взаимодействие сети с внешней средой, то есть нагрузку, создаваемую в сети решаемыми прикладными задачами и передаваемыми в вычислительной сети данными.

В качестве *структурных* параметров компьютерных сетей используются:

- количество узлов, входящих в состав сети, и их взаимосвязь (топология сети);
- типы узлов, состав и количество оборудования (ЭВМ и сетевых устройств);
- технические данные устройств (производительность ВС и сетевых устройств – маршрутизаторов и коммутаторов, пропускные способности каналов связи и т.п.).

К *функциональным* параметрам компьютерных сетей относятся:

- способ коммутации;
- метод доступа к каналу связи;
- алгоритм выбора маршрута передачи данных в сети;
- распределение прикладных задач по узлам сети;
- режим функционирования ВС;
- последовательность выполнения прикладных задач в ВС;
- приоритеты задач и т.д.

В качестве *нагрузочных* параметров компьютерных сетей могут использоваться:

- число типов потоков данных (аудио, видео, компьютерные данные);
- интенсивности поступления сообщений (пакетов, кадров) разных типов в сеть или к отдельным ресурсам (узлам и каналам связи);
- длина передаваемых по сети блоков данных (пакетов, кадров);
- число типов прикладных задач;
- ресурсоемкость каждой задачи и т.д.

### 1.6.2. Характеристики компьютерных сетей

Характеристики компьютерных сетей – это совокупность показателей эффективности (качества) сети.

Характеристики компьютерных сетей можно разделить на две группы (см. рис.1.50):

- качественные;
- количественные.

Примерами *качественных* характеристик могут служить:

- **операционные возможности** сети, представляющие собой перечень услуг (сервисов) по передаче и обработке данных, предоставляемых пользователям сети, таких как передача данных между удаленными пользователями сети, доступ к удаленным файлам, доступ к



разнообразным вычислительным средствам, в том числе, к высокопроизводительным ВС, электронная почта, возможность передачи по сети разнообразных данных (речь, аудио, видео) и т.д.;

- **масштабируемость** – способность сети при ее наращивании (при увеличении ресурсов) линейно увеличивать свою производительность, которую можно оценить количественно через отношение прироста производительности системы к приросту ресурсов: чем ближе это отношение к единице, тем выше масштабируемость;

- **управляемость** – возможность администрирования с целью выявления и разрешения возникающих в сети проблем, а также планирования развития и модернизации сети;

- **гибкость** – сохранение качества функционирования сети при изменении её состава и конфигурации в результате выхода из строя оборудования или добавления новых устройств.

Количественные характеристики компьютерных сетей можно разделить на две группы (рис.1.51):

- **глобальные**, определяющие наиболее важные свойства сети как целостного объекта;

- **локальные**, определяющие свойства отдельных устройств или частей сети и позволяющие получить более детальное представление об эффективности сети.



К глобальным характеристикам относятся:

- характеристики производительности;
- характеристики оперативности;
- характеристики надёжности;
- стоимостные характеристики;
- прочие характеристики (энергопотребления, массогабаритные и т.п.).

### 1.6.2.1. Характеристики производительности

**Производительность компьютерной сети** – мера мощности сети, определяющая количество работы, выполняемой сетью в единицу времени.

Понятие производительности охватывает широкую номенклатуру показателей эффективности компьютерной сети, определяющих качество

функционирования как сети в целом, так и отдельных ее подсистем и элементов – технических и программных средств.

Производительность сети зависит, в первую очередь, от производительности отдельных ее элементов, называемой *скоростью* работы или *быстродействием* устройств, например, скорость передачи данных по каналам связи, измеряемая объемом данных, передаваемых за единицу времени, быстродействие ЭВМ или, точнее, процессора, измеряемое числом команд, выполняемых в единицу времени, и т.п.

Для оценки производительности компьютерной сети в целом используется следующая совокупность показателей:

- **производительность СТК** (сети передачи данных), измеряемая числом сообщений (пакетов, кадров, бит) передаваемых по сети за единицу времени;

- **производительность СВТ** (средств обработки данных), представляющая собой суммарную производительность всех средств ВТ (ЭВМ и систем), входящих в состав сети.

*Производительность СТК (коммуникационная мощность)* может быть задана следующими показателями:

- *максимальная* или *предельная производительность*, называемая **пропускной способностью сети передачи данных** и измеряемая количеством пакетов (кадров), передаваемых в сети за единицу времени;

- *реальная* или *фактическая производительность* сети передачи данных, которая может быть задана как среднее значение на некотором интервале времени или как мгновенное значение в конкретный момент времени.

*Производительность СВТ (вычислительная мощность)* в целом складывается из производительностей ВС, выполняющих обработку данных в сети.

Наиболее важным показателем производительности ВС, как совокупности технических и программных средств, является **системная производительность**  $\lambda_0$ , измеряемая числом задач, выполняемых системой за единицу времени:

$$\lambda_0 = \lim_{T \rightarrow \infty} \frac{m(T)}{T},$$

где  $m(T)$  - число задач, выполненных за время  $T$ .

Очевидно, что системная производительность зависит от режима функционирования, реализуемого управляющими программами операционной системы, и класса решаемых задач, т.е. вычислительной нагрузки.

#### **1.6.2.2. Характеристики оперативности**

Характеристики оперативности описывают задержки, возникающие при передаче и обработке данных в сети.

Для оценки оперативности сети в целом используются следующие показатели:

- время доставки пакетов (сообщений);
- время отклика (ответа).

**Время доставки (время задержки)** пакетов характеризует эффективность организации передачи данных в вычислительной сети и представляет собой интервал времени, измеряемый от момента поступления пакета или сообщения в сеть до момента получения пакета адресатом.

В общем случае, время задержки – величина случайная, что обусловлено случайным характером процессов поступления и передачи данных в сети. В компьютерных сетях обычно время доставки задаётся средним значением  $T$ , на которое может налагаться ограничение  $T < T^*$  в зависимости от типа передаваемых данных.

При передаче мультимедийных данных кроме среднего значения времени доставки пакетов важной характеристикой является *вариация* или *джиттер задержки*, представляющая собой среднеквадратическое отклонение времени задержки разных пакетов.

**Время отклика (ответа)** – интервал времени от момента поступления запроса (сообщения, транзакции) в сеть до момента завершения его обслуживания, связанного с выполнением некоторой прикладной или обслуживающей программы, с обращением к базе данных и т.п. Время ответа представляет собой *время пребывания запроса в сети* и характеризует эффективность как телекоммуникационных, так и вычислительных средств компьютерной сети.

Время отклика, как и время задержки, – величина случайная и может задаваться средним значением  $U$  или в виде вероятности  $P(t_u < U^*)$  непревышения некоторого заданного значения  $U^*$ .

В сетях реального времени вместо термина "время ответа" часто используют термин "время реакции".

### **1.6.2.3. Характеристики надежности**

**Надежность** - способность компьютерной сети сохранять свои наиболее существенные свойства на заданном уровне и выполнять возложенные на нее функции в течение фиксированного промежутка времени при определенных условиях эксплуатации.

При рассмотрении вопросов надежности следует различать отказы и сбои.

**Отказ** – частичная или полная утрата работоспособности сети, приводящая к невыполнению или неправильному выполнению возложенных на нее функций. Для восстановления работоспособности системы при отказе требуется проведение ремонта.

**Сбой** – кратковременная утрата работоспособности сети, характеризуемая возникновением ошибки при передаче и обработке данных. Для восстановления работоспособности сети при сбое требуется проведение повторных действий по передаче (обработке) данных или части данных или перезагрузки отдельных узлов или всей сети. Сбои не

приводят к выходу сети из строя, однако могут существенно снизить эффективность функционирования, что проявляется в ухудшении характеристик функционирования сети (увеличивается время доставки сообщений и снижается производительность сети).

В качестве характеристик надежности обычно используются следующие показатели:

- **вероятность безотказной работы** сети  $P(t)$  – вероятность того, что в течение времени  $t$  не произойдет отказа;

- **интенсивность отказов**  $\lambda_o$  – среднее число отказов за единицу времени;

- **время наработки на отказ** – промежуток времени между двумя смежными отказами – величина случайная, а ее среднее значение  $T_o$  называется *средней наработкой на отказ*  $T_o = 1/\lambda_o$ ;

- **время восстановления** – интервал времени от момента наступления отказа до момента восстановления работоспособности системы – величина случайная и обычно задается средним значением  $T_b$ , называемым средним временем восстановления;

- **коэффициент готовности**  $K_r$  – доля времени, в течение которого сеть работоспособна:  $K_r = T_o/(T_o+T_b)$ .

Величина  $K_r$  может трактоваться как вероятность того, что в любой момент времени сеть работоспособна. Аналогично, значение  $(1-K_r)$  определяет вероятность того, что сеть находится в состоянии восстановления (неработоспособна).

#### 1.6.2.4. Стоимостные характеристики

В качестве стоимостных (экономических) характеристик компьютерной сети могут использоваться следующие показатели:

- **полная стоимость владения** (*Total cost of ownership, TCO*) – затраты, рассчитываемые на всех этапах жизненного цикла сети и включающие стоимость технических, информационных и программных средств (прямые затраты) и затраты на эксплуатацию сети (косвенные затраты);

- **стоимость (цена) передачи данных и обработки данных** в сети, определяемая объемом и стоимостью используемых ресурсов сети соответственно при передаче и обработке данных.

#### 1.6.2.5. Локальные характеристики СВ

В качестве локальных характеристик компьютерных сетей могут использоваться *в зависимости от целей исследования* самые разнообразные показатели эффективности.

Локальные характеристики описывают эффективность функционирования:

- узлов и каналов связи;
- отдельных сегментов сети;
- узлов обработки данных: ВС и ее подсистем.

Локальные характеристики могут быть разбиты на две группы:

- временные;
- безразмерные.

К временным характеристикам относятся:

- время доставки (задержки) пакетов при передаче между соседними узлами сети;
- время ожидания передачи данных в узлах сети или освобождения ресурсов ВС (сервера);
- время пребывания данных в различных узлах, устройствах или подсистемах.

К безразмерным характеристикам относятся:

- число пакетов, находящихся в буферной памяти узлов (маршрутизаторов, коммутаторов);
- коэффициенты загрузки узлов, каналов связи и устройств ВС и т.д.

**Коэффициент загрузки** или просто **загрузка**  $\rho$  устройства – это

доля времени, в течение которого устройство работает:  $\rho = \lim_{T \rightarrow \infty} \frac{t}{T}$ , где  $t$  –

время, в течение которого устройство работало;  $T$  – время наблюдения. Загрузка  $\rho$  характеризует степень использования устройства и часто называется *коэффициентом использования устройства*. Поскольку  $0 \leq \rho \leq 1$ , то загрузка может трактоваться как вероятность того, что в любой момент времени устройство работает. Величина  $\eta = 1 - \rho$  называется **коэффициентом простоя** устройства и характеризует долю времени, в течение которого устройство не работает (простаивает).

## 1.7. Сетевые протоколы

В современном мире существует большое количество различных сетевых технологий, каждая из которых реализуется множеством протоколов.

Множество протоколов разных уровней одной сетевой технологии называется *стеком протоколов*.

В настоящее время существует большое количество сетевых технологий с соответствующими стеками протоколов, в том числе: TCP/IP, XNS, IPX, AppleTalk, DECnet, SNA и др. Рассмотрим кратко некоторые из них и попытаемся установить их соответствие разработанной OSI-модели.

### 1.7.1. TCP/IP

Стек протоколов TCP/IP (Transmission Control Protocol/Internet Protocol) разработан по заказу Министерства обороны США с целью обеспечения быстрого увеличения числа компьютеров с разными операционными системами в сети за счет стандартизации.

Содержит 4 уровня.

**Уровень 1 – сетевой интерфейс** – реализует функции физического и канального уровня в OSI-модели:

- управляет обменом данными между устройством и сетью;
- маршрутизирует данные между устройствами одной сети.

**Уровень 2 – межсетевой** – соответствует сетевому уровню в OSI-модели:

- управляет обменом данными между устройствами, находящимися в разных сетях (обеспечивает дейтаграммный сервис в терминах IEEE-модели);
- отвечает за функции сетевой адресации.

**Уровень 3 – транспортный** – соответствует транспортному уровню в OSI-модели: обеспечивает связь "end-to-end" между источником и приемником данных.

**Уровень 4 – прикладной** – соответствует высшим уровням (5-7) в OSI-модели и обеспечивает функции, необходимые пользовательским (прикладным) программам, например, удаленное подключение к машине, передача файлов и т.д.

### 1.7.2. XNS

Стек протоколов XNS (Xerox Network Services Internet Transport Protocol) разработан компанией Xerox для передачи данных по сетям Ethernet.

Содержит 5 уровней.

**Уровень 1 – среда передачи** – реализует функции физического и канального уровня в OSI-модели:

- управляет обменом данными между устройством и сетью;
- маршрутизирует данные между устройствами одной сети.

**Уровень 2 – межсетевой** – соответствует сетевому уровню в OSI-модели:

- управляет обменом данными между устройствами, находящимися в разных сетях (обеспечивает дейтаграммный сервис в терминах IEEE-модели);
- описывает способ прохождения данных через сеть.

**Уровень 3 – транспортный** – соответствует транспортному уровню в OSI-модели:

- обеспечивает связь "end-to-end" между источником и приемником данных.

**Уровень 4 – контрольный** – соответствует сессионному и представительному уровню в OSI-модели:

- управляет представлением данных;
- управляет контролем над ресурсами устройств.

**Уровень 5 – прикладной** – соответствует высшим уровням в OSI-модели:

- обеспечивает функции обработки данных для прикладных задач.

### 1.7.3. IPX

Протокол IPX (Internet Packet Exchange) описан компанией Novell как "сервис", который позволяет приложениям посылать и получать сообщения через сеть. Поддерживает большое многообразие топологий ЛВС и физических средств передачи данных.

Содержит, как и протокол XNS, 5 уровней и во многом повторяет XNS.

Отличие заключается только в том, что IPX имеет несколько добавочных функций, например, возможность передачи служебных сообщений.

Протокол IPX обеспечивает:

- высокую производительность файлового сервера в ЛВС;
- простоту администрирования в малых и средних сетях;
- может работать в больших сетях и сетях с неоднозначными маршрутами, в том числе с несколькими соединениями сервера для распределения нагрузки.

Протокол IPX не гарантирует доставки сообщения, т.е. IPX-пакет может быть потерян. Для обеспечения гарантированной доставки разработан протокол SPX (Sequenced Packet Exchange - последовательный обмен пакетами), обеспечивающий подтверждение успешного прохождения сообщения по сети. В большинстве случаев IPX и SPX реализуются как единый протокол (одной программой) IPX/SPX.

### 1.7.4. AppleTalk

Протокол AppleTalk (компании Apple Computer) предназначен для связи между компьютерами Macintosh и наиболее близок к OSI-модели - содержит 6 уровней, причем высший (представительный) уровень объединяет в себе функции прикладного и представительного уровней OSI-модели.

### 1.7.5. DECnet

Стек протоколов DECnet (Digital Equipment Corporation net) содержит 7 уровней (рис.1.52).

Уровень	Наименование
7	Прикладной (пользовательский)
6	Сетевые приложения
5	Контроль сессии
4	Коммуникации "конец связи"
3	Маршрутизационный
2	Канальный
1	Физический канал

1.52

Несмотря на разницу в терминологии, уровни DECnet очень похожи на уровни OSI-модели.

DECnet реализует концепцию сетевой архитектуры DNA (Digital Network Architecture), разработанную фирмой DEC, согласно которой разнородные вычислительные системы (ЭВМ разных классов), функционирующие под управлением различных операционных систем, могут быть объединены в территориально-распределенные информационно-вычислительные сети.

### 1.7.6. SNA

Протокол SNA (System Network Architecture) компании IBM предназначен для удаленной связи с большими компьютерами и содержит 7 уровней (рис.1.53).

Уровень	Наименование
7	Сервис транзакций
6	Представительный сервис
5	Контроль потока данных
4	Контроль передачи
3	Контроль маршрута
2	Контроль канала
1	Физический контроль

1.53

SNA основана на концепции главной (хост)-машины и обеспечивает доступ удаленных терминалов к мейнфреймам IBM.

Основной отличительной чертой SNA является наличие возможности доступа каждого терминала к любой прикладной программе главной ЭВМ. Системная сетевая архитектура реализована на базе виртуального телекоммуникационного метода доступа (Virtual Telecommunication Access Method - VTAM) в главной ЭВМ. VTAM управляет всеми линиями связи и терминалами, причем каждый терминал имеет доступ ко всем прикладным программам.

### 1.7.7. Сопоставление коммуникационных моделей и протоколов

Ниже в табл.1.2 представлены рассмотренные стеки протоколов и показано их соответствие рекомендованной Международной организацией по стандартизации OSI-модели.



Таблица 1.2

	<b>OSI</b>	<b>TCP/IP</b>	<b>XNS(IPX)</b>	<b>AppleTalk</b>	<b>DECnet</b>	<b>SNA</b>
7	Прикладной	Прикладной	Прикладной	Представления	Прикладной (пользовательский)	Сервис транзакций
6	Представления		Контрольный		Сетевые приложения	Представительский сервис
5	Сеансовый (сессионный)		Сессионный	Контроль сессии	Контроль потока данных	
4	Транспортный	Транспортный	Транспортный	Транспортный	Коммуникации "конеч-связи"	Контроль передачи
3	Сетевой	Межсетевой	Межсетевой	Сетевой	Маршрути-зационный	Контроль маршрута
2	Канальный (передачи данных)	Сетевой интерфейс	Канальный интерфейс	Канальный	Канальный	Контроль канала
1	Физический			Физический	Физический канал	Физический контроль