

О. И. КУТУЗОВ, Т. М. ТАТАРНИКОВА,  
В. В. ЦЕХАНОВСКИЙ

# ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ И СЕТИ

*Учебник*



САНКТ-ПЕТЕРБУРГ  
МОСКВА  
КРАСНОДАР  
2020

УДК 004.7  
ББК 32.971.35я723

**К 95**     **Кутузов О. И.** Инфокоммуникационные системы и сети : учебник для СПО / О. И. Кутузов, Т. М. Татарникова, В. В. Цехановский. — Санкт-Петербург : Лань, 2020. — 244 с. : ил. — Текст : непосредственный.

**ISBN 978-5-8114-5774-8**

Целью изучения дисциплины «Инфокоммуникационные системы и сети» является формирование у студентов комплексных теоретических и практических знаний, навыков и умений в области организации информационного взаимодействия в процессе различных видов деятельности при помощи инфокоммуникационных технологий.

В результате изучения дисциплины студент должен быть подготовлен к решению следующих задач: эффективное использование возможностей инфокоммуникационных технологий и систем для ориентации предприятий и организаций на предоставление услуг нового вида; развитие технологической и технической базы предприятий и организаций за счет внедрения современных инфокоммуникационных технологий и систем; предупреждение угроз, возникающих в процессе внедрения и использования инфокоммуникационных технологий.

Книга предназначена для подготовки студентов учреждений среднего специального образования по специальностям «Программирование в компьютерных системах», «Прикладная информатика», «Информационные системы и программирование» и т. д.

УДК 004.7  
ББК 32.971.35я723

**Рецензенты:**

*В. А. БОГАТЫРЕВ* — доктор технических наук, профессор факультета программной инженерии и компьютерной техники Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики, почетный работник науки и техники РФ;

*Е. П. ИСТОМИН* — доктор технических наук, профессор, зав. кафедрой прикладной информатики Российского государственного гидрометеорологического университета.

**Обложка**  
*Ю. В. ГРИГОРЬЕВА*

© Издательство «Лань», 2020  
© Коллектив авторов, 2020  
© Издательство «Лань»,  
художественное оформление, 2020

## ОГЛАВЛЕНИЕ

Список сокращений.....	6
ПРЕДИСЛОВИЕ .....	7
1. ПРИНЦИПЫ ПОСТРОЕНИЯ ИНФОКОММУНИКАЦИОННЫХ СЕТЕЙ.....	9
1.1. Концептуальная модель инфокоммуникационной сети.....	9
1.2. Многоуровневый подход к построению архитектуры сети.....	11
1.3. Уровневая организация ЭМ ВОС .....	13
1.4. Структуризация сетей .....	17
1.4.1. Физическая структуризация сетей .....	17
1.4.2. Логическая структуризация сетей .....	20
1.5. Классификация сетей .....	29
1.6. Технология «клиент – сервер».....	30
1.7. Сетевые топологии .....	34
1.8. Характеристики инфокоммуникационных сетей.....	36
Контрольные вопросы .....	39
2. ПЕРЕДАЧА ДАННЫХ В СЕТИ.....	40
2.1. Элементы процессов передачи данных на физическом уровне .....	41
2.1.1. Кодирование источника.....	41
2.1.2. Понятие канала связи.....	43
2.1.3. Характеристики сигналов и каналов связи .....	45
2.1.4. Скорость передачи данных.....	46
2.1.5. Модуляция несущего колебания.....	48
2.1.6. Цифровое кодирование .....	50
2.1.7. Синхронизация при передаче данных .....	52
2.2. Методы передачи на канальном уровне .....	53
2.2.1. Общая структура кадра.....	53
2.2.2. Обнаружение и исправление ошибок.....	56
2.2.3. Методы восстановления искаженных и потерянных кадров.....	57
2.2.4. Протокол канального уровня HDLC.....	59
2.2.5. Уровень передачи данных в Интернете .....	63
Контрольные вопросы .....	64
3. ПРОВОДНЫЕ ЛОКАЛЬНЫЕ СЕТИ.....	65
3.1. Предназначение локальной сети.....	65
3.2. Стандарты базовых локальных систем.....	66
3.3. Протокол LLC уровня управления логическим каналом.....	67
3.4. Архитектура и технологии построения сетей Ethernet.....	69
3.4.1. Ethernet. Стандарт IEEE 802.3 .....	69

3.4.2. Fast Ethernet как развитие классического Ethernet'a .....	74
3.4.3. Протокол Gigabit Ethernet .....	76
3.5. Стандарт Token Ring .....	77
3.6. Стандарт FDDI .....	79
3.7. Технология Fibre Channel .....	81
3.8. Виртуальные локальные сети .....	82
Контрольные вопросы .....	85
4. ГЛОБАЛЬНЫЕ СЕТИ .....	86
4.1. Функциональная модель глобальной сети .....	86
4.2. Архитектура и технологии построения систем TCP/IP .....	90
4.2.1. Концептуальная модель сети TCP/IP .....	90
4.2.2. Стек протоколов TCP/IP .....	93
4.2.2.1. Прикладной уровень .....	94
4.2.2.2. Транспортный уровень .....	94
4.2.2.3. Сетевой уровень .....	99
4.2.2.4. Уровень доступа (уровень сетевых интерфейсов) .....	109
Контрольные вопросы .....	111
5. ОБЪЕДИНЕНИЕ СЕТЕЙ .....	112
5.1. Устройства объединения сетей .....	112
5.2. Технологии межсетевого взаимодействия .....	112
5.3. Средства согласования протоколов на физическом уровне .....	115
5.4. Согласование протоколов канального уровня .....	116
5.5. Объединение сетей на сетевом уровне .....	117
5.6. Коммутации с использованием техники виртуальных каналов .....	119
5.7. Корпоративные сети .....	120
5.8. Транспортная сеть .....	123
5.8.1. Распределение группового канала .....	123
5.8.2. Первичные сети .....	126
Контрольные вопросы .....	131
6. БЕСПРОВОДНЫЕ ЛОКАЛЬНЫЕ СЕТИ. ИНТЕРНЕТ ВЕЩЕЙ .....	132
6.1. Топологии беспроводных локальных сетей .....	132
6.1.1. Стандарт IEEE 802.11 .....	132
6.1.2. Стандарт IEEE 802.16 .....	134
6.1.3. Стандарт IEEE 802.15 .....	135
6.2. Самоорганизующаяся беспроводная сеть .....	135
6.3. Сенсорные сети .....	138
6.3.1. Узлы беспроводной сенсорной сети .....	138

6.3.2. Способы взаимодействия узлов в сенсорной сети .....	140
6.3.3. Механизмы кластеризации беспроводных сенсорных сетей.....	142
6.3.4. Разрешение коллизий источников данных в кластере БСС.....	143
6.4. Интернет вещей.....	145
6.4.1. Архитектура интернета вещей .....	145
6.4.2. Идентификация в интернете вещей .....	147
6.4.3. Способы взаимодействия в сети интернета вещей.....	148
6.4.4. Облачные технологии в интернете вещей.....	149
6.4.5. Протоколы интернета вещей.....	150
Контрольные вопросы .....	156
7. СЕТЕВЫЕ СЛУЖБЫ.....	158
7.1. Качество обслуживания (службы QoS) .....	158
7.1.1. Требования разных типов приложений .....	159
7.1.2. Управление трафиком. Службы QoS.....	161
7.2. Службы трансляции имен интернета .....	170
7.2.1. Функции DNS .....	170
7.2.2. Иерархия службы имен.....	171
7.2.3. Общие принципы функционирования DNS.....	172
7.3. Электронная почта .....	174
7.3.1. Основные элементы службы электронной почты .....	174
7.3.2. Угрозы безопасности электронной почты .....	177
Контрольные вопросы.....	180
8. ПРАКТИКУМ.....	181
8.1. Исследование информационного канала.....	181
8.2. Исследование шинной ЛВС с методом доступа МДКН/ОК.....	189
8.3. Исследование кольцевой локальной вычислительной сети.....	192
8.4. Исследование транспортного соединения в глобальной сети.....	193
8.5. Сетевые утилиты.....	196
8.6. IP-адресация.....	202
8.7. Аутентификация, авторизация и учет.....	208
8.8. Маршрутизация в IP-сетях.....	212
8.9. Статистическое описание функциональной надежности сети.....	214
Заключение .....	218
Приложение 1. Технологии построения глобальных сетей.....	219
Приложение 2. Описание основных протоколов семейства TCP/IP .....	229
Приложение 3. Листинги имитационных моделей .....	231
СПИСОК ЛИТЕРАТУРЫ.....	241

## СПИСОК СОКРАЩЕНИЙ

АЛ – абонентская линия  
АИ – адрес источника  
АП – адрес приемника  
АС – абонентская система  
АТ – абонентские терминалы  
БС – базовая станция  
БСС – беспроводная сенсорная сеть  
БЛС, WLAN (Wireless Local Area Network) – беспроводная локальная сеть  
ВК – виртуальный канал  
ВЛС, VLAN (Virtual Local Area Network) – виртуальная локальная сеть  
ГВС, WAN (Wide Area Network) – глобальная вычислительная сеть  
ИКС – инфокоммуникационная сеть  
ИКТ – инфокоммуникационные технологии  
ИС – информационная сеть  
ИП – информационный процесс  
КАМ – квадратурная амплитудная модуляция  
КС – канал связи  
КЭ – коммутационный элемент  
ЛВС, LAN (Local Area Networks) – локальная вычислительная сеть  
МДКН – множественный доступ с контролем несущей  
МДКН/ОК – множественный доступ с контролем несущей и обнаружением конфликтов  
ОС – операционная система  
ПОО – поле обнаружения ошибок  
ПП – прикладной процесс  
РС – рабочая станция  
САД – сеть абонентского доступа  
СУ – сенсорное устройство  
УК – узел коммутации  
ЦКП – центр коммутации пакетов  
ЭМ ВОС – эталонная модель взаимодействия открытых систем  
DNS (Domain Name System) – система доменных имен  
IoT (Internet of Things) – интернет вещей  
ID (Identifier) – идентификатор  
VPN (Virtual Private Network) – виртуальная частная сеть

## ПРЕДИСЛОВИЕ

В начале текущего столетия в массовом обиходе появилась аббревиатура ИКТ – инфокоммуникационные технологии.

Понятие «инфокоммуникационные технологии» объединяет две составляющие: информационные технологии и телекоммуникационные технологии. Обе эти технологии обеспечивают создание информационных сетей различного назначения – частных, корпоративных, локальных, ведомственных, глобальных.

Если информационную технологию можно определить как совокупность методов и средств получения, обработки, представления информации, осуществляемых в интересах пользователей, то под телекоммуникационными технологиями понимают средства, создающие инфраструктуру, или, другими словами, системно-технический базис для той или иной прикладной деятельности. Это и глобальная телекоммуникационная сеть (транспортная среда, абонентский доступ), это и сетевое оборудование (локальные сети, маршрутизаторы, серверы). Все виды обеспечения (программное, информационное, организационное) подчас относят к инфокоммуникационной составляющей, которая реализуется в виде **инфокоммуникационных** сетей.

В рассматриваемой области появилось довольно много книг как учебно-го, так и справочного характера, по самым разным вопросам аппаратно-программной реализации конкретного оборудования, по построению, функционированию и эксплуатации инфокоммуникационных сетей в целом. Однако они не все одинаково доступны для широкого круга пользователей. С другой стороны – относительная легкость изменения функционального наполнения разного инфокоммуникационного оборудования (преимущественно за счет его перепрограммирования) привела к значительному возрастанию многообразия форм и способов воплощения сетевых технологий в разнообразном сочетании и с различным оборудованием, что затрудняет понимание отличительных особенностей, достоинств и недостатков конкретных реализаций.

В то же время основные принципы построения информационных сетей за последние годы мало изменились. Потому одной из задач данного учебника является систематизация и разъяснение этих принципов с демонстрацией их воплощения в конкретных сетевых технологиях.

Учебник предназначен для студентов бакалавриата, обучающихся по направлению 09.03.02 «Информационные системы и технологии».

Материал учебника содержит описание широкого перечня сетевых технологий в лаконичной справочной форме с выделением основных закономерностей и отличительных особенностей.

Книга включает в предисловие себя семь глав, заключение, контрольные вопросы к каждой главе, приложения, список рекомендованной литературы и раскрытие условных обозначений (аббревиатур).

В главе I рассматриваются общие принципы построения информационных систем и инфокоммуникационных сетей.

В главе II представлены основные понятия и процедуры, связанные с преобразованиями при передаче данных на физическом и канальном уровнях. Необходимость этих общих сведений обусловлена тем, что развертывание и эксплуатацию локальных сетей выполняют «компьютерщики», часто по-своему понимающие физические процессы передачи данных.

В главе III рассматриваются стандарты и технологии построения базовых локальных компьютерных сетей (Ethernet, Token Ring), а затем – их развитие Ethernet. 802.3, Fast Ethernet, Gigabit Ethernet, FDDI, сеть Fibre Channel. Рассмотрены способы построения виртуальных сетей.

В главе IV описывается архитектура и технологии сетей TCP/IP, поскольку стек протоколов TCP/IP доминирует при построении глобальных сетей, в частности интернета. В Приложении 1 представлены технологии сетей X.25, ISDN, Frame Relay, ATM.

В главе V рассматриваются сложившиеся подходы, технологии и средства к объединению сетей; коммутация с использованием техники виртуальных каналов для организации межсетевых сцеплений; корпоративные сети, как пример применения инкапсуляции, и первичные сети, выполняющие в объединенных сетях роль транспортной сети. Глава содержит необходимые сведения по данной теме.

В главе VI представлен материал по беспроводным радиосетям, объемно и содержательно по архитектуре и технологиям сенсорных сетей и «Интернету вещей» как их новому развитию.

В главе VII в качестве примера предоставляемых инфокоммуникационной сетью услуг пользователю рассмотрены сетевые службы: качества обслуживания (Quality of Service – QoS) службы QoS, трансляции имен интернета, использующей систему доменных имен (Domain Name System, DNS), элементы электронной почты.

Контрольные вопросы соответствуют содержанию глав.

Девять практических работ, как и контрольные вопросы, предназначены для закрепления изучаемого материала.

Выполнение четырех работ основано на взаимодействии с Internet, пять работ представлены в виде имитационных моделей: звена передачи данных, сетей Ethernet, Token Ring, транспортного канала и функциональной надежности глобальной сети.

Имитационное моделирование в качестве метода исследования для практических занятий студентов не может в полной мере заменить сетевое оборудование. Тем не менее, рабочая имитационная модель близка к физической имитации, наглядно отражает процесс функционирования реальной системы. Имитационное моделирование, как наукоемкая технология, способствует реализации принципа сознательности и активности обучаемых. Активные методы предполагают использование проблемного обучения. Подходящей формой такого обучения и является имитационное моделирование. Оно способствует развитию у студентов навыка самостоятельного поиска и принятия решений.



# 1. ПРИНЦИПЫ ПОСТРОЕНИЯ ИНФОКОММУНИКАЦИОННЫХ СЕТЕЙ

Результатом слияния отраслей обработки и обмена информацией между удаленными в пространстве друг от друга компьютерами явилось появление информационных сетей, реализующих все множество процессов обработки и передачи информации.

Постепенно информационные сети стали обеспечивать обмен между абонентами не только данными, но и передачу телефонных сообщений, музыки, изображений, телевизионных передач. Трафик сетей стал интегрированным, мультимедийным. Такое объединение понятия информационных (компьютерных) и телекоммуникационных систем и сетей «породило» новый вид инфокоммуникационных систем и сетей. Однако принципы построения и понятия, характерные для информационных систем, остаются основой построения и для инфокоммуникационных систем и сетей.

## 1.1. Концептуальная модель инфокоммуникационной сети

*Информационная сеть* (ИС) – это сложная техническая система, состоящая из территориально распределенных информационных узлов (подсистем обработки информации) и каналов передачи информации, соединяющих данные узлы. В роли элементов инфокоммуникационных систем выступают информационные и прикладные процессы.

*Информационные процессы* (ИП) представляют собой совокупность взаимосвязанных процессов выявления, отбора, формирования информации, ее ввода в компьютерную систему, обработки, хранения и передачи.

*Прикладные процессы* (ПП) – это ИП в конечных системах сети, выполняющие обработку информации для конкретной услуги связи или приложения. Так, пользователь, организуя запрос на предоставление той или иной услуги, активизирует в своей конечной системе некоторый прикладной процесс.

Обобщенно функциональную архитектуру ИС можно представить в виде трехуровневой концептуальной модели (рис. 1.1):

- **первый уровень** (внутренний) описывает функции и правила взаимодействия при передаче различных видов информации между территориально удаленными абонентскими системами через физические каналы связи (передачи) и реализуется *транспортной сетью*;

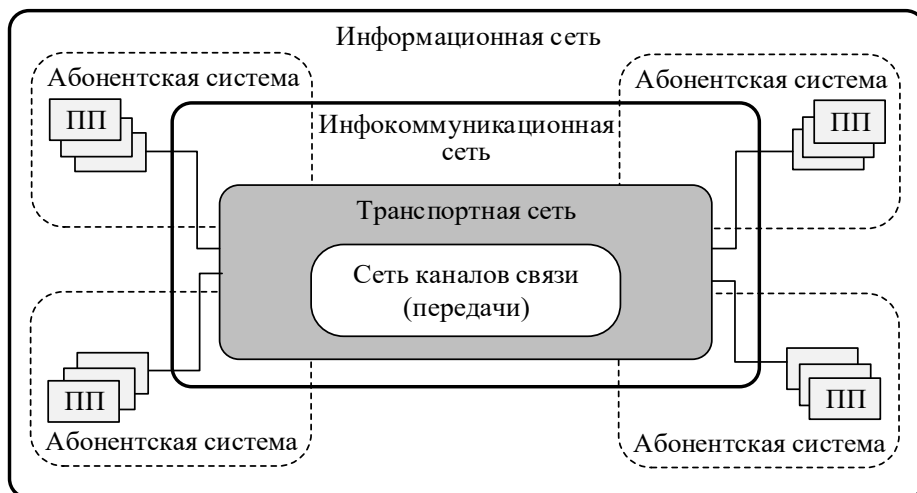
- **второй уровень** (промежуточный) описывает функции и правила обмена информацией в интересах взаимосвязи прикладных процессов различных абонентских систем и реализуется *инфокоммуникационной сетью* (ИКС).

ИКС представляет собой единую инфраструктуру для обмена различными видами информации в интересах пользователей ИС;

- **третий уровень** (внешний) образуется совокупностью прикладных процессов, размещенных в территориально удаленных абонентских системах.

*Абонентские системы* (терминалы, локальные сети) являются потребителями информации и выполняют ее содержательную обработку. Третий уровень,

дополняя первый и второй указанными функциями обработки информации, образует внешний облик *инфокоммуникационной сети*.



**Рис. 1.1. Концептуальная модель информационной сети**

Кроме того, в реальности, между транспортной сетью и абонентскими системами (АС) могут существовать достаточно большие расстояния, преодоление которых является функцией *сетей абонентского доступа* (САД). САД можно рассматривать как дополнение к транспортной сети или как ее составную часть (в последнем случае в составе транспортной сети различают магистральную сеть и САД).

Ядром представленной модели является ИКС, обеспечивающая взаимодействие удаленных процессов.

*Процесс* – это динамический объект, реализующий целенаправленный акт обработки данных. В информационных сетях объектом может быть прикладной процесс, пользователь, клиент, сервер, функциональный блок (устройство либо программа, выполняющая определенную часть задачи), операционная система, абонентская система и т. д.

Процесс, как любой динамический объект, протекает во времени и состоит из этапов инициализации, выполнения и завершения. При этом процесс может порождаться пользователем, системой или другим процессом. Ввод и вывод данных, необходимых процессу, производится в форме сообщений.

*Сообщение* – последовательность данных, имеющих законченное смысловое значение.

Взаимодействие удаленных процессов сводится к обмену сообщениями. Промежуток времени, в течение которого взаимодействуют процессы, называется *сеансом (сессией)*.

## 1.2. Многоуровневый подход к построению архитектуры сети

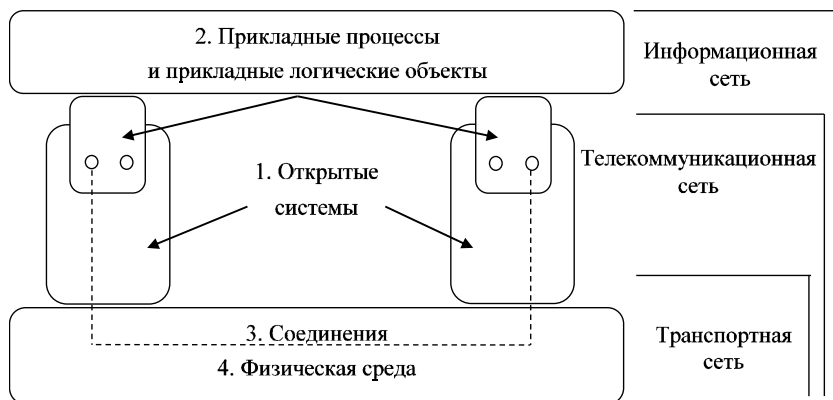
Взаимодействие между удаленными процессами может быть представлено в виде последовательности разнообразных и, как правило, многократных функциональных преобразований информационных сообщений в различных сетевых элементах из одной цифровой формы в другую и из одного вида физических (электрических) сигналов в другие.

Эта задача чрезвычайно сложна. Ее решение оказывается возможным лишь после предварительного разбиения требуемых функций на отдельные подфункции и создания их спецификаций и принципов их согласованного взаимодействия. Результатом такого разбиения стали открытые многоуровневые модели, описывающие функционирование сети.

Основу многоуровневого подхода к построению ИС составляют четыре элемента, представленные на рис. 1.2.

Понятие *открытости* систем означает взаимное признание и поддержку соответствующих стандартов взаимосвязи и не связано с их конкретной реализацией и используемыми техническими и программными средствами.

Для различных телекоммуникационных сетей, создававшихся в разное время различными производителями, группирование указанных функциональных преобразований неодинаково. Отличается также количество выделяемых этапов и функций процесса взаимосвязи, зачастую объединяемых в рамках той или иной функциональной архитектуры ИС (ИКС) в отдельные уровни или слои.



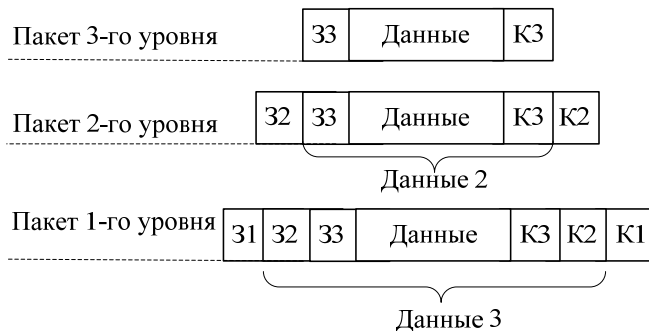
**Рис. 1.2. Основные элементы многоуровневой модели и их соотношение с архитектурой ИС**

Формальной процедуры выбора числа уровней не существует. Выбор производится эмпирическим путем на основе анализа различных вариантов архитектуры ИКС, опыта разработки и эксплуатации ранее созданных сетей.

За каждым уровнем закрепляется фиксированная часть функций и в соответствии с ними ему присваивается определенное название, отражающее его назначение. Уровни подстраховывают и проверяют работу друг друга.

Многоуровневая организация управления процессами в сети порождает необходимость модифицировать на каждом уровне передаваемые сообщения применительно к функциям, реализуемым на этом уровне.

Данные, передаваемые в форме сообщения, снабжаются *заголовком* (З) и *концевиком* (К) (рис. 1.3), в которых содержится информация, необходимая для обработки сообщения на соответствующем уровне: указатели типа сообщения, адреса отправителя, получателя, канала, порта и т. д.



**Рис. 1.3. Вложенность протокольных блоков данных различных уровней**

Средства управления нижнего уровня оперируют с данными, указанными в обрамлении, как с данными на конверте. При передаче на вышестоящий уровень сообщение «освобождается от конверта», в результате чего на следующем уровне обрабатывается очередной «конверт». Таким образом, каждый уровень управления оперирует не с самим сообщением, а только с «конвертами», в которые «упаковано» сообщение, поэтому содержание сообщения, формируемого на верхних уровнях, никак не влияет на функционирование нижних уровней управления передачей.

Гибкость организации и простота реализации сетей достигается, в частности, за счет того, что обмен сообщениями (данными) допускается только между процессами одного уровня. Это означает, что прикладной процесс может взаимодействовать только с прикладным процессом, а процессы управления передачей сообщений на уровнях 1, 2, ... – только с процессами одноименных уровней. Эта схема взаимодействия процессов, как и процедура обрамления сообщений, – необходимое условие логической независимости уровней организации сети.

Процедура взаимодействия процессов на основе обмена сообщениями (данными) определяется протоколом. *Протокол* – это набор правил и соглашений для передачи и приема сообщений между уровнями. Для взаимодействия процессов каждого уровня используются горизонтальные протоколы.

Поскольку процессы выполняются в различных территориально удаленных системах, им присущи взаимная неопределенность состояния, связанная с отсутствием у каждого из них полной информации о состоянии другого процесса; отсутствие однозначной зависимости между событиями и действиями, выполняемыми при их наступлении. По этой причине может немотивированно

измениться состояние любого из процессов: пользователь может прекратить работу, прикладная программа перейти в состояние ожидания или завершиться из-за особой ситуации, возникшей при ее выполнении, и т. п. Нет полной гарантии доставки сообщений. Сообщение может не достичь адресата, в результате чего процесс, пославший сообщение, может не получить необходимой ему ответной реакции. Эти факторы существенно увеличивают сложность протоколов.

При описании протокола принято выделять его логическую и процедурную характеристики.

*Логическая характеристика протокола* – это структура (формат) и содержание (семантика) сообщений; задается перечислением типов сообщений и их смысла.

Правила выполнения действий, предписанных протоколом взаимодействия, называются *процедурной характеристикой протокола*.

Таким образом, логика организации информационной сети в наибольшей степени определяется протоколами, устанавливающими как тип и структуру сообщений, так и процедуры их обработки – реакцию на входящие сообщения и генерацию собственных сообщений.

Число уровней управления и типы используемых протоколов определяют архитектуру сети.

В настоящее время существует ряд различных архитектур сетей, ставших «де-факто» или «де-юре» международными открытыми (общепринятыми) стандартами. Примером наиболее известной и детально проработанной архитектуры является семиуровневая эталонная модель взаимодействия открытых систем (ЭМ ВОС), предложенная Международной организацией стандартов (МОС/ISO – International Standards Organization).

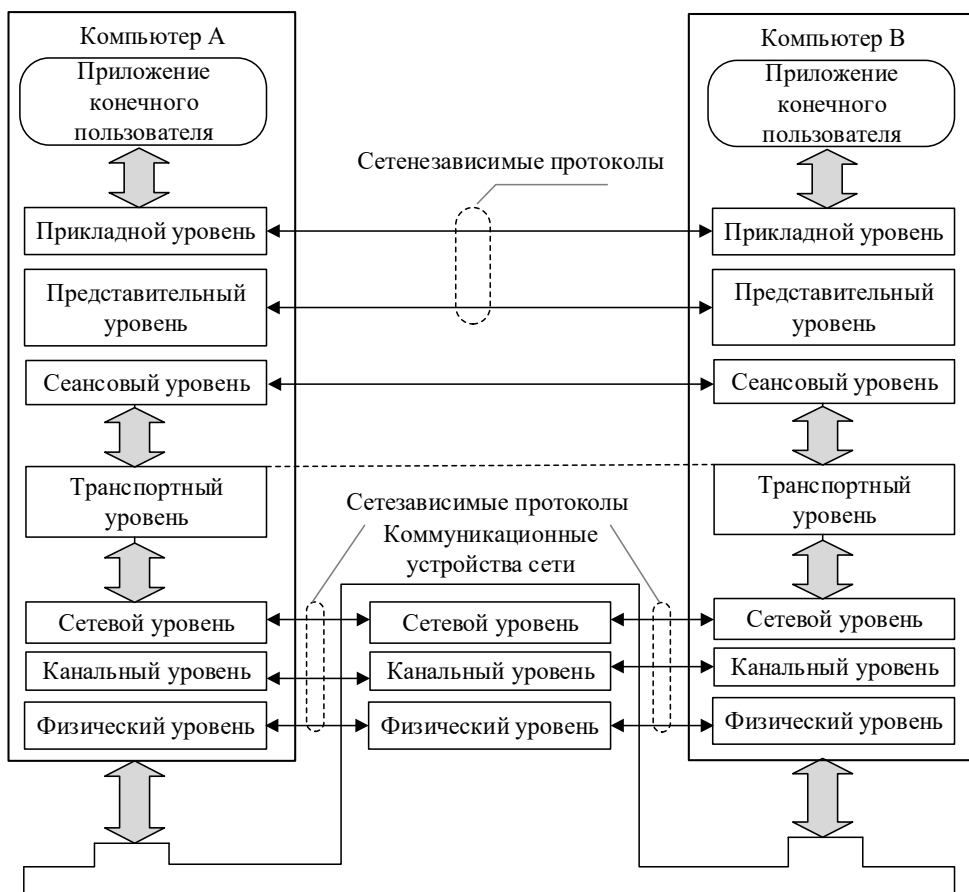
### 1.3. Уровневая организация ЭМ ВОС

Архитектура сети, представленная ЭМ ВОС, ориентирована на описание реализации только функций взаимодействия информационных процессов, выполняющих содержательную обработку информации в территориально распределенных узлах ИКС.

Под ВОС понимается совокупность взаимодействующих реальных открытых систем вместе с физической средой, предназначенной для передачи информации между ними. В качестве физической среды для ВОС обычно выступают цифровые каналы передачи различной физической природы (проводные, оптические и радио).

Данная модель, отражающая взаимодействие двух автономных систем, приведена на рис. 1.4.

Модуль уровня  $N$  физически взаимодействует только с модулями соседних уровней  $(N + 1)$  и  $(N - 1)$ . Модуль уровня 1 взаимодействует с передающей средой, которая может рассматриваться как объект уровня 0. Прикладные процессы принято относить к высшему уровню иерархии, в данном случае к уровню 7.



**Рис. 1.4. Семиуровневая ВОС систем (OSI)**

Физически связь между процессами обеспечивается передающей средой. Взаимодействие прикладных процессов с ней организуется с использованием шести промежуточных уровней управления 1–6, которые удобнее рассматривать с нижнего.

Такая многоуровневая организация обеспечивает независимость управления на уровне  $N$  от порядка функционирования нижних и верхних уровней. В частности, управление каналом (уровень 2) происходит независимо от физических аспектов функционирования канала связи, которые учитываются только на уровне 1. Управление сетью (уровень 3) базируется на использовании надежных каналов передачи данных и не зависит от способов, применяемых для обеспечения надежности на уровне 2.

Управление сетью реализует специфичные процессы передачи данных по сети, но транспортный уровень взаимодействует с сетью передачи данных как единой системой, обеспечивающей доставку сообщений абонентам сети.

В результате прикладной процесс создается только для выполнения определенной функции обработки данных без учета структуры сети, типа каналов связи, способа выбора маршрутов и т. д. Этим обеспечивается открытость и гибкость системы.

Сказанное проиллюстрировано рис. 1.4, на котором представлено взаимодействие приложений (процессов) конечных пользователей, реализуемых в двух различных системах (компьютерах) *A* и *B*. Процессы *A* и *B* опираются на службу взаимодействия, которая для них является целостной системой, наделенной необходимыми функциями. Взаимодействие между процессами организуется средствами прикладного, представительного уровней и уровня управления сеансами (уровень 5), которые работают на основе транспортного канала, обеспечивающего передачу сообщений в течение сеанса. Транспортный канал, создаваемый на уровне 4, включает в себя сеть передачи данных, организующую связи, т. е. требуемые каналы между любыми заданными абонентами сети.

Модель ISO/OSI определяет функции уровней следующим образом:

**Уровень 1 – физический.** Имеет дело с передачей битов по физическим каналам. К этому уровню имеют отношение *характеристики физических сред передачи данных*, на нем определяются *характеристики электрических сигналов*, стандартизируются *типы разъемов* и *назначение каждого контакта*.

Функции физического уровня реализуются во всех устройствах, подключенных к сети. Повторители и концентраторы – это те оборудование, которое работает только на физическом уровне. Со стороны компьютера функции физического уровня выполняются сетевым адаптером.

**Уровень 2 – канальный.** Обеспечивает проверку *доступности среды передачи*, реализацию механизмов *обнаружения и коррекции ошибок*. На канальном уровне биты группируются в наборы, называемые *кадрами (frame)*.

Протоколы канального уровня используются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются сетевыми адаптерами и их драйверами.

**Уровень 3 – сетевой.** Несет ответственность за доставку данных от узла-отправителя к узлу-получателю. На этом уровне осуществляется выбор маршрутов следования пакетов данных, частично решаются вопросы адресации, сопряжения сетей, а также управления скоростью передачи информации для предотвращения перегрузок в сети.

Сообщения сетевого уровня принято называть *пакетами (packet)*. Узлы сети соединяются между собой маршрутизаторами.

**Уровень 4 – транспортный.** Регламентирует передачу данных между *удаленными процессами*. Обеспечивает доставку информации вышележащим уровням с необходимой степенью надежности. Наряду с сетевым уровнем может управлять скоростью передачи данных и частично решать проблемы адресации.

**Уровень 5 – сеансовый.** Координирует (синхронизирует) взаимодействие связывающихся процессов. Средства синхронизации позволяют создавать контрольные точки при передаче больших объемов информации. В случае сбоя

в работе сети передачу данных можно возобновить с последней контрольной точки, а не начинать заново.

**Уровень 6 – представления данных.** Отвечает за форму представления данных, перекодирует текстовую и графическую информацию из одного формата в другой, обеспечивает ее сжатие и распаковку, шифрование и декодирование.

**Уровень 7 – прикладной.** Служит для организации интерфейса между пользователем и сетью. На этом уровне реализуются такие сервисы, как удаленная передача данных, удаленный терминальный доступ, почтовая служба и работа во всемирной паутине (web-браузеры).

Функции всех уровней модели OSI могут быть отнесены к одной из двух групп: зависящие и не зависящие от конкретной технической реализации сети.

Три нижних уровня – физический, канальный и сетевой – являются сетезависимыми. Протоколы этих уровней тесно связаны с технической реализацией сети и используемым коммуникационным оборудованием.

Три верхних уровня – прикладной, представительный и сеансовый – ориентированы на приложения и мало зависят от технических особенностей построения сети. На протоколы этих уровней не влияют какие бы то ни были изменения топологии сети, а также замена оборудования или переход на другую технологию сети.

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

Для реализации функций управления передачей данных используются и технические и программные средства. Как правило, уровни 1 и 2 реализуются, в основном, техническими средствами: на уровне 1 применяются электронные схемы; на уровне 2 – программируемые контроллеры; на уровнях 3–6 программные средства, образующие сетевое программное обеспечение компьютера.

Уровни управления 1 и 2 связываются между собой и с уровнем 3 посредством схемных интерфейсов – интерфейсных шин. Порядок взаимодействия между уровнями управления, реализуемыми с помощью программных средств, определяется программными интерфейсами.

Управление логическими объектами одинаковых уровней разных систем осуществляется в соответствии с горизонтальными протоколами. Различают следующие виды протоколов:

- **ориентированные и не ориентированные на соединение.** Первые устанавливают соединение между приложениями для передачи данных; вторые не устанавливают прямого сетевого соединения;

- **надежные и ненадежные.** Надежный протокол гарантирует доставку данных, ненадежный – нет;

- **потокосные и датаграммные.** Потокосный протокол рассматривает данные в качестве непрерывного последовательного потока; датаграммный – в качестве одиночных самостоятельных блоков.



Надо отметить, что к приведенной эталонной модели большинство практиков относится без излишнего пиетета. Эта модель не предвосхитила появления различных семейств протоколов, таких как, например, семейство протоколов TCP/IP, а наоборот, была создана под их влиянием. Ее не следует рассматривать как готовое решение для создания любого сетевого средства связи. Наличие некоторой функции на определенном уровне не гарантирует, что это ее наилучшее место; некоторые функции (например, коррекция ошибок) дублируются на нескольких уровнях, да и само деление на семь уровней носит отчасти произвольный характер. Хотя, в конце концов, были созданы работающие реализации этой модели, но наиболее распространенные семейства протоколов лишь до некоторой степени согласуются с ней. Ценность ЭМ ВОС заключается в том, что она показывает направление, в котором должны двигаться разработки новых инфокоммуникационных сетей.

## **1.4. Структуризация сетей**

В сетях с небольшим (10–30) количеством компьютеров используется одна из типовых топологий – общая шина, кольцо, звезда или полносвязная сеть. Все они обладают свойством однородности. Однородность структуры упрощает процедуру наращивания числа компьютеров, облегчает обслуживание и эксплуатацию сети.

При построении корпоративных сетей использование типовых структур порождает различные ограничения на:

- длину связи между узлами;
- количество узлов в сети;
- интенсивность трафика, порождаемого узлами сети.

Для снятия этих ограничений используются специальные методы структуризации сети и специальное структурообразующее оборудование: повторители, концентраторы, мосты, коммутаторы, маршрутизаторы, шлюзы. Оборудование такого рода называют коммутационным, имея в виду, что с помощью него отдельные сегменты сети взаимодействуют между собой.

Под физической структуризацией понимается конфигурация связей, образованных отдельными частями кабеля, а под логической – конфигурация информационных потоков между компьютерами сети. Физическая и логическая топологии могут совпадать, а могут и не совпадать.

### **1.4.1. Физическая структуризация сетей**

На уровне самого общего представления физическая структура информационной сети состоит из совокупности узлов и объединяющих их средств соединений.

*Физические средства соединений* – это совокупность физической среды, аппаратных и программных средств, обеспечивающая передачу сигналов меж-

ду узлами. Ее основой является используемая физическая среда: витая пара, коаксиальный и оптический кабели, эфир.

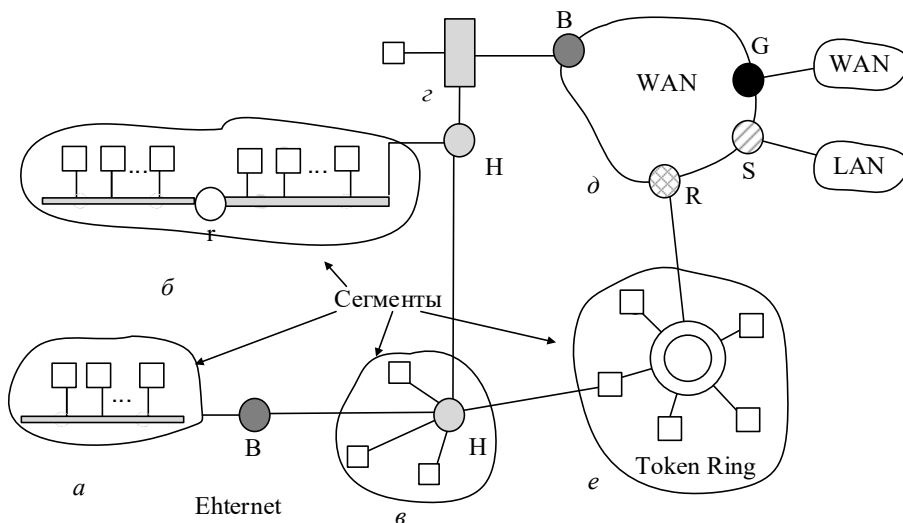
*Узел сети (node)* – это устройство, соединённое с другими устройствами как часть информационной сети. Узлы сети бывают двух типов: оконечный, расположенный в конце только одной ветви; промежуточный (ретрансляционный), расположенный на концах более чем одной ветви.

В оконечных узлах размещаются оконечные системы информационной сети: отдельные компьютеры, терминальные системы, абонентские пункты и системы, информационные банки и мэйнфреймы, которые являются поставщиками основных информационных и вычислительных ресурсов сети, узлы доступа (access nodes), серверы доступа.

Первое устройство, которое входит в состав аппаратного обеспечения, – сетевой адаптер (network adapter). Существуют и другие его названия, например, сетевая карта (network card), карта сетевого интерфейса (Network Interface Card, NIC). Работа сетевой карты заключается в физическом подключении компьютера к сети, для того чтобы компьютер за счет получения и передачи данных мог участвовать в сетевом взаимодействии. Сетевая карта должна соответствовать сетевой среде (network medium), к которой подключен компьютер.

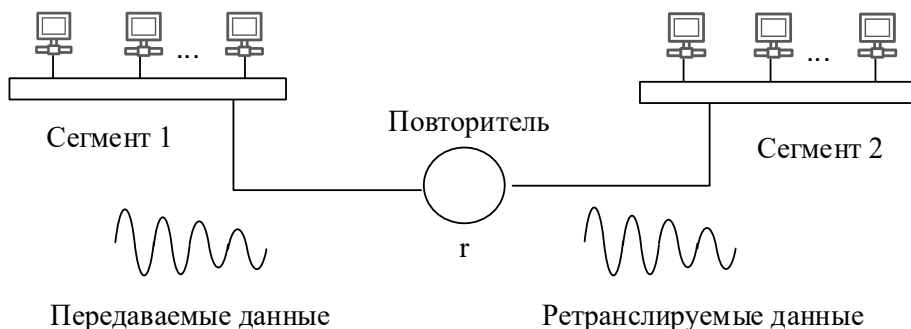
Применение ретрансляционных узлов обеспечивает сегментацию локальных сетей и интеграцию разнородных сетей в единую интерсеть, т. е. физическую структуризацию сети (рис. 1.5).

Простейшее из коммуникационных устройств – *повторитель (repeater)* – используется для физического соединения сегментов кабеля локальной сети с целью увеличения общей длины шины. Повторитель передает сигналы, приходящие из одного сегмента сети, в другие ее сегменты (рис. 1.5) и позволяет преодолеть ограничения на длину линий связи за счет улучшения качества передаваемого сигнала – восстановления его мощности, улучшения фронтов.



**Рис. 1.5. Физическая структура и элементы интерсети**

На физическом уровне пакет представляет собой счетную последовательность импульсов, распространяющихся по кабелю. За счет частичного отражения от точек подключения и поглощения в среде импульсы в пакете «расплываются» и искажаются (ухудшается отношение сигнал/шум), что является одной из причин ограничения длин кабельных сегментов. Для преодоления этих ограничений и вводятся сетевые повторители.



**Рис. 1.6. Усиление и ретрансляция сигналов повторителем**

Повторитель принимает сигналы из одного сегмента кабеля и побитно синхронно повторяет их в другом сегменте, улучшая форму и мощность импульсов, а также синхронизируя импульсы. Редактирование или анализ поступающих данных не производится. Задержка сигнала повторителем не должна превышать 7,5 тактов (750 нс для Ethernet). Все входы/выходы повторителя с точки зрения пакетов эквивалентны.

Многопортовый повторитель называют *концентратором* (hub). Концентраторы образуют из отдельных физических отрезков кабеля общую среду передачи данных – логический сегмент. В концентраторе пакет, пришедший по любому из входов, будет ретранслирован на каждый компьютер в сети. Проблема заключается в том, что любой компьютер может начать передачу в любое время.

Когда разные компьютеры, подключенные к концентратору, начинают передачу с небольшим сдвигом во времени, сигналы – переносчики пакетов, в общей физической среде «сталкиваются» и искажаются, возникает конфликт (коллизия – collision). Пакеты, участвующие в коллизии, уничтожаются. Компьютеры должны подождать произвольное количество времени и заново передать уничтоженные пакеты.

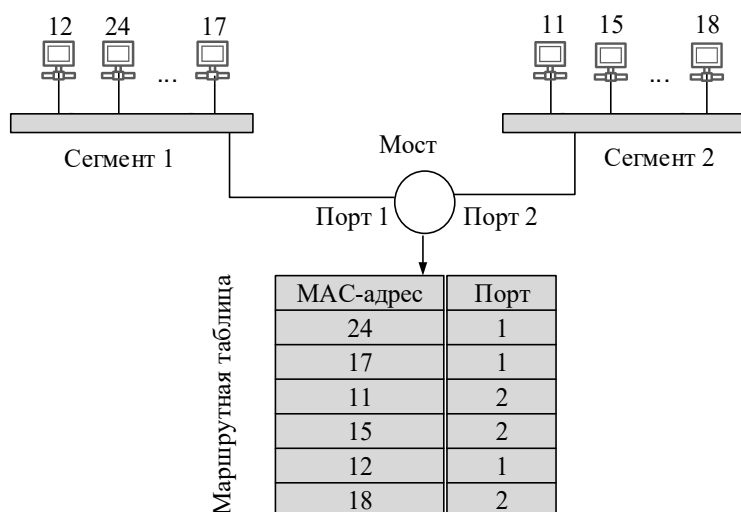
Повторители и концентраторы работают на физическом уровне.

По мере увеличения числа компьютеров, подключенных к концентратору, увеличивается и число столкновений, а по мере увеличения столкновений снижается эффективность сети. Именно по этой причине мосты и коммутаторы практически полностью вытеснили концентраторы.

## 1.4.2. Логическая структуризация сетей

Распространение трафика, предназначенного для компьютеров некоторого сегмента сети и только в пределах этого сегмента, называется локализацией трафика. Логическая структуризация сети – это процесс разбиения сети на сегменты с локализованным трафиком. Для этого используются мосты, коммутаторы, маршрутизаторы и шлюзы.

*Мост (bridge)* – это устройство, которое обеспечивает взаимосвязь двух (реже нескольких) локальных сетей (сегментов) посредством передачи кадров из одной сети (сегмента) в другую(ой) с помощью их промежуточной буферизации (рис. 1.7).



**Рис. 1.7. Мост как коммуникационное устройство канального уровня**

Мост по адресу источника составляет списки рабочих станций (PC) сети (сегмента) и формирует маршрутную таблицу. Она содержит для каждого MAC-адреса соответствующий номер порта моста, к которому подключены PC сети (сегмента). Мост выступает по отношению к каждой из сетей (сегментов) как конечный узел: принимает кадр, буферизует его, по полю адреса приемника проверяет список. Если приемник и источник находятся в одной сети (сегменте), мост такой кадр уничтожает, если адресуемый узел принадлежит другой сети (сегменту), передает его адресату.

Для передачи кадра в другую сеть мост должен получить доступ к ее разделяемой среде передачи данных в соответствии с теми же правилами, что и обычный узел (необычность в том, что мост не адресуем).

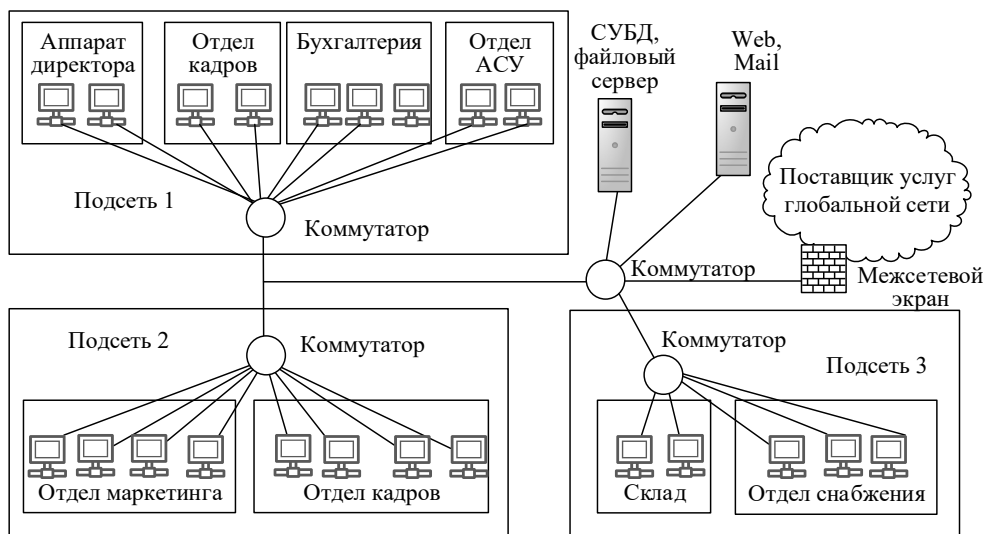
Мост изолирует трафик одного сегмента от трафика другого сегмента, фильтруя кадры. Коэффициент загрузки сегментов уменьшается.

Мосты обеспечивают выполнение функций канального уровня путем поддержки протоколов канального уровня (сети Ethernet, Token Ring, FDDI).

Мост не только снижает нагрузку в объединенной сети, но и уменьшает возможности несанкционированного доступа благодаря изолированности сегментов. Он может задерживать кадры и терять их. Задержка обусловлена записью в буфер и обработкой кадра (анализом адресов и их сопоставлением с таблицей), а также тем, что обработкой кадров управляет один процессор.

Производительность моста должна превышать среднюю интенсивность межсегментного трафика. Буферная емкость моста рассчитывается, исходя из пиковой нагрузки.

*Сетевой коммутатор (switch – переключатель)* – устройство, предназначенное для соединения нескольких узлов в пределах одного или нескольких сегментов сети и/или нескольких сетей. Коммутатор работает на канальном уровне модели OSI. Характерный пример применения коммутаторов для структуризации локальной вычислительной сети представлен на рис. 1.8.



**Рис. 1.8. Схема локальной вычислительной сети организации**

Коммутаторы были разработаны с использованием мостовых технологий. Разница между мостом и коммутатором состоит в том, что мост в каждый момент времени может осуществлять передачу кадров только между одной парой портов, а коммутатор одновременно поддерживает потоки данных между всеми своими портами. Другими словами, мост передает кадры последовательно, а коммутатор – параллельно.

Коммутатор – мультипроцессорная система. В каждом порту коммутатора свой процессор и буферная память для временного хранения пакетов.

В ассоциативной памяти коммутатор хранит таблицу коммутации, в которой указывается соответствие MAC-адреса узла порту коммутатора. При включении эта таблица пуста, и коммутатор работает в режиме обучения. В этом режиме поступающие на какой-либо порт данные передаются на все

остальные порты. При этом коммутатор анализирует кадры и, определив MAC-адрес хоста-отправителя, заносит его в таблицу на некоторое время.

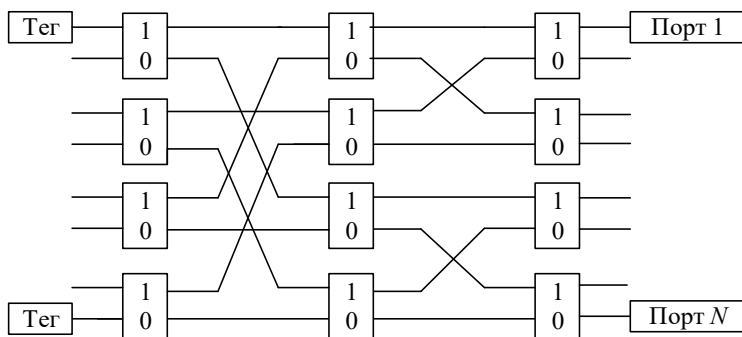
Впоследствии если на один из портов коммутатора поступит кадр, предназначенный для хоста, MAC<sup>1</sup>-адрес которого уже есть в таблице, то этот кадр будет передан только через порт, указанный в таблице. Если MAC-адрес хоста-получателя не ассоциирован с каким-либо портом коммутатора, то кадр будет отправлен на все порты за исключением того, с которого он был получен. Со временем коммутатор строит таблицу для всех активных MAC-адресов, в результате чего трафик локализуется.

Коммутаторы характеризует малая задержка и высокая скорость пере-сылки на каждом порту интерфейса.

Существуют три архитектурных решения аппаратной реализации коммутаторов, различающиеся способами комплексирования его функциональных модулей. Это коммутаторы на основе матрицы, общей шины и с общей памятью.

**Коммутаторы на основе матрицы.** Коммутатор матричного типа обеспечивает самый быстрый способ взаимодействия входных портов с выходными. Построение таких коммутаторов осуществляется на основе двоичных коммутационных элементов с двумя входами и двумя выходами.

Пример реализации коммутационной матрицы для восьми портов дан на рис. 1.9. Во входном порту по адресу назначения, записанному в служебной части информационного кадра, на основании просмотра адресной таблицы определяется номер выходного порта. Эта информация добавляется к байтам исходного кадра в виде специального ярлыка – тега (tag). Для данного примера тег представляет собой 3-разрядное двоичное число, соответствующее номеру выходного порта.

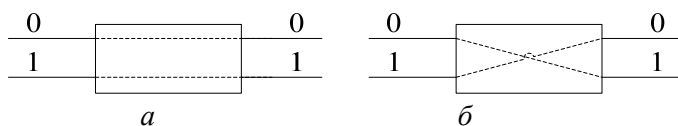


**Рис. 1.9. Вариант реализации коммутационной матрицы**

Матрица состоит из трех каскадов двоичных переключателей – коммутационных элементов, которые соединяют свой вход с одним из двух выходов в зависимости от значения бита тега.

Коммутационный элемент (КЭ) может работать в одном из двух режимов: «транзит» или «кросс» (рис. 1.10).

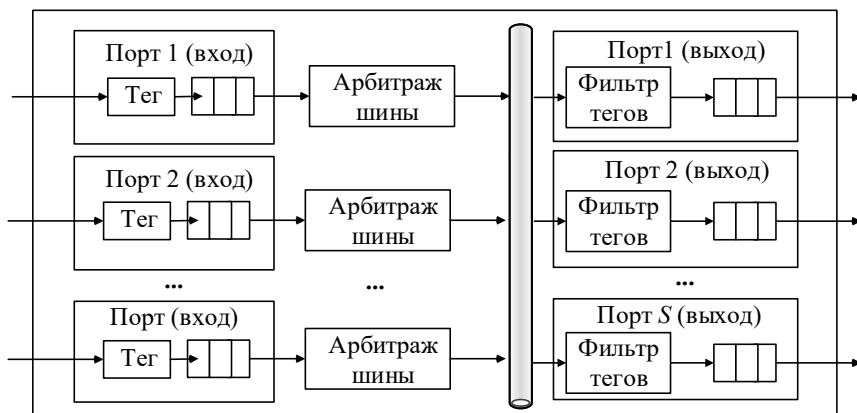
<sup>1</sup> MAC-адрес – уникальный идентификатор PC в локальной сети



**Рис. 1.10. Режимы работы коммутационного элемента:**  
а) «транзит» б) «кросс»

Переключатели первого каскада управляются первым битом тега, второго – вторым, а третьего – третьим. Известным недостатком этой технологии является отсутствие буферизации данных внутри коммутационной матрицы – если составной канал невозможно построить из-за занятости выходного порта или промежуточного КЭ, то данные должны накапливаться в буферных запоминающих устройствах порта коммутатора.

**Коммутаторы на базе общей шины.** Коммутаторы с общей шиной для связи входных портов с выходными применяют высокоскоростную шину, используемую в режиме разделения времени. В этой архитектуре шина (моноканал) пассивна, а активную роль выполняют специализированные процессоры портов. Пример такой архитектуры приведен на рис. 1.11.



**Рис. 1.11. Структура коммутатора на базе общей шины**

Кадр передается по шине в псевдопараллельном режиме небольшими частями (по несколько байт). Размер такой ячейки данных определяется производителем коммутатора.

Во входном порту формируется тег, в котором указывается номер порта назначения, и который добавляется к информационной ячейке, переносимой по шине. Каждый выходной порт содержит фильтр тегов, который выбирает только те теги, которые предназначены данному порту

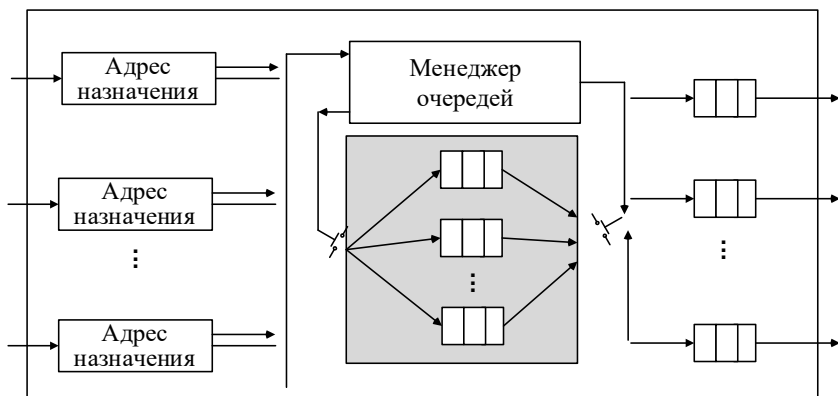
Шина не может осуществлять промежуточную буферизацию. Однако поскольку доступ портов к шине осуществляется циклически, задержка при доступе накапливается за счет промежутков ожидания между частями транспортируемого по шине пакета. Для того чтобы шина не была узким местом коммутатора, ее

производительность должна быть в несколько раз выше скорости поступления данных на входные порты.

**Коммутатор с общей разделяемой памятью.** В коммутационной схеме с общей разделяемой памятью входные и выходные порты коммутатора соединены между собой не через шину, а через общую память. Пример такой архитектуры приведен на рис. 1.12.

Входные порты (точнее специализированные процессоры этих портов) соединяются с переключаемым входом разделяемой памяти, а выходные – с переключаемым выходом этой памяти. Переключением входа и выхода разделяемой памяти управляет менеджер очередей. Он организует в разделяемой памяти несколько очередей данных, по одной для каждого выходного порта.

Входные порты передают менеджеру запросы на запись данных в очередь того порта, который соответствует адресу назначения пакета. Менеджер по очереди подключает вход памяти к одному из входных портов, и тот переписывает данные в очередь определенного выходного порта. По мере заполнения очередей менеджер производит также поочередное подключение выхода разделяемой памяти к выходным портам, и данные из очереди переписываются в выходной буфер соответствующего порта.



**Рис. 1.12. Структура коммутатора на базе общей памяти**

К недостаткам коммутаторов этого типа относят их сложность и высокую стоимость.

Функциональные возможности коммутаторов постоянно расширяются. Так, все большее распространение в сети приобретают коммутаторы Ethernet, поддерживающие функции маршрутизации. Они выполняют маршрутизацию на аппаратной базе. Однако сложные правила фильтрации и маршрутизации трафика остаются за магистральными маршрутизаторами.

Важной функцией современных корпоративных коммутаторов является *аппаратная классификация поступающего трафика* на третьем-четвертом уровне модели OSI. Классификация трафика на транспортном уровне позволяет интеллекту коммутаторов различать уже не только отдельные IP-пакеты с различными установленными классами обслуживания, но и различные типы вы-

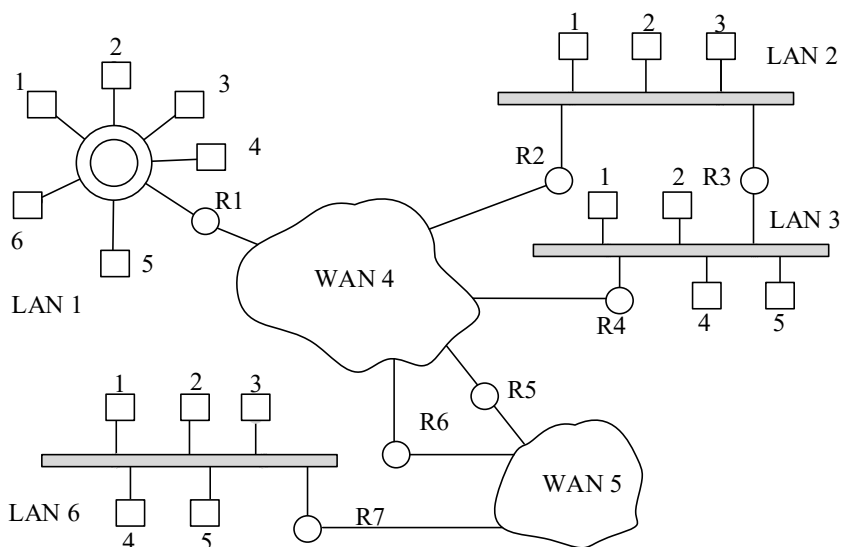


шестоящих протоколов (например, HTTP, FTP, SMTP) благодаря анализу заголовков TCP-пакетов.

Использование функций четвертого уровня обеспечивает качество обслуживания трафика критически важных приложений. Например, современные коммутаторы способны блокировать трафик потокового видео или аудио для обеспечения своевременной доставки электронной почты.

Итак, коммутаторы становятся основным типом сетевых устройств на первом, втором и третьем уровнях ЭМ ВОС, заменяя концентраторы, мосты и маршрутизаторы в локальных сетях.

**Маршрутизаторы (routers).** Маршрутизаторы образуют логические сегменты посредством явной адресации, поскольку используют не плоские аппаратные, а составные числовые адреса (рис. 1.13). В этих адресах имеется поле номера сети, так что все компьютеры, у которых значение этого поля одинаково, принадлежат к одному сегменту, называемому в данном случае *подсетью*.



**Рис. 1.13. Структура интерсети, построенной на основе маршрутизаторов:**  
**R1, R2, ..., R7 – маршрутизаторы; LAN1, LAN2, LAN3, WAN4, WAN5,**  
**LAN6 – уникальные номера сетей в едином формате;**  
**PC1, PC2, ... – локальные номера узлов (рабочих станций)**

Маршрутизатор позволяет организовывать в сети избыточные связи. Он «видит» всю картину связей подсетей друг с другом, на основе чего составляет маршрутную таблицу и может выбрать наиболее подходящий маршрут при наличии нескольких альтернативных.

Таблица маршрутизации в общем случае содержит следующие колонки:

- пункт назначения (Destination) – определяет IP-адрес сети назначения;
- маска сети (Subnet Mask) – задает количество лидирующих бит в IP-адресе, которые определяют адрес сети;

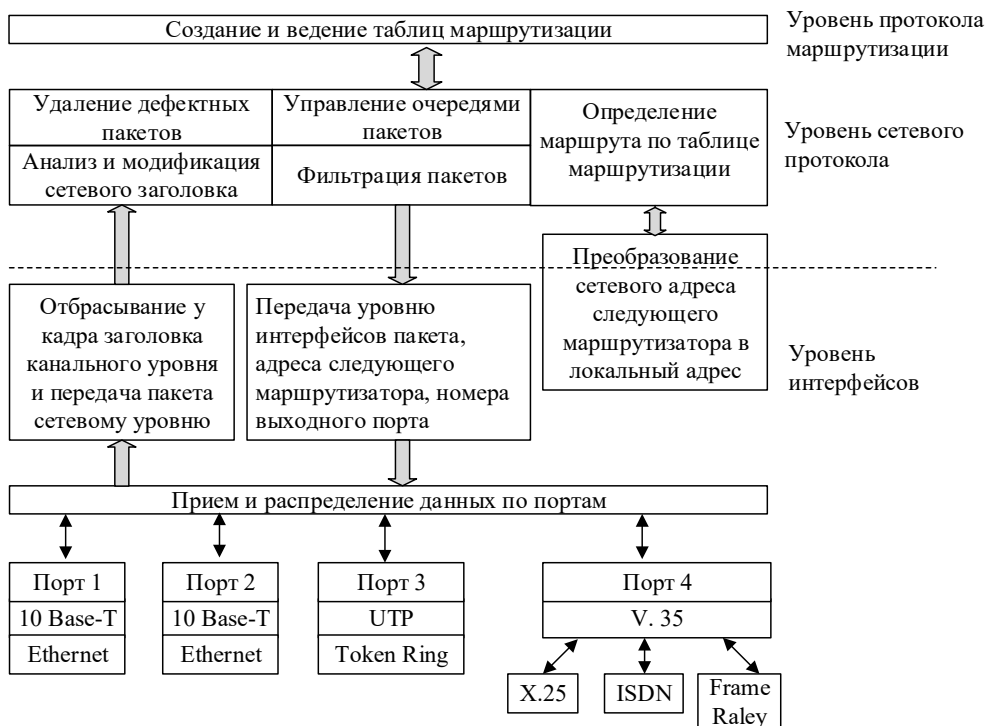
- пункт пересылки (Next Hop) – задает IP-адрес интерфейса следующего маршрутизатора, на который следует направить поступивший пакет;
- интерфейс (Interface) – задает собственный выходной порт маршрутизатора, на который следует направить поступивший пакет;
- метрика (metric) – задает предпочтение в выборе альтернативных маршрутов. Маршруты с меньшей метрикой более предпочтительны.

Маршрутизаторы не только объединяют сети, но и надежно защищают их друг от друга – маршрутизатор отказывается передавать «неправильный» пакет дальше, изолируя дефектный узел от остальной сети.

Маршрутизатор предоставляет администратору удобные средства фильтрации потока сообщений за счет того, что сам распознает многие поля служебной информации в пакете и позволяет их именовать понятным администратору образом.

Чтобы составить карту связей в сети, маршрутизаторы обмениваются специальными служебными сообщениями.

Функции маршрутизатора могут быть разбиты на три группы в соответствии с уровнями модели OSI: уровень интерфейсов, уровень сетевого протокола, уровень протокола маршрутизации (рис. 1.14).



**Рис. 1.14. Функциональная модель маршрутизатора**

**Уровень протоколов маршрутизации.** Построением и поддержанием таблицы маршрутизации занимаются *протоколы маршрутизации* (RIP, OSPF, ICMP). Они обеспечивают обмен между маршрутизаторами информацией о топологии сети, анализ этих сведений, определение наилучших маршрутов. Результаты заносятся в таблицы маршрутизации.

Различают три класса маршрутизаторов: магистральные, региональные, офисные. Как было отмечено ранее, все региональные и офисные маршрутизаторы практически заменяются быстродействующими управляемыми коммутаторами.

Шлюзы работают на самом высоком уровне стека протоколов и поддерживают взаимодействие систем и сетей, которые используют несовместимые протоколы. Шлюзы обеспечивают соединение и необходимые преобразования в терминах как аппаратуры, так и программного обеспечения.

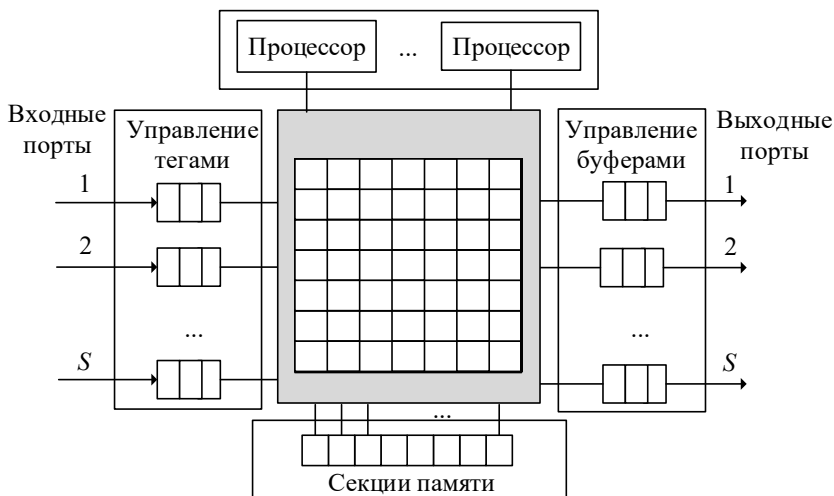
Сетевой *шлюз* – это точка сети, которая служит выходом в другую сеть. Например, сервер, контролирующий трафик между локальной сетью компании и сетью Интернет – это сетевой шлюз. Сетевой шлюз часто объединен с роутером, который управляет распределением и конвертацией пакетов в сети.

Основная задача маршрутизатора/шлюза – конвертировать протокол между сетями. Сетевой шлюз/маршрутизатор должен понимать все протоколы, используемые в объединенной сети.

В каждой из сетей, образующих интернет, сохраняются присущие им принципы адресации узлов и протоколы обмена информацией, поэтому маршрутизаторы могут объединять не только локальные сети с различной технологией, но и локальные сети с глобальными сетями. Эти особенности делают шлюз/маршрутизатор сложным интеллектуальным устройством, построенным на базе нескольких мощных процессоров (рис. 1.15). Такой специализированный мультипроцессор работает, как правило, под управлением специализированной операционной системы.

Когда пакет прибывает на маршрутизатор/шлюз, в порту отрезаются заголовки и концевики кадров и остаются только поля данных, которые и передаются в общее поле памяти шлюза/маршрутизатора. Далее анализируется заголовок пакета и в соответствии с записанным в нем заданием строится последовательный алгоритм (цепочка команд) обработки пакета протокольными процессами. В маршрутизаторе/шлюзе одновременно выполняется несколько заданий, так как протоколы могут иметь свои копии по уровням ЭМ ВОС и общая память разделена на секции. Это обеспечивает параллельную обработку пакетов в маршрутизаторе.

Сетевой шлюз может принять пакет, сформатированный под один протокол (например, Apple Talk) и конвертировать его в пакет другого протокола (например, TCP/IP) перед отправкой в другой сегмент сети (другую сеть). Сетевые шлюзы работают медленнее, чем сетевые коммутаторы.

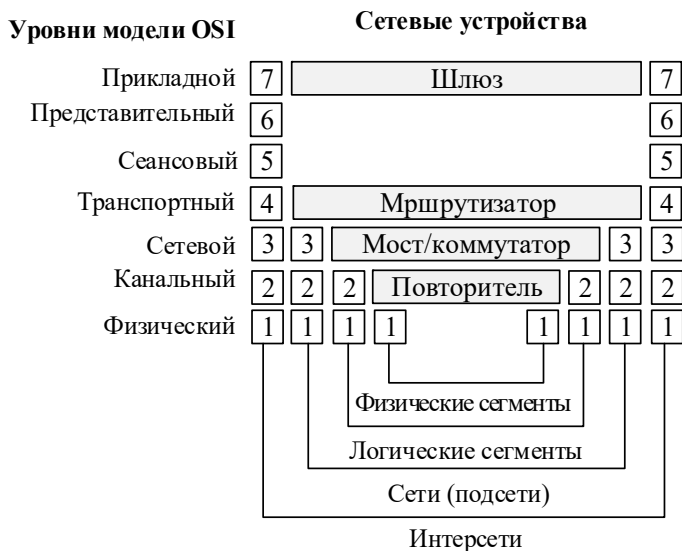


**Рис. 1.15. Архитектура шлюза/маршрутизатора**

На рис. 1.16 представлено распределение рассмотренных сетевых устройств по уровням ЭМ ВОС.

В крупных сетях сервер, работающий как сетевой шлюз, обычно интегрирован с прокси-сервером и межсетевым экраном.

Примерами межсистемных продуктов являются пакеты электронной почты. Они позволяют обмениваться почтовыми файлами пользователей в самых различных системах (домашних, офисных).



**Рис. 1.16. Соответствие функций коммуникационного оборудования модели OSI**

Сетевым оборудованием часто используется DHCP-протокол (Dynamic Host Configuration Protocol). DHCP – это протокол, который позволяет получить различные данные, необходимые клиенту для работы с протоколом IP. С использованием этого протокола добавление новых устройств и сетей становится простым и практически автоматическим.

## 1.5. Классификация сетей

Единой общепринятой системы, которой удовлетворяли бы все сети, не существует. Но есть два важнейших показателя: размеры и технологии передачи.

По территориальной распространенности, точнее – по охвату территорий, различают глобальные, локальные и региональные инфокоммуникационные сети.

Однако два главных термина в классификации сетей – это WAN и LAN.

*Wide Area Network (WAN)* – глобальные сети (рис. 1.17) покрывают большие регионы, в которые входят локальные и региональные сети. Глобальные сети охватывают территории государства или нескольких государств, к примеру, всемирная сеть Интернет.

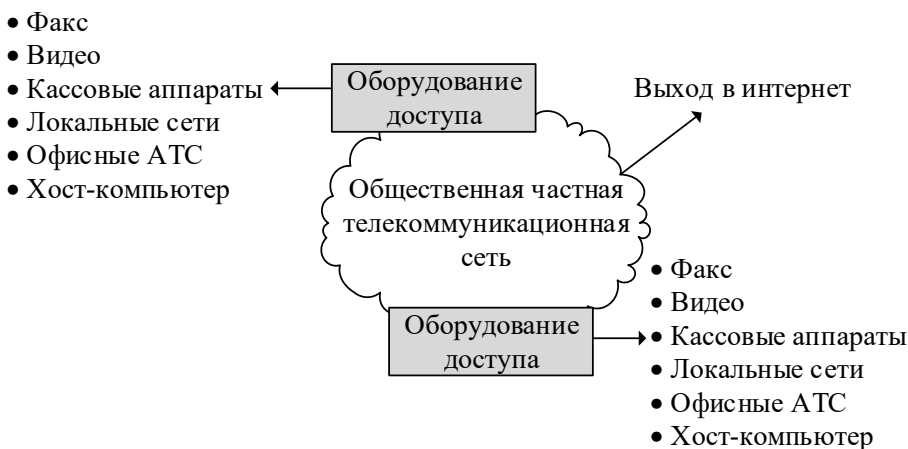


Рис. 1.17. Глобальная инфокоммуникационная сеть

*Local Area Networks (LAN)* – локальные сети (рис. 1.18), изначально предназначенные для офисов, в сочетании с коммутационными устройствами расширяют свои размеры вплоть до городских.

В отдельный класс выделяют промышленные сети, включающие на нижних уровнях сенсорные и контроллерные.

Многие организации создают собственные, так называемые *корпоративные сети*. Корпоративная сеть может объединять тысячи и десятки тысяч компьютеров, размещенных в различных странах и городах.



**Рис. 1.18. Структура локальной сети**

Существует два подхода решения проблемы предельной дальности.

**Первый подход.** За дальность отвечают внешние технологии передачи, которые предоставляют готовый цифровой канал. Такой подход дает возможность строить глобальные сети с неограниченной дальностью между сетевыми элементами.

**Второй подход** предполагает добавление к функциям физического уровня функций формирования и обработки сигналов для передачи через конкретную физическую среду на конкретные максимальные дальности. Такие сетевые технологии применяются для построения локальных сетей с ограниченной дальностью между сетевыми элементами.

Другая отличительная особенность LAN от WAN: способ разделения физической среды передачи между отдельными парами сетевых элементов при обеспечении передачи информации между ними.

Для WAN типично использование поделенной на отдельные каналы физической среды системами передачи. Принцип «точка-точка» действует в сетях с передачей от узла к узлу. Соединение любой конкретной пары PC осуществляется в соответствии с протоколами маршрутизации сетевого уровня, минимум через один узел коммутации, а максимум через все.

Для LAN рабочие станции связаны через общую физическую среду по принципу «точка-многоточка». Протоколы уровня звена данных регулируют поочередное использование общей среды передачи всеми PC.

Принцип «точка-многоточка» – «один передает – все принимают», но если информация предназначена кому-то одному, то только он ее и принимает, а остальные игнорируют.

Носителем адресных признаков PC источника и PC получателя в LAN являются специальные адресные поля кадров, которые не используются в WAN.

## **1.6. Технология «клиент – сервер»**

Цель создания сети – коллективное использование ресурсов (ресурсы – диски, файлы, принтеры, модемы, факс-аппараты и т. п.).

Специальные программные модули в режиме ожидания запросов называются *программными серверами* (*serve* – обслуживать).

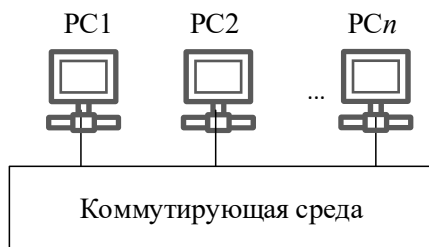
Специальные программные модули, вырабатывающие запросы, – *программные клиенты* (*client*).

Это части операционных систем. Пара клиент-сервер, обеспечивая доступ к определенному ресурсу, «образует» определенную службу: файловая служба, служба печати, служба электронной почты, служба удаленного доступа и др. (прикладной уровень ЭМ ВОО).

Термины «клиент», «сервер» используются и для обозначения соответствующих компьютеров.

Архитектура клиент-сервер может использоваться как в одноранговых локальных вычислительных сетях, так и в сети с выделенным сервером.

В **одноранговой сети** все компьютеры равноправны – имеют один ранг. Поэтому любой компьютер может выступать как в роли сервера, то есть предоставлять свои ресурсы (файлы, принтеры) другому компьютеру, так и в роли клиента – использовать предоставленные ему ресурсы других компьютеров – рабочих станций (рис. 1.19).



**Рис. 1.19. Одноранговая сеть**

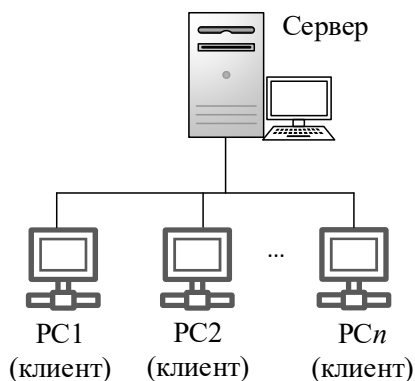
Достоинства одноранговых сетей в их относительной простоте и низкой стоимости. Недостаток – зависимость эффективности работы сети от количества рабочих станций.

В отличие от одноранговой существуют сети со специально выделенными компьютерами – серверами. Все остальные компьютеры сети – клиенты. Сервер предоставляет определенные услуги другим компьютерам. Существуют различные виды серверов (в зависимости от предоставляемых ими услуг): серверы баз данных, файловые серверы, серверы печати (принт-серверы), почтовые серверы, web-серверы, и т.д.

В программных средствах сетевых компьютеров выделяют три базовых компонента: *компонент представления* отвечает за пользовательский интерфейс; *прикладной компонент* реализует алгоритм решения конкретной задачи; *компонент управления* ресурсом обеспечивает доступ к необходимым ресурсам. В зависимости от того как эти компоненты распределены между клиентами и серверами, различают двухзвенные, трехзвенные и многозвенные архитектуры «клиент/сервер».

В рамках **двухзвенной архитектуры** (рис. 1.20) выделяют две основные модели: файл-сервер и сервер БД.

В модели **файл-сервер** представление данных, прикладной компонент и управление ресурсами – на стороне клиента, на стороне сервера – используемые файлы данных.



**Рис. 1.20. Типовая двухзвенная архитектура «клиент-сервер»**

В модели БД – представление данных и прикладной компонент – на стороне клиента, управление ресурсами и удаленные данные на стороне сервера.

Модель БД эффективнее, поскольку при запросе удаленных данных в модели файл-сервер считывается и перекачивается клиенту полный файл, а в модели БД – только выделенные запрашиваемые данные.

Третьим звеном в *трехзвенной архитектуре* становится сервер приложений, т.е. компоненты распределяются следующим образом: представление данных – на стороне клиента; прикладной компонент — на выделенном сервере приложений; управление ресурсами – на сервере БД, который и представляет запрашиваемые данные.

Трехзвенная архитектура может быть расширена до многозвенной путем выделения дополнительных серверов, каждый из которых будет представлять собственные сервисы и пользоваться услугами прочих серверов разного уровня для обслуживания требований клиентов.

В свою очередь, возможности клиентов характеризуются понятием «толщины».

Понятие «*тонкий клиент*» определяет такого клиента, у которого вычислительных ресурсов достаточно лишь для выполнения представительного компонента: на своем рабочем месте пользователь только вводит исходные данные и видит у себя на мониторе отображение результата.

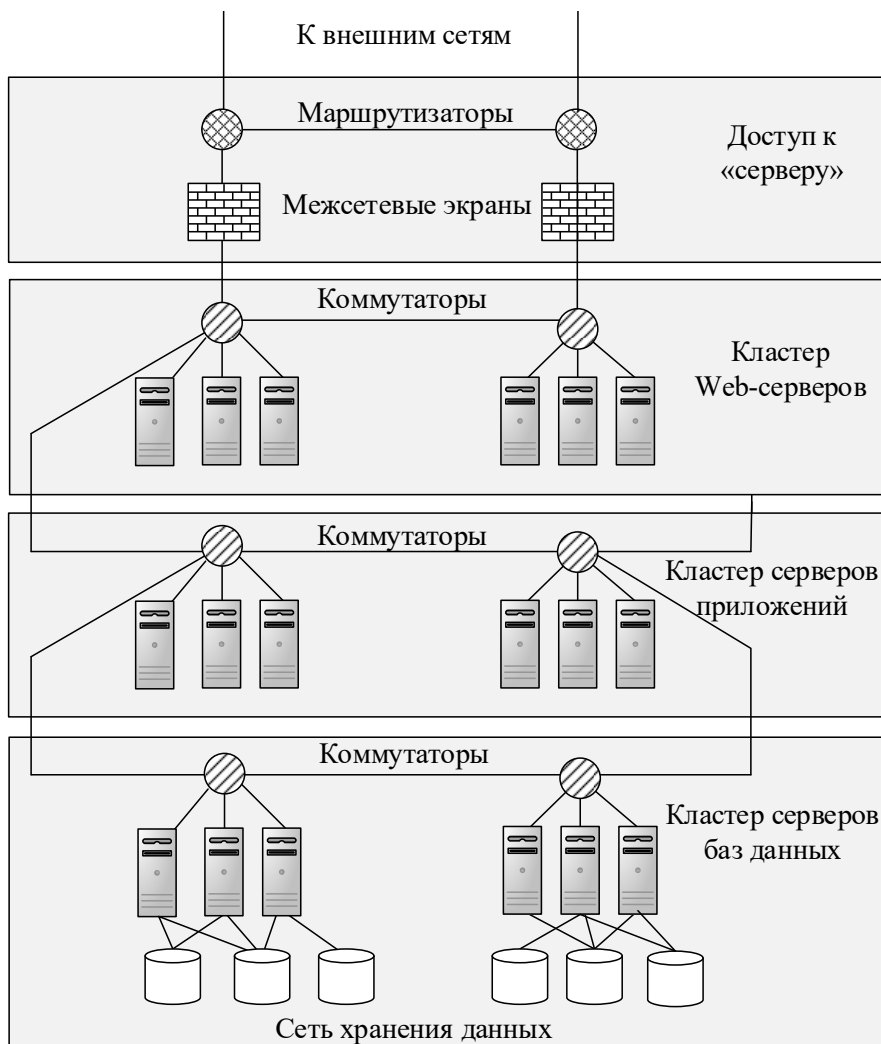
У «толстого клиента» все операции над данными проводятся непосредственно на рабочем месте пользователя, а сами данные хранятся на сервере, куда и обращается за ними клиентское приложение. Такой клиент может выполнять практически все функции и работать с прикладными типами данных.

Технологию «клиент-сервер» также называют облачными вычислениями (cloud computing), когда хотят подчеркнуть прозрачность технологии, то есть когда пользователь не «видит» сложности происходящих процессов, вычислений, расстояний.

*Облачные вычисления* – это модель обеспечения удобного сетевого доступа по требованию к некоторому общему фонду конфигурируемых вычислительных ресурсов, например к сетям передачи данных, серверам, системам хранения данных, приложениям и сервисам – как вместе, так и по отдельности.



«Облако» строится на основе центров обработки данных (ЦОД). Структура ЦОД состоит из серверов и систем хранения данных, объединенных локальной сетью (рис. 1.21).



**Рис. 1.21. Структурная модель ЦОД**

Серверы объединяются в вычислительные кластеры по назначению, например серверы приложений, систем управления базами данных, терминальные, web-серверы и др. Системы хранения данных предназначены для организации надежного хранения информационных ресурсов и предоставления доступа к ним серверов. Соединение кластеров между собой и с системами хранения данных внутри ЦОД реализуется коммутаторами. Соединение ЦОД с внешними сетями и удаленными клиентами реализуется маршрутизаторами. Межсетевые экраны обеспечивают информационную безопасность ЦОД.

## 1.7. Сетевые топологии

*Сетевая топология* – это граф, вершинам которого соответствуют конечные узлы сети (компьютеры, абонентские системы) и коммуникационное оборудование (коммутаторы, маршрутизаторы), а ребрам – связи (физические или информационные) между вершинами.

Выделяют типовые топологии сетей: шина, звезда, кольцо, ячеистая, со-  
товая.

**Шина** (рис. 1.22, а). Эту топологию часто называют линейной шиной. Она наиболее простая из всех топологий и весьма распространенная. В ней используется один кабель, называемый магистралью или сегментом, к которому подключены все компьютеры (РС).

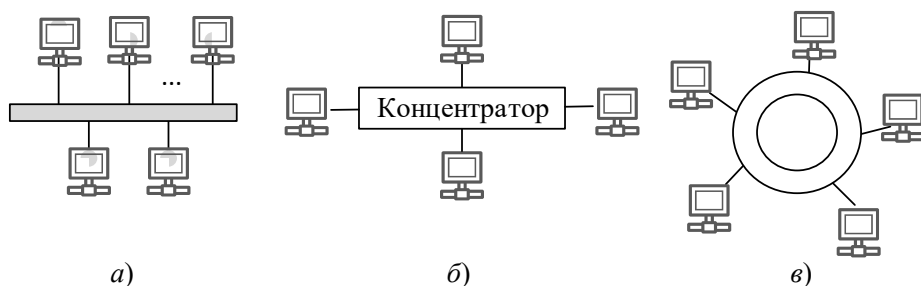


Рис. 1.22. Примеры сетевых топологий ЛС

Общая шина является очень распространенной топологией для локальных сетей. Применение общей шины снижает стоимость проводки, унифицирует подключение различных модулей, обеспечивает возможность широковещательного обращения ко всем станциям сети.

**Звезда.** В сети, построенной по топологии типа «звезда» (рис. 1.22, б), каждая РС подсоединяется кабелем (витой парой) к концентратору. Концентратор обеспечивает параллельное соединение РС и, таким образом, все компьютеры, подключенные к сети, могут общаться друг с другом.

Данные от передающей станции сети передаются через концентратор по всем линиям связи всем РС, но принимаются только теми станциями, которым она предназначена. Поскольку передача сигналов в топологии физическая звезда является широковещательной («точка-многоточка»), т. е. сигналы от РС распространяются одновременно во все направления, то логическая топология данной локальной сети является логической шиной.

Сеть с такой топологией устойчива к неисправностям отдельных РС и к разрывам соединения отдельных РС. Однако отказ центрального узла (концентратора) влияет на работу всей сети.

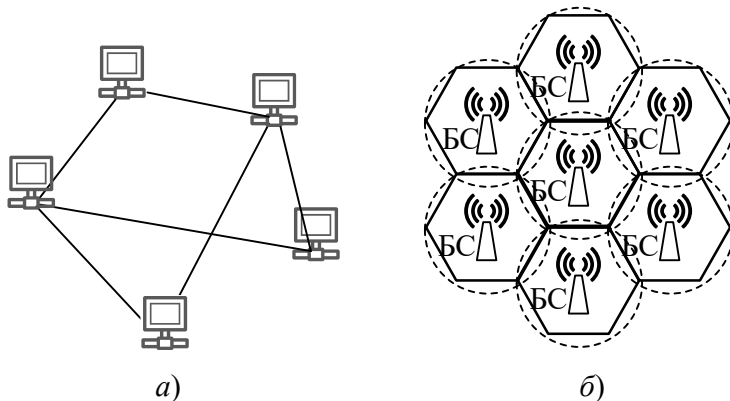
**Кольцо.** В сети с топологией типа «кольцо» (рис. 1.20, в) все узлы соединены каналами связи последовательно в неразрывное кольцо (не обязательно окружность), по которому передаются данные. Выход одной РС соединяется со входом другой РС. Начав движение из одной точки, данные, в конечном счете,

попадают на его начало. Данные в кольце всегда движутся в одном и том же направлении.

Принимающая РС распознает и получает только адресованное ей сообщение. В сети с топологией типа физическое кольцо используется маркерный доступ, который предоставляет станции право на использование кольца в определенном порядке. Логическая топология данной сети – логическое кольцо. Данную сеть очень легко создавать и настраивать.

Основной недостаток сетей топологии кольцо заключается в том, что повреждение линии связи в одном месте или отказ РС приводит к неработоспособности всей сети.

**Ячеистая топология.** Характеризуется наличием избыточных связей между устройствами. Получается из полносвязной топологии путем удаления некоторых связей (рис. 1.23, а). Полносвязная топология – это такое размещение узлов, при котором каждый узел соединен с каждым. Неоднозначность связей между узлами повышает надежность сетей с ячеистой топологией – обрыв одного соединения не нарушает функционирования сети в целом. Ячеистая топология характерна, как правило, для глобальных сетей.



**Рис. 1.23. Сетевые топологические схемы: а – ячеистая; б – сотовая**

**Сотовая топология.** Определяет принципы беспроводной связи для географических областей, разделенных на ячейки (соты). Каждая ячейка представляет собой часть общей области. Внутри таких сот функционируют конкретные соединения, связывающие устройства с базовой станцией (БС) (рис 1.23, б). Базовые станции соединены в виде сетки, что обеспечивает множество альтернативных маршрутов. Это позволяет поддерживать отказоустойчивость сети. Пример таких сетей – Wi-Fi и Wi-Max, сети сотовой связи.

Однако при построении крупных сетей однородная структура превращается из достоинства в недостаток. Поэтому в крупных сетях преобладает *смешанная топология* – топология с произвольными связями между элементами сети. В таких сетях можно выделить отдельные подсети, имеющие типовую топологию и не одну, поэтому их и называют сетями со смешанной топологией.

## 1.8. Характеристики инфокоммуникационных сетей

По целевому назначению к инфокоммуникационным сетям предъявляются три основных требования к доставке сообщений: по надежности, времени и верности. Надежность в значительной мере определяет основные характеристики сети. Различают структурную и функциональную надежность. Последнюю часто интерпретируют как живучесть сети.

При системном проектировании сети в качестве критериев используются вероятностно-временные характеристики (VBX) доставки сообщения, стоимость сети, структурная надежность либо живучесть сети. Причем стоимость сети выступает часто в виде ограничения.

Структурная надежность оценивается вероятностью безотказной работы сети, чаще ее отдельных узлов, средним временем безотказной работы или более полной характеристикой – функцией распределения вероятностей значений времени наработки на отказ.

В больших сетях с ячеистой топологией для обеспечения высокой надежности предусматривают  $k$ -связность между узлами (не менее  $k \geq 1$  независимых связей между любой парой узлов сети). Поэтому при отказе отдельных элементов сеть сохраняет свою функциональность, но снижаются ее временные характеристики. Передача по обходным путям увеличивает время доставки сообщений адресату, причем, как правило, на изменяющуюся случайную величину. Характеристики этой задержки – важные внешние характеристики сети, по которым в основном и судят о качестве предоставляемых услуг.

В качестве основного показателя своевременности широко используют среднее время доставки. Возможные случайные отклонения времени доставки (задержки) от их средних значений обычно оценивают дисперсией или вероятностью отклонения, превышающего (или не превышающего) определенную допустимую величину отклонений.

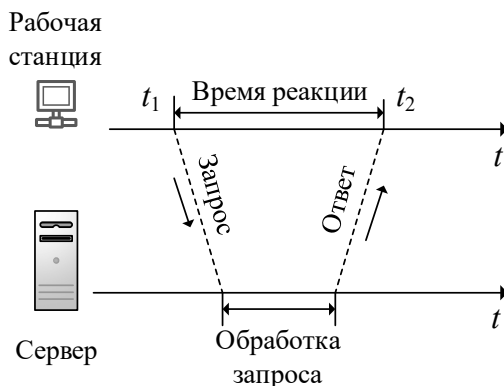
Следует различать *время доставки* сообщения и *время задержки* доставки сообщения, к которым могут предъявляться отдельные требования.

Если сообщение передается одним блоком данных, то эти показатели практически совпадают. Если же оно передается по частям в виде последовательности блоков данных, то время доставки сообщения соответствует времени доставки всех блоков, а время задержки – времени доставки одного блока.

Вследствие конвейерной обработки последовательно передаваемых блоков данных сразу на всех узлах вдоль маршрута их следования, время доставки всех блоков, как правило, заметно меньше произведения времени доставки одного блока на их количество.

Продолжительность обслуживания запросов пользователей характеризуют временем реакции на запрос (рис. 1.24). Смысл и значение этого показателя зависят от загруженности сегментов, через которые проходит запрос, загруженности сервера, коммутатора, моста, маршрутизатора и т.п.

Вариантами критерия могут служить времена реакции, измеренные при различных, но фиксированных состояниях сети (крайние значения): полностью ненагруженная сеть, нагруженная сеть.



**Рис. 1.24. Время реакции – интервал между запросом и ответом**

Важным показателем работоспособности сети, связанным с ее временными характеристиками, является пропускная способность сети.

Пропускная способность характеризует качество выполнения основной функции сети – транспортировки сообщений, и поэтому чаще используется при анализе производительности сети, чем время реакции.

Пропускная способность измеряется в пакетах (кадрах) в единицу времени (чаще в секунду). Пропускная способность может быть мгновенной, максимальной и средней.

При оценке пропускной способности сети в целом и по отношению отдельным узлам используются критерии двух типов: *средневзвешенные* и *пороговые*.

Средняя пропускная способность вычисляется путем деления общего объема переданных данных на время их передачи, причем выбирается достаточно длительный промежуток времени – час, день или неделя.

Мгновенная пропускная способность – для усреднения выбирается очень маленький промежуток времени – например, 10 мс или 1 с.

Максимальная пропускная способность – это наибольшая мгновенная пропускная способность, зафиксированная в течение периода наблюдения.

Наряду с показателями своевременности в состав показателей качества входят показатели достоверности, характеризующие различные искажения информации при передаче.

В телекоммутационных сетях протоколы нижних уровней контролируют наличие ошибок в принимаемых протокольных блоках данных (ПБД). При этом, если ошибки в принятых ПБД не удастся исправить, то они с некоторой вероятностью стираются. При стирании или повторяется передача ПБД, или сигнализируется об этом верхним уровням.

При идеальности подобных процедур обнаружения ошибок итоговое сообщение выдается прикладному процессу или абсолютно без ошибок, или не выдается совсем. Для оценки качества услуг при таком способе передачи данных понятие «достоверности» на верхних уровнях, в принципе, теряет смысл.

Неидеальность любых способов обнаружения и исправления ошибок приводит к тому, что существует определенная вероятность приема сообщений с необнаруженными ошибками.

Качество выполнения сетью основной задачи по предоставлению ряда услуг характеризуют совместимостью, управляемостью, защищенностью, расширяемостью, масштабируемостью. Кратко охарактеризуем их.

*Сохранность* данных (и их защита от искажений).

*Согласованность* данных (их непротиворечивость). Это требуется, например, когда несколько копий данных хранятся на разных файловых серверах.

*Безопасность* как способность системы защитить данные от несанкционированного доступа. Сюда относятся: защита каналов, защита компьютеров, защита от взлома паролей и т. д.

Специфичными для сетей являются показатели расширяемости и масштабируемости.

*Расширяемость* – это возможность легко добавлять в сеть новые элементы (пользователей, компьютеры, приложения, службы), наращивать длину сегментов и заменять аппаратуру более мощной.

*Масштабируемость* означает, что в сети возможно наращивание количества узлов и протяженности связей в очень широких пределах. При этом производительность сети не ухудшается.

Часто термины «расширяемость» и «масштабируемость» используются как синонимы. Однако если взять, к примеру, сеть Ethernet, то можно говорить о хорошей расширяемости (количество компьютеров на сегменте можно увеличить до 100), но при этом резко снижается производительность сети, что указывает на плохую масштабируемость.

*Прозрачность* сети достигается в том случае, когда для пользователя сеть представляется не как множество компьютеров, связанных сложной системой каналов, а как единая вычислительная машина с разделением времени.

Символом прозрачности считают лозунг компании Sun Microsystems «Сеть – это компьютер».

Прозрачность может достигаться на двух уровнях – пользователя и программиста. На уровне пользователя – для работы в сети используются те же команды и привычные процедуры, что и для работы с локальными ресурсами. На уровне программиста – приложению для доступа к удаленным ресурсам требуются те же вызовы, что и для локальных ресурсов.

Сеть должна скрывать различия операционных систем и компьютеров. От пользователя не требуется знание места расположения ресурса. Ресурсы должны свободно перемещаться с одного компьютера на другой без изменения их имен.

*Поддержка разных видов трафика.* Наряду с традиционным трафиком передачи данных все увеличивается доля мультимедийного трафика – передаваемых в цифровой форме речи и изображения.

Особенность мультимедийного трафика – жесткие требования к синхронизации передаваемых данных. При запаздывании сообщений в передаваемой последовательности будут наблюдаться искажения.

В сети в общем случае должны сосуществовать два вида трафика: традиционный компьютерный (пульсирующий) и мультимедийный (синхронный).

*Управляемость* – это возможность централизованно контролировать состояние основных элементов сети, выявлять и устранять неисправности, выполнять анализ производительности и планировать развитие сети. В этой области еще много нерешенных проблем. В основном существующие системы не управляют сетью, а лишь осуществляют наблюдение за ее работой.

*Совместимость* или *интегрируемость* означает, что сеть способна включать в себя разнообразное программное и аппаратное обеспечение. То есть сеть может быть неоднородной или гетерогенной. Основным путем обеспечения совместимости – это использование открытых стандартов и спецификаций.

### **Контрольные вопросы**

1. В чем отличие информационных процессов от прикладных?
2. В чем смысл многоуровневой организации эталонной модели ВОС?
3. Чем отличается транспортный уровень от сетевого? Физический от канального?
4. Какие уровни ЭМ ВОС зависят от технической реализации сети, а какие не зависят? И почему?
5. Сетевой протокол и интерфейс. В чем их различие и в чем сходство?
6. На каком уровне OSI работает концентратор? Мост? Коммутатор?
7. Отличается ли таблица коммутации от таблицы маршрутизации?
8. Какие уровни выделяют в функциональной модели маршрутизатора?
9. Когда вместо маршрутизатора применяется шлюз?
10. Какое наиболее значимое отличие локальных сетей от глобальных?
11. В чем смысл технологии клиент-сервер?
12. Перечислите модели клиент-серверного взаимодействия.
13. Виртуальная сеть – понятие физическое или логическое?
14. Как оценивается производительность сети?
15. Чем отличается структурная надежность сети от функциональной?

## 2. ПЕРЕДАЧА ДАННЫХ В СЕТИ

Передача данных в сети включает логические и физические процессы на физическом и канальном уровнях ЭМ ВОС.

Физический уровень – это первый (нижний) уровень эталонной модели, который отвечает за непосредственную передачу данных и связь между устройствами.

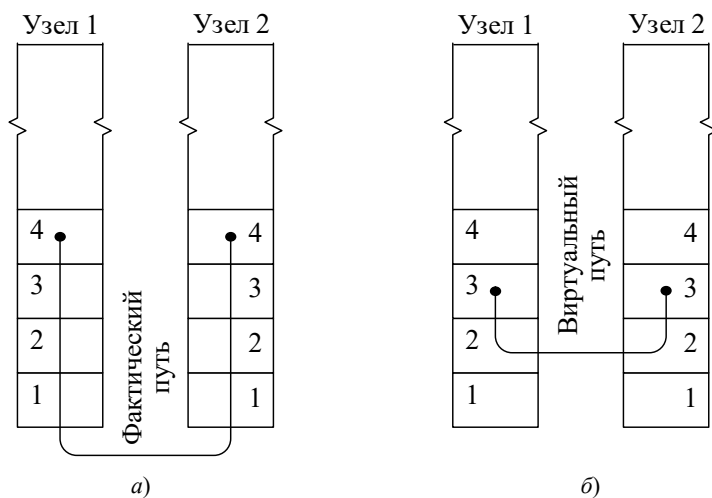
Единицей измерения на физическом уровне является бит. Физический уровень описывает способы передачи битов между устройствами по физической среде, параметры сигналов, которые передаются в той или иной среде, тип модуляции сигнала и другие параметры, задачи синхронизации и линейного кодирования.

Физический уровень – по сути, фундамент, на котором строится реальное соединение между сетевыми уровнями разных узлов (рис. 2.1, а).

Канальный уровень модели OSI отвечает за локальную связь между устройствами. На втором уровне происходит проверка целостности и правильности передачи данных, поступающих с физического уровня. Единицей измерения на канальном уровне является кадр, который содержит биты полезной информации и биты служебной информации. Кадр имеет свою строго определенную структуру.

Если на физическом уровне в качестве среды передачи выступает реальная физическая среда, которую можно «потрогать», то на втором уровне модели OSI в качестве среды передачи рассматривается виртуальный канал.

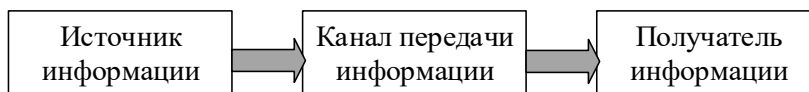
Виртуальный канал образует виртуальное соединение, которое не зависит от физической среды и среды распространения сигнала (рис. 2.1, б).



**Рис. 2.1. Соединение узлов: а – реальное соединение; б – виртуальное соединение**

В общем случае *система передачи данных* – это совокупность средств, служащих для передачи информации. На рис. 2.2 представлена метаструктура системы передачи информации.





**Рис. 2.2. Метаструктура системы передачи информации**

Задача источника информации – преобразовать физическое сообщение в, например, электрический сигнал, который будет передан по каналу передачи получателю информации. Получатель информации выполнит обратное преобразование электрического сигнала в требуемый физический носитель информации.

Передача данных может быть связана с разными технологическими явлениями. Наиболее широко она сопряжена с индустрией компьютерных коммуникаций. Передача данных в таком аспекте – это обмен файлами (отправка, получение), папками и иными реализациями машинного кода. Источниками информации могут быть и ТВ-камера, телефонный аппарат, автоматический датчик. У этих источников первичный информационный сигнал представлен в аналоговой форме.

Однако современный тренд развития коммуникационных технологий таков, что каналы передачи данных, какого бы типа информация не передавалась посредством них, активно «оцифровываются». Поэтому одна из первых задач современных технологий передачи данных связана с кодированием источника.

## **2.1. Элементы процессов передачи данных на физическом уровне**

Рекомендации X.200 и ISO 7498 определяют понятия, назначения и выполняемые функции физического уровня.

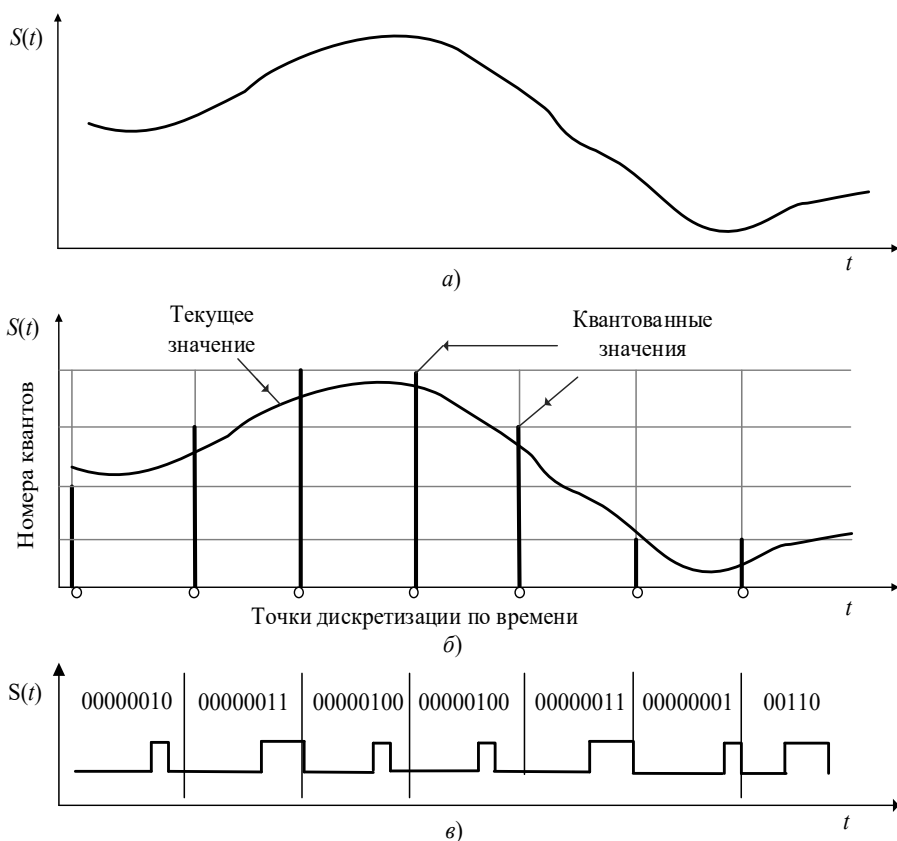
### **2.1.1. Кодирование источника**

Во многих микропроцессорных системах и персональных ЭВМ для представления алфавитно-цифровых символов и их передачи используется код *ASCII* (*American Standart Code for Information Interchange* – американский код обмена информацией), расширенный путем добавления букв русского алфавита. Для представления каждого символа отводится один байт.

Все виды представления символьной информации отличаются только кодировкой символов.

При «оцифровании» аналоговых сигналов, снимаемых, например, с автоматических датчиков, кодирование источника осуществляется с использованием трех процедур: дискретизации, квантования, кодирования.

Дискретизация и квантование преобразуют непрерывные во времени  $t$  значения аналогового сигнала  $S$  (рис. 2.3, а) в дискретные (рис. 2.3, б), которые отображаются в «цифре» (рис. 2.3, в). Такой вид преобразования называют импульсно-кодовой модуляцией (ИКМ).



**Рис. 2.3. Дискретизация, квантование, кодирование**

Дискретизация аналоговых сигналов по времени основана на *теории отображения Найквиста – Котельникова*. В соответствии с этой теорией, аналоговая непрерывная функция, переданная в виде последовательности ее дискретных по времени значений, может быть точно восстановлена, если частота дискретизации была в два или более раз выше, чем частота самой высокой гармоники спектра исходной функции. Если это условие не соблюдается, то восстановленная функция будет существенно отличаться от исходной.

Преимуществом цифровых методов записи, воспроизведения и передачи аналоговой информации является возможность контроля достоверности считанных с носителя или полученных по линии связи данных. Для этого можно применять те же методы, которые применяются для компьютерных данных – вычисление контрольной суммы, повторная передача искаженных кадров, применение самокорректирующихся кодов.

Представленные в цифровой форме непрерывные данные легко можно передать через компьютерную сеть. Для этого достаточно поместить несколько замеров в кадр какой-нибудь стандартной сетевой технологии, снабдить кадр правильным адресом назначения и отправить адресату.

### 2.1.2. Понятие канала связи

Канал связи – система технических средств и среда распространения сигналов для передачи сообщений от источника к получателю (и наоборот).

В самом общем виде функциональная структура канала связи представлена на рис. 2.4.



**Рис. 2.4. Функциональная структура канала связи**

Передатчик с помощью модулятора (М) и усилителя (У) формирует сигнал  $S(t)$ , в одном или нескольких параметрах которого отображается подлежащая передаче информация (данные)  $A(t)$ . Сигнал  $S(t)$  поступает на вход приемного устройства (приемника) в виде сигнала  $S^*(t)$ . Определенное отличие сигнала  $S^*(t)$  от сигнала  $S(t)$  возможно в силу действия помех и не идеальности характеристик физической среды. Далее из сигнала  $S^*(t)$  с помощью усилителя и демодулятора (Д) приемного устройства выделяется информация  $A^*(t)$  и передается потребителю с определенной степенью достоверности.

Обмен данными может осуществляться посредством трех основных типов каналов: дуплексного, симплексного и полудуплексного. При дуплексе передача данных по каналу возможна одновременно в обе стороны. В симплексном канале передача возможна только в одну сторону. Полудуплексные каналы обеспечивают передачу в обе стороны, но поочередно.

Среды передачи данных разбиваются на две большие категории:

- кабельная среда передачи данных;
- беспроводная среда передачи данных.

В современных компьютерных сетях чаще всего используются: витые пары, оптоволоконные провода, коаксиальные кабели, USB<sup>2</sup>-кабели, телефонные провода.

Различают два типа витых пар: неэкранированную витую пару (Unshielded Twisted Pair, UTP) и экранированную витую пару (Shielded Twisted Pair, STP).

Сегодня практически все сети проектируются на базе UTP и волоконно-оптических кабелей, коаксиальный кабель применяют лишь в исключительных случаях.

---

<sup>2</sup> **USB** (*Universal Serial Bus* – «универсальная последовательная шина») – последовательный интерфейс для подключения периферийных устройств к вычислительной технике.

Каждая среда вносит затухание в передаваемый сигнал. Витая пара отличается высокой степенью затухания, из-за чего дальность передачи по витой паре существенно ограничена.

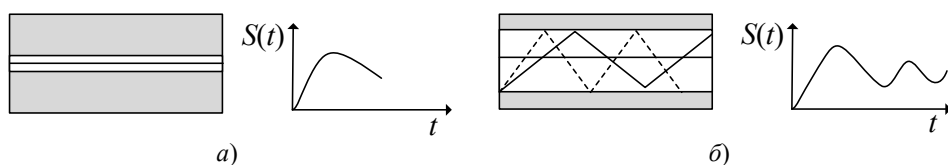
В зависимости от электрических и механических характеристик кабеля UTP разделяются на 7 категорий. Кабели категории 1 применяются там, где требования к скорости передачи минимальны. Кабели категории 3 широко распространены и предназначены как для передачи данных, так и для передачи голоса. Кабели категории 5 и выше специально разработаны для поддержки высокоскоростных протоколов FDDI, Fast Ethernet, ATM, Gigabit Ethernet.

Достоинством сети на базе витой пары является низкая стоимость оборудования и возможность использования имеющейся телефонной сети.

Существует несколько типов коаксиального кабеля, отличающихся характеристиками и областями применения – для локальных сетей, глобальных сетей, для кабельного телевидения. Расстояние между передающими устройствами до двух километров. Обычные скорости передачи данных –  $2.5 \div 100$  Мбит/с.

Оптоволоконный кабель состоит из тонких (5-60 микрон) волокон, по которым распространяются световые сигналы. Скорости передачи данных – 2 и более Гбит/с.

Существует два типа источников света: одномодовые (рис. 2.5, а) и многомодовые (рис. 2.5, б).



**Рис. 2.5. Типы источников света: а – одномодовый, б – многомодовый**

При *одномодовом* источнике расстояние до 100 км, при *многомодовом* около 4 км. Это естественно вытекает из физики явления: одномодовый источник выдает жестко сфокусированный пучок; многомодовый – значительно более рассеянный.

**Беспроводная среда передачи данных** применяется в случае, когда большое расстояние или препятствия затрудняют применение другого носителя.

Существует два основных типа беспроводной среды передачи данных: микроволновое и инфракрасное излучение.

**В инфракрасных средах передачи данных** применяется свет.

Беспроводная передача данных в компьютерных сетях осуществляется чаще всего через стандарты:

- «малого радиуса» – Bluetooth;
- «среднего радиуса» – Wi-Fi;
- «большого радиуса» – 3G, 4G<sup>3</sup>, WiMAX.

<sup>3</sup> 3G, 4G – поколения мобильной связи с повышенными требованиями.

Скорость, с которой передаются файлы, может сильно разниться в зависимости от того или иного стандарта связи, равно как устойчивость соединения и защищенность его от помех.

Выбор сетевой среды передачи данных диктуется типом сети и выбранной топологией.

На линиях связи большой протяженности обычно используется промежуточная аппаратура, которая решает две основные задачи:

- улучшение качества сигнала;
- создание составного канала связи между любой парой абонентов сети.

В локальных сетях промежуточная аппаратура может совсем не использоваться, если протяженность физической среды – кабелей или радиозэфира – позволяет одному сетевому адаптеру принимать сигналы непосредственно от другого сетевого адаптера без промежуточного усиления. В противном случае применяются устройства типа повторителей и концентраторов.

В глобальных сетях необходимо обеспечить качественную передачу сигналов на расстояния в сотни и тысячи километров. Поэтому без усилителей сигналов, установленных через определенные расстояния, построить территориальную линию связи невозможно. В глобальной сети необходима также и промежуточная аппаратура другого рода – мультиплексоры, демультиплексоры и коммутаторы. Эта аппаратура создает между двумя абонентами сети составной канал из некоммутируемых отрезков физической среды – кабелей с усилителями.

Между мультиплексорами и коммутаторами используется высокоскоростная физическая среда, например волоконно-оптический или коаксиальный кабель, по которому передаются одновременно данные от большого числа сравнительно низкоскоростных абонентских линий. Такой высокоскоростной канал обычно называют уплотненным каналом.

Промежуточная аппаратура образует сложную *первичную сеть*. Первичная сеть служит основой для построения инфокоммуникационных сетей.

Особенности построения первичных сетей будут рассмотрены специально в разделе 5.8.2.

### 2.1.3. Характеристики сигналов и каналов связи

Передача по каналу связи сопряжена с искажениями сигналов. Чтобы эти искажения были в допустимых пределах, необходимо согласование характеристик сигналов и каналов.

Для описания общих свойств сигнала вполне достаточно указания основных его характеристик:

- *длительности сигнала*  $T_c$  – время передачи сигнала по каналу;
- *ширины частотного спектра* сигнала  $\Delta F_c$ ;
- *превышения сигнала над помехой*  $H_c = \log(P_c/P_{ш})$ , где  $P_c$  – средняя мощность сигнала,  $P_{ш}$  – средняя мощность помехи (шума).

Часто употребляемыми характеристиками являются объем сигнала  $V_c = T_c \Delta F_c H_c$  и произведение  $T_c \Delta F_c$ . При  $T_c \Delta F_c \gg 1$  сигнал считается широкополосным. Широкополосные сигналы реализуют высокие скорости передачи данных.

В свою очередь для характеристик канала используют:

- *полосу пропускания*  $\Delta F_k$  – эффективную полосу частот колебаний, пропускаемых каналом связи без значительного ослабления;
- *время действия канала*  $T_k$  – время, в течение которого канал связи предоставлен для передачи сигналов;
- *динамический диапазон*  $D_k = \log_2(P_{\max}/P_{\min})$ , который зависит от чувствительности приемника  $P_{\min}$  и допустимых нагрузок  $P_{\max}$  аппаратуры канала связи.

Емкость канала  $V_k$  определяется по формуле  $V_k = T_k \Delta F_k D_k$ .

Канал, по сути, – это фильтр. Чтобы сигнал прошел через него без искажений, емкость этого канала должна быть больше объема сигнала или равна ему, т. е.  $V_k \geq V_c$ .

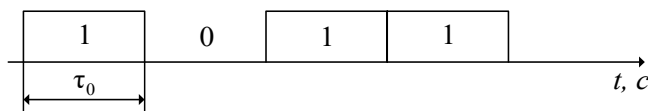
## 2.1.4. Скорость передачи данных

В числе важнейших параметров инфокоммуникационной сети – показатели скорости передаваемых данных в сети. Скорость передачи данных – это фактический показатель, отражающий, какой объем файлов может направляться с одного компьютера на другой за установленный промежуток времени.

Различают техническую и информационную скорости передачи данных.

Техническая скорость измеряется в *бодах* ( $B$ ) – количеством элементарных посылок (рис. 2.6), передаваемых в единицу времени – в секунду;  $B = 1 / \tau_0$  [бод].

Принято считать, что спектр сигнала прямоугольной посылки длиной  $\tau_0$  занимает полосу частот  $\Delta F_c = 1/\tau_0$ . В силу неидеальности частотной характеристики каналов должно выполняться условие  $\Delta F_k \geq \Delta F_c$ , где  $\Delta F_k$  – полоса частот канала, отводимая для передачи данных.

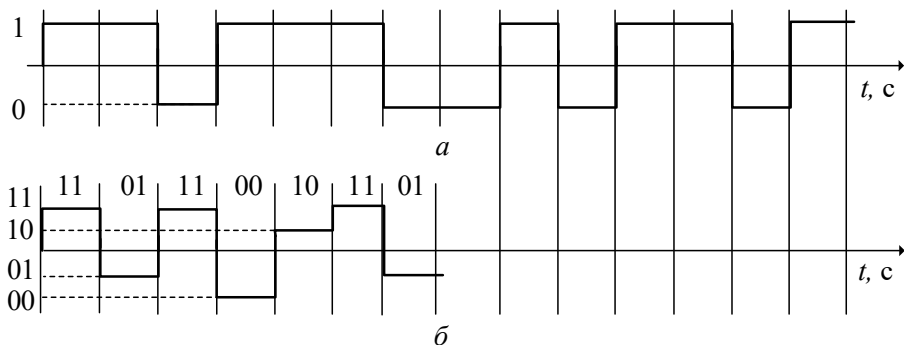


**Рис. 2.6. Последовательность элементарных посылок**

Информационная скорость передачи  $V$  измеряется в битах в секунду – бит/с, передаваемых последовательностью посылок, а также в производных единицах, таких как килобит в секунду (Кбит/с), мегабит в секунду (Мбит/с), гигабит в секунду (Гбит/с) и т.д.

Если сигнал посылки имеет 2 различных состояния, то (при равновероятном выборе уровня сигнала) каждая посылка переносит один бит информа-

ции (рис. 2.7, а). Если же передатчик использует более чем 2 устойчивых состояния сигнала для кодирования данных, то скорость передачи информации повышается – за один такт работы передатчик передает несколько бит исходных данных. Например, 2 бита в одной посылке можно передавать при наличии четырех различных состояний сигнала (рис. 2.7, б).



**Рис. 2.7. Повышение скорости передачи за счет многоуровневого сигнала**

Величина  $V$  – максимально возможная скорость передачи информации, которую называют *пропускной способностью* линии. Действие помех снижает скорость передачи. Так, если вероятность ошибочного приема сигнала посылки в канале с независимыми ошибками  $p$ , то реальную скорость передачи информации можно оценить как  $R = V(1 - p)$ .

Скорость передачи информации зависит от частотных характеристик канала связи. Связь между полосой пропускания канала и его пропускной способностью, вне зависимости от принятого способа физического кодирования, выражается формулой К. Шеннона:

$$V = F \log_2(1 + P_c/P_{\text{ш}}),$$

где  $P_c$  – мощность сигнала,  $P_{\text{ш}}$  – мощность шума

Значение вероятности ошибочного приема сигнала посылки  $p$  для каналов связи без дополнительных средств защиты от ошибок (например, самокорректирующихся кодов или протоколов с повторной передачей искаженных кадров) составляет, как правило,  $10^{-4} \div 10^{-6}$ , в оптоволоконных линиях связи –  $10^{-9}$ . Значение достоверности передачи данных, например, при  $p = 10^{-4}$  говорит о том, что в среднем из 10 000 бит искажается значение одного бита.

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

При передаче данных по каналам связи применяются два типа физического кодирования – на основе синусоидального несущего колебания и на после-

довательности прямоугольных импульсов (посылок). Первый способ часто называется также *модуляцией*, подчеркивая тот факт, что кодирование осуществляется за счет изменения параметров аналогового сигнала. Второй обычно называют *цифровым кодированием*.

Термины «модуляция» и «кодирование» часто используют как синонимы.

### 2.1.5. Модуляция несущего колебания

Одной из основных тенденций развития сетевых технологий является передача в одной сети как дискретных, так и аналоговых по своей природе данных. Источниками дискретных данных являются компьютеры и другие вычислительные устройства, а источниками аналоговых данных являются такие устройства, как телефоны, видеокамеры, звуко- и видеовоспроизводящая аппаратура. На ранних этапах решения этой проблемы в территориальных сетях все типы данных передавались в аналоговой форме, при этом дискретные по своему характеру компьютерные данные преобразовывались в аналоговую форму с помощью модемов.

Устройство, которое выполняет функции модуляции несущей синусоиды на передающей стороне и демодуляции на приемной стороне, носит название *модем* (модулятор-демодулятор). Модулятор переносит спектр сигнала данных в полосу пропускания канала связи и демодулятор – обратно (рис. 2.8).

Однако по мере развития техники съема и передачи выяснилось, что передача данных в аналоговой форме сопряжена с существенными искажениями, которые трудно устранить при приеме.

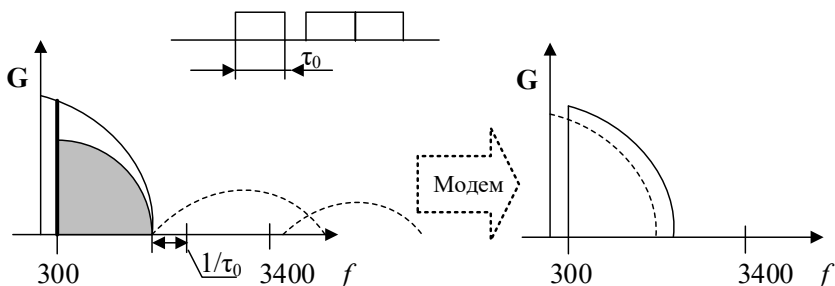
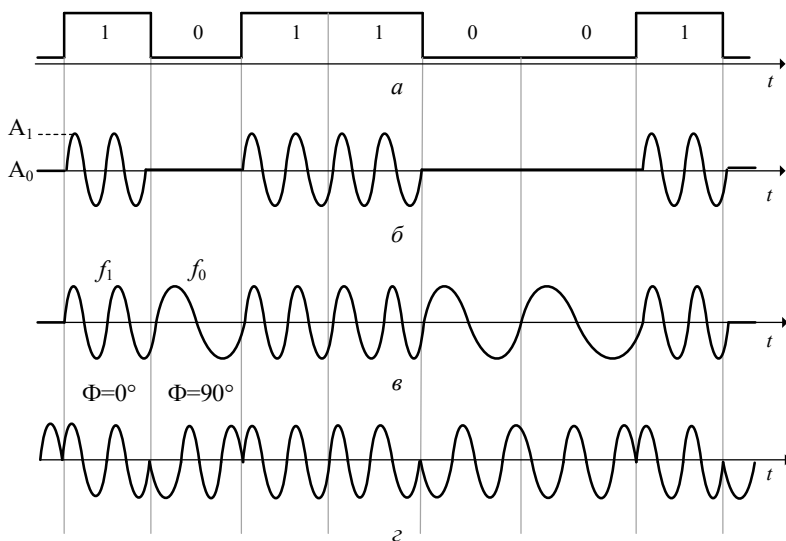


Рис. 2.8. Роль телефонного модема

Поэтому на смену аналоговой технике записи и передачи звука и изображения пришла цифровая техника. Эта техника использует так называемую дискретную модуляцию сигнала переносчика.

Модуляция является таким способом физического кодирования, при котором информация (данные) отображается (кодируется) в изменении амплитуды, частоты или фазы синусоидального сигнала несущей частоты. Основные способы модуляции показаны на рис. 2.9.





**Рис. 2.9. Различные типы аналоговой модуляции**

На диаграмме (рис. 2.9, а) показана последовательность бит исходной информации, представленная потенциалами высокого уровня для логической единицы и потенциалом нулевого уровня для логического нуля. Такой способ кодирования называется потенциальным кодом, который часто используется при передаче данных между блоками компьютера.

При **амплитудной модуляции** (рис. 2.9, б) для логической единицы выбирается один уровень амплитуды синусоиды несущей частоты ( $A_1$ ), а для логического нуля – другой ( $A_0$ ). Этот способ редко используется в чистом виде на практике из-за низкой помехоустойчивости, но часто применяется в сочетании с другим видом модуляции – фазовой модуляцией.

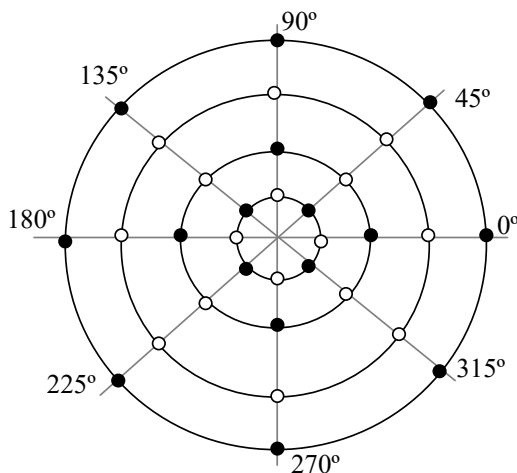
При **частотной модуляции** (рис. 2.5, в) значения 0 и 1 исходных данных передаются синусоидами с различной частотой –  $f_0$  и  $f_1$ . Этот способ модуляции не требует сложных схем и обычно применяется в низкоскоростных модемах.

При **фазовой модуляции** (рис. 2.5, г) значениям данных 0 и 1 соответствуют сигналы одинаковой частоты, но с различной фазой ( $\Phi$ ), например 0 и 180 градусов или 0,90,180 и 270 градусов.

В современных системах передачи данных используются дискретные методы модуляции несущего колебания. т. е. несколько счетных значений амплитуды, либо частоты, либо фазы.

Для повышения скорости передачи данных используют комбинированные методы модуляции. Наиболее распространенными являются дискретные методы **квадратурной амплитудной модуляции**. (*Quadrature Amplitude Modulation, QAM*). Эти методы основаны на сочетании фазовой модуляции с несколькими значениями величины сдвига фазы и амплитудной модуляции с несколькими уровнями амплитуды.

Так, например, при сочетании фазовой модуляции с 8 значениями величины сдвига фазы и амплитудной модуляции с 2 уровнями амплитуды (рис. 2.10) осуществляется, по сути, цифровое кодирование несущего колебания с основанием кода  $m=16$ , что позволяет в одной посылке передавать четыре бита первичного информационного сигнала.



**Рис. 2.10. Квадратурная амплитудная модуляция КАМ-16**

Однако из возможных 16 комбинаций сигнала посылки используются далеко не все. Например, в кодах *Треллиса* допустимы всего 6, 7 или 8 комбинаций для представления исходных данных, а остальные комбинации являются запрещенными. Такая избыточность кодирования требуется для распознавания модемом ошибочных сигналов, являющихся следствием искажений из-за помех в канале связи.

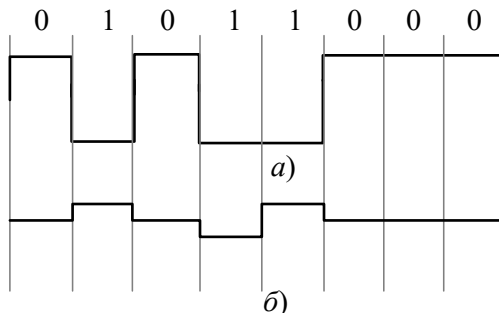
### 2.1.6. Цифровое кодирование

При цифровом кодировании дискретной информации, когда передача осуществляется в первичной полосе частот, то есть без использования сигнала-переносчика, применяют потенциальные и импульсные коды.

В потенциальных кодах для представления логических единиц и нулей используется только значение потенциала сигнала, а его перепады, формирующие законченные импульсы, во внимание не принимаются. Импульсные коды позволяют представить двоичные данные либо импульсами определенной полярности, либо частью импульса — перепадом потенциала определенного направления.

Наиболее простым потенциальным кодом является код без возвращения к нулю (NRZ) (рис. 2.11, а), однако он не является самосинхронизирующимся и создает постоянную составляющую.

Метод NRZ прост в реализации, обладает хорошей распознаваемостью ошибок (из-за двух резко отличающихся потенциалов), но не обладает свойством самосинхронизации. При передаче длинной последовательности единиц или нулей сигнал на линии не изменяется, что создает постоянную составляющую в принимаемом сигнале.



**Рис. 2.11. Потенциальный код: *a* – код без возвращения к нулю NRZ; *б* – (NRZI)**

Для улучшения свойств потенциального кода NRZ используются методы логического кодирования, исключающие длинные последовательности нулей. Эти методы основаны:

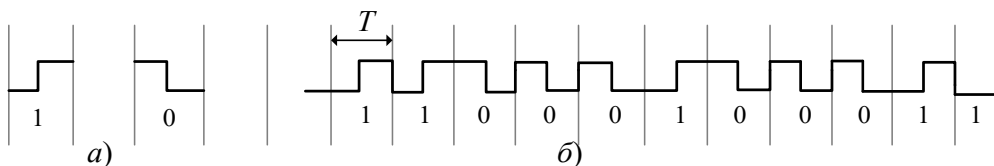
- на введении избыточных бит в исходные данные (коды типа 4B/5B);
- скремблировании исходных данных.

Операция скремблирования заключается в предварительном «перемешивании» исходной информации таким образом, чтобы вероятность появления единиц и нулей на линии становилась близкой. При скремблировании используется известный алгоритм, поэтому приемник, получив двоичные данные, передает их на *дескремблер*, который восстанавливает исходную последовательность бит.

Улучшенные потенциальные коды обладают более узким частотным спектром, чем импульсные, поэтому они находят применение в высокоскоростных технологиях, таких как FDDI, Fast Ethernet, Gigabit Ethernet.

Самым распространенным методом кодирования, применяемым в технологиях Ethernet и Token Ring, является манчестерский код.

*Манчестерский код* – это самосинхронизирующийся двоичный код без постоянной составляющей, в котором значение каждого передаваемого бита определяется направлением смены логического уровня в середине обусловленного заранее временного интервала (такта  $T$ , рис. 2.12). При двух логических уровнях у бита (1 и 0), вариантов тут немного: либо смена  $1 \Rightarrow 0$ , либо  $0 \Rightarrow 1$ . Согласно общепринятым стандартам для манчестерского кода переход от нуля к единице считается 1, а если наоборот, то 0 (рис. 2.12, *a*). Кодированная манчестерским кодом битовая последовательность показана на рис. 2.12, *б*.



**Рис. 2.12. Пример манчестерского кодирования: а – код «1» и «0»; б – пример передачи данных**

### 2.1.7. Синхронизация при передаче данных

При обмене данными на физическом уровне единицей информации является бит, поэтому средства физического уровня всегда поддерживают побитовую синхронизацию между приемником и передатчиком.

Существуют различные способы битовой (тактовой) самосинхронизации. Наиболее употребительным решением является система тактовой синхронизации с дискретным управлением.

Фазовый дискриминатор позволяет сравнить два сигнала и выделить опережение или отставание одного относительно другого. Инерционная схема вызывает устойчивое расхождение.

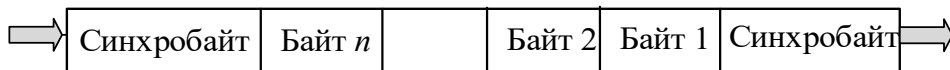
Специальный генератор ВЧ вырабатывает импульсы высокой частоты  $f_{вч}$ . Последовательность этих импульсов поступает на делитель частоты, на выходе которого получают последовательность тактовых импульсов. Управляющая схема добавляет или исключает отдельные импульсы ВЧ, подтягивая или отпуская тактовые импульсы на приемной стороне. Добавление в ВЧ-последовательности импульсов приводит к смещению фазы в сторону опережения на шаг  $\Delta t = 1/f_{вч}$  и наоборот. Это смещение фазы в долях длительности элементарной посылки называется шагом коррекции  $\delta\phi = \Delta t/\tau_0$ . Обычно значение шага коррекции составляет 1-3% от длительности такта.

Канальный уровень оперирует кадрами данных и обеспечивает синхронизацию между приемником и передатчиком на уровне байт. В обязанности приемника входит распознавание начала первого байта кадра, распознавание границ полей кадра и распознавание признака окончания кадра.

Обычно достаточно обеспечить синхронизацию на указанных двух уровнях — битовом и байтовом, чтобы передатчик и приемник смогли обеспечить устойчивый обмен информацией.

При синхронном режиме передачи пользовательские данные собираются в кадр, который предваряется байтами синхронизации (рис. 2.13). Байт синхронизации — это байт, содержащий заранее известное кодовое слово, например 0111110, которое оповещает приемник о приходе кадра данных. При его получении приемник должен войти в байтовый синхронизм с передатчиком, то есть правильно понимать начало очередного байта кадра. Иногда применяется несколько синхробайт для обеспечения более надежной синхронизации приемника и передатчика. Так как при передаче длинного кадра у приемника могут по-

явиться проблемы с синхронизацией бит, то в этом случае используются само-синхронизирующиеся коды.



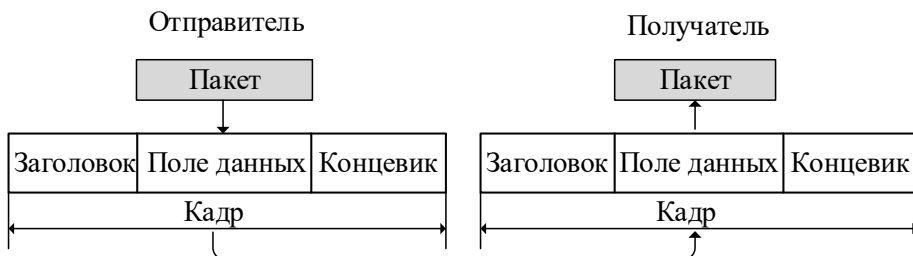
**Рис. 2.13. Синхронная передачи на уровне байт**

Функции физического уровня реализуются во всех устройствах, подключенных к сети. Повторители и концентраторы – это то оборудование, которое работает только на физическом уровне. Со стороны компьютера функции физического уровня выполняются сетевым адаптером.

## 2.2. Методы передачи на канальном уровне

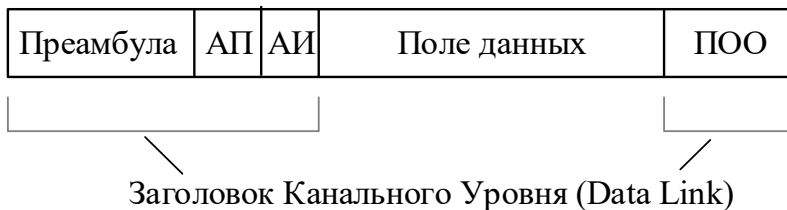
### 2.2.1. Общая структура кадра

Канальный уровень обеспечивает передачу протокольных блоков данных (пакетов), поступающих от протоколов верхних уровней (рис. 2.14). Адрес узла назначения также указывает протокол верхнего уровня.



**Рис. 2.14. Взаимодействие между пакетами и кадрами**

Каждый стандарт локальной сети определяет свой формат кадра. Они различаются по длине, расположению полей, однако, в независимости от типа сети, структура кадра одинакова (рис. 2.15).



**Рис. 2.15. Структура кадра**

Назначение полей в кадре:

Преамбула (Preamble) – служит для синхронизации работы приемника и передатчика;

АП – Адрес Приемника (DA, Destination Address) – адрес станции, которой направляется пакет;

АИ – Адрес Источника (SA, Source Address) – адрес передающей станции;

Поле Данных (Data) – содержит управляющую информацию, собственно данные либо пакет, поступающий с сетевого уровня;

ПОО – Поле Обнаружения Ошибок (CRC) – служит для определения достоверности полученной информации.

В качестве адресов могут использоваться логические или физические адреса.

*Логический адрес (Logical Address)* – определяется используемым протоколом обмена данными и может быть изменен в процессе работы. С помощью логических адресов можно создать группы устройств, выполняющих одинаковые функции, – серверы, мосты и т.п. Это упрощает управление работой сети. Логические адреса используются, например, протоколом TCP/IP.

*Физический адрес (Physical Address)* – определяется стандартом локальной сети, однозначно идентифицирует в сети данный узел (node) и не может быть изменен после подключения устройства к сети. В Ethernet на сетевом адаптере устанавливается ПЗУ, в которой прошит физический адрес сетевого адаптера. Изменить его можно, только заменив микросхему ПЗУ.

В качестве адреса приемника могут использоваться:

**Широковещательный, или Общий, Адрес (Broadcast).** Пакет с таким адресом принимается и обрабатывается всеми станциями сети. Каждый стандарт локальной сети определяет такой адрес. Например, в Ethernet это пакет, у которого в поле адреса приемника все символы "FF" hex. Широковещательный адрес используется и при логической адресации.

**Групповой Адрес (Multicast).** Пакет с таким адресом принимается и обрабатывается определенной группой станций. Например, только серверами, только маршрутизаторами и т.п. Этот адрес может быть только логическим.

**Частный Адрес (Unicast или Private).** Пакет с таким адресом принимается и обрабатывается только определенной станцией, адрес которой соответствует частному адресу. В качестве частных адресов используются логические или физические адреса.

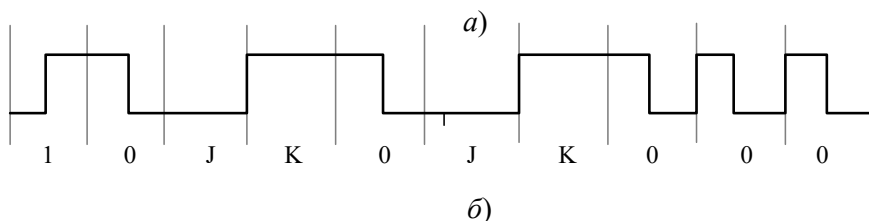
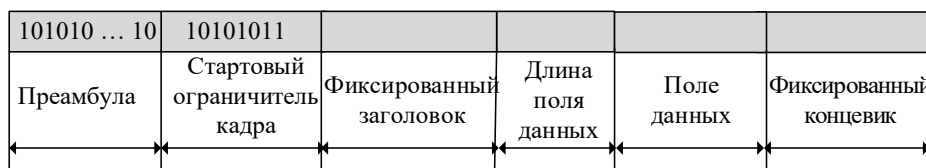
Протоколы канального уровня оформляют переданные им пакеты в кадры собственного формата, помещая указанный адрес назначения в одно из полей такого кадра, а также сопровождая кадр контрольной суммой. Протокол канального уровня имеет локальный смысл. Он предназначен для доставки кадров данных, как правило, в пределах локальных сетей с однотипной или близкой технологией. Например, в односегментных сетях Ethernet или же в многосегментных сетях Ethernet и Token Ring иерархической топологии, разделенных только мостами и коммутаторами. Во всех этих конфигурациях адрес назначения имеет локальный смысл для данной сети и не изменяется при прохождении

кадра от узла-источника к узлу назначения. Возможность передавать данные между локальными сетями разных технологий связана с тем, что в этих технологиях используются адреса одинакового формата.

Другой областью действия протоколов канального уровня являются связи типа «точка-точка» глобальных сетей, когда протокол канального уровня ответственен за доставку кадра непосредственному соседу. Адрес в этом случае не имеет принципиального значения. На первый план выходит способность протокола восстанавливать искаженные и утерянные кадры, поскольку недостаточно высокое качество территориальных каналов часто требует выполнения подобных действий. Такие протоколы называют линейными, они регламентируют передачу данных между соседними узлами в сети.

И в той и другой областях для передачи данных на канальном уровне используются бит-ориентированные синхронные протоколы.

Структура кадра при бит-ориентированной передаче показана на рис. 2.16, а.



**Рис. 2.16. Способы выделения начала и конца кадра при синхронной передаче**

В схеме (см. рис. 2.16, а) начало кадра задает стартовый ограничитель кадра (флаг), а конец кадра фиксированный концевик. Чтобы все остальные станции вошли в битовую синхронизацию, передающая станция предваряет содержимое кадра последовательностью бит, известной как преамбула. Преамбула состоит из чередования единиц и нулей 101010... Войдя в битовую синхронизацию, приемник исследует входной поток на побитовой основе, пока не обнаружит байт начала кадра 10101011. За этим байтом следует заголовок кадра, в котором в определенном месте находится поле длины поля данных.

В схеме (рис. 2.16, б) для обозначения начала кадра имеется стартовый флаг, а для определения конца кадра используется поле длины кадра. Чтобы все остальные станции вошли в битовую синхронизацию, передающая станция предваряет содержимое кадра последовательностью бит, известной как преамбула. Преамбула состоит из чередования единиц и нулей 101010... Войдя в битовую синхронизацию, приемник исследует входной поток на побитовой осно-

ве, пока не обнаружит байт начала кадра 10101011. За этим байтом следует заголовок кадра, в котором в определенном месте находится поле длины поля данных.

При передаче кадров данных на канальном уровне используются как дейтаграммные процедуры, работающие без установления соединения (*connectionless*), так и процедуры с предварительным установлением логического соединения (*connectionoriented*).

Особенности этих режимов передачи данных рассмотрены в разделе «Глобальные сети».

### 2.2.2. Обнаружение и исправление ошибок

Канальный уровень должен обнаруживать ошибки передачи данных, связанные с искажением бит в принятом кадре данных или с потерей кадра, и по возможности их корректировать. Для обнаружения и исправления возможных ошибок применяется избыточное кодирование.

Существует два вида введения избыточности:

- Кодовая избыточность – сам метод кодирования подразумевает внесения избыточности посредством проверочных символов.
- Избыточность повторений – при приеме неправильного блока посылается запрос на его повторение, либо принятие решений осуществляется мажоритарным методом (2 из 3, 3 из 5, и т.п.).

Обычно используют сочетания двух видов избыточности. Например, при кодировании сообщения в него вносится кодовая избыточность, которая используется для обнаружения ошибок. При их обнаружении на приемной стороне через канал обратной связи производится запрос на повторение передаваемого блока.

Все зависит от конкретных условий (характеристик КС, метода передачи, метода (его реализации) введения избыточности).

При симплексной передаче – постоянная избыточность (кодирование с исправлением ошибок, повторение фиксированное число раз и мажоритарный прием, совместное использование избыточного кодирования и повторения).

При дуплексе и полудуплексе – возможность использования переменной избыточности – сочетания кодирования и повторений.

При передаче данных используется бит четности – вертикальный контроль (*vertical redundancy control, VRC*). Семь битов байта дополняются восьмым с тем, чтобы сумма битов байта по модулю два была 0 либо 1.

Используется также горизонтальный контроль, когда суммируются по модулю два определенные биты различных байтов, а результат записывается на соответствующую позицию в специальном (контрольном) байте – продольный контроль (*LRC – longitudinal redundancy control*).

Кодирование VRC/LRC называется геометрическим кодированием.

Когда идет блоковая передача, то используется так называемая контрольная сумма (остаток R от деления суммы значений кодовых комбинаций, используемых в блоке, на 255).



Этот остаток приписывается к концу блока. На приемной стороне производится вычисление контрольной суммы по принятым байтам информационной части блока и сравнение контрольных сумм.

Такой механизм защиты может обнаружить до 97% случайных ошибок.

В сетях при синхронной передаче используется помехоустойчивое кодирование циклическим кодом (cyclic redundancy check, CRC).

Большая часть протоколов канального уровня выполняет только первую задачу – обнаружение ошибок, считая, что корректировать ошибки, то есть повторно передавать данные, содержавшие искаженную информацию, должны протоколы верхних уровней. Так работают, например протоколы локальных сетей Ethernet, Token Ring, FDDI. В этих сетях искажения и потери кадров являются очень редкими событиями, в условиях надежной работы сети предусматривать процедуры устранения ошибок являлось бы избыточным.

Напротив, если в сети искажения и потери случаются часто, то желательно уже на канальном уровне использовать протокол с коррекцией ошибок, а не оставлять эту работу протоколам верхних уровней. Протоколы верхних уровней, например транспортного или прикладного, работая с большими тайм-аутами, восстановят потерянные данные с большой задержкой. Поэтому существуют протоколы канального уровня, например LLC2 или LAP-B, которые самостоятельно решают задачу восстановления искаженных или потерянных кадров (см гл. 3).

Не следует считать, что один протокол лучше другого потому, что один восстанавливает ошибочные кадры, а другой – нет. Каждый протокол должен работать в тех условиях, для которых он разработан.

### 2.2.3. Методы восстановления искаженных и потерянных кадров

Коррекция ошибок основана на повторной передаче кадра данных в том случае, если кадр теряется и не доходит до адресата или приемник обнаружил в нем искажение информации. Отправитель нумерует отправляемые кадры и для каждого кадра ожидает от приемника так называемой *положительной квитанции* – служебного кадра, извещающего о том, что исходный кадр был получен и данные в нем оказались корректными. Время этого ожидания ограничено – при отправке каждого кадра передатчик запускает таймер. Если по истечении таймера положительная квитанция не получена, кадр считается утерянным. Приемник в случае получения кадра с искаженными данными может отправить *отрицательную квитанцию* – явное указание на то, что данный кадр нужно передать повторно.

Существуют два подхода к организации процесса обмена квитанциями: с простоями и с организацией «окна».

**Метод с простоями (Idle Source)** требует, чтобы источник, пославший кадр, ожидал получения квитанций (положительной или отрицательной) от приемника и только после этого посылал следующий кадр (или повторял искаженный). Если же квитанция не приходит в течение тайм-аута, то кадр считается утерянным и его передача повторяется. При этом методе производитель-

ность обмена данными существенно снижается – хотя передатчик и мог бы послать следующий кадр сразу же после отправки предыдущего, он обязан ждать прихода квитанции. Снижение производительности этого метода коррекции особенно заметно в низкоскоростных территориальных сетях.

Второй метод называется методом **скользящего окна (sliding window)**. В этом методе для повышения коэффициента использования линии источнику разрешается передать некоторое количество кадров в непрерывном режиме, то есть в максимально возможном для источника темпе, без получения на эти кадры положительных ответных квитанций. Количество  $W$  кадров, которые разрешается передавать таким образом, называется размером окна.

Метод скользящего окна более сложен в реализации, чем метод с простоями, так как передатчик должен хранить в буфере все кадры, на которые пока не получены положительные квитанции. Кроме того, требуется отслеживать несколько параметров алгоритма: размер окна  $W$ , номер кадра, на который получена квитанция, номер кадра, который еще можно передать до получения новой квитанции.

Некоторые методы используют отрицательные квитанции. Отрицательные квитанции бывают двух типов – групповые и избирательные. Групповая квитанция содержит номер кадра, начиная с которого нужно повторить передачу всех кадров, отправленных передатчиком в сеть. Избирательная отрицательная квитанция требует повторной передачи только одного кадра.

Метод скользящего окна реализован во многих протоколах: LLC2, LAP-B, X.25, TSP.

Метод с простоями является частным случаем метода скользящего окна, когда размер окна равен единице.

Повторная передача с использованием квитанции относится к классу протоколов с *решающей обратной связью* (РОС).

Существуют три основных варианта РОС:

- с ожиданием (РОС-ОЖ),
- с повторной передачей последовательности (РОС-ПП),
- с адресным переспросом (РОС-АП).

Варианты отличаются объемом дополнительной служебной информации в кадре, временем ожидания повторной передачи и расходом буферной памяти.

В качестве отличительных признаков отдельных кадров может использоваться их *циклическая нумерация* по  $\text{mod } M$  (по модулю  $M$ ), где  $M$ , как правило, равно 8 или 128).

Наличие нумерации кадров позволяет при любом варианте РОС использовать *квитанции в виде номера очередного ожидаемого кадра*.

При этом допускается передавать кадры с номерами, превышающими номер в последней квитанции, но не более некоторого количества кадров, соответствующего размеру так называемого «окна».

Длина информационной части кадра и размер окна являются основными управляемыми количественными параметрами протоколов канального уровня.

Для выявления аварийных ситуаций потери квитанций задается *тайм-аут*, после истечения которого автоматически принимается решение о повторной передаче.

Выбор тайм-аута зависит не от надежности сети, а от задержек передачи кадров сетью.

Во многих реализациях метода скользящего окна величина окна и тайм-аут выбираются адаптивно, в зависимости от текущего состояния сети.

Для управления передачей применяются синхронные бит-ориентированные протоколы. Наиболее распространенный HDLC (High-level Data Link Control).

## 2.2.4. Протокол канального уровня HDLC

*HDLC (High Level Link Control)* – протокол управления каналом передачи данных, является базовым стандартом и для построения других протоколов канального уровня (LAP, LAPB, LAPD, LAPX и LLC). Он реализует механизм управления потоком посредством скользящего окна и имеет необязательные возможности (опции), поддерживающие полудуплексную и полнодуплексную передачу, одноточечную и многоточечную конфигурации, а также коммутируемые и некоммутируемые каналы.

Существует три типа станций HDLC:

- **первичная** (ведущая), управляющая звеном передачи данных (каналом);
- **вторичная** (ведомая), работающая как зависимая по отношению к первичной станции (ведущей);
- **комбинированная** сочетает в себе одновременно функции первичной и вторичной станций.

Применяется два основных способа *конфигурирования канала* для обеспечения совместимости взаимодействия между станциями:

- **несбалансированная конфигурация** (*UN – Unbalanced Normal*), обеспечивающая работу одной первичной станции и одной или большего числа вторичных;
- **сбалансированная конфигурация** (*BA – Balanced Asynchronous*), состоящая из двух комбинированных станций. Станции имеют равный статус в канале и могут несанкционированно посылать друг другу трафик. Каждая станция несет одинаковую ответственность за управление каналом.

**Формат кадра HDLC.** Кадр – протокольный блок данных для передачи на канальном уровне (рис. 2.17).

Кадры подразделяются на три типа: **I** – информационные; **S** – управляющие или супервизорные; **U** – нумерованные (рис. 2.17).

*Информационные кадры* предназначены для передачи информации и содержат поле информации (Data). При передаче информационные кадры нумеруются путем использования разрядов поля управления N(S) – номер передаваемого кадра и N(R) – номер ожидаемого кадра.

Флаг		Адрес		Управляющее поле		Информационное поле		CRC	Флаг
1	2	3	4	5	6	7	8	Разряды	
0	N(S)			P/F	N(R)			I-формат	
1	0	S-коды		P/F	N(R)			S-формат	
1	1	U-коды		P/F	U-коды			U-формат	

**Рис. 2.17. Формат кадра и управляющего поля HDLC**

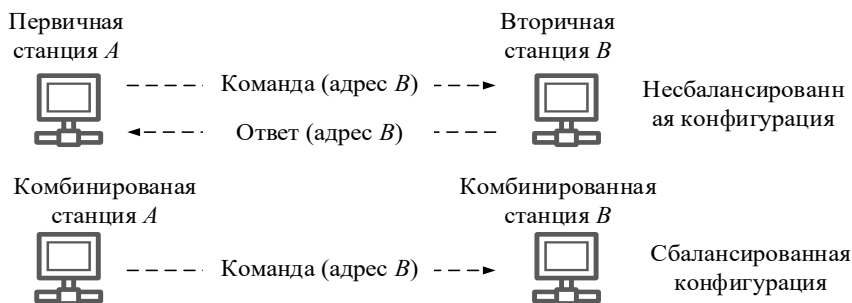
*Управляющие (супервизорные) кадры* предназначены для передачи команд и ответов (в служебном поле S из 2 бит), в том числе запросов на повторную передачу искаженных информационных кадров, начиная с номера N(R), если нет попутных информационных кадров.

*Ненумерованные кадры* предназначены для передачи ненумерованных команд и ответов при установлении соединения.

Все кадры должны начинаться и заканчиваться полями флага «01111110».

**Управление потоком** в HDLC осуществляется с помощью передающих и принимающих окон. Окно обеспечивает буферное пространство и правила нумерации (сообщений). Передающий узел поддерживает порядковый номер N(S) следующего по очереди I-кадра, который должен быть передан. Принимающий узел поддерживает номер N(R), который, как ожидается, является порядковым номером следующего I-кадра.

**Адресное поле** определяет первичную или вторичную станцию, участвующую в передаче конкретного кадра (рис. 2.18). Каждой станции присваивается уникальный адрес. В несбалансированной системе адресные поля в командах и ответах содержат адрес вторичной станции. В сбалансированных конфигурациях командный кадр содержит адрес получателя, а кадр ответа – адрес передающей станции.



**Рис. 2.18. Передача конкретного кадра**

В табл. 2.1 представлены команды и ответы, используемые в случае сбалансированной и несбалансированной конфигураций канала.

SNRM и SABM являются командами установки режима. HDLC требует, чтобы была установлена сбалансированная или несбалансированная конфигурация.

**Управляющее поле HDLC** задает тип команды или ответа, а также порядковые номера, используемые для отчетности о прохождении данных в канале между первичной и вторичной станциями. Формат и содержание управляющего поля определяют кадры трех упомянутых ранее типов: информационные (I), супервизорные (S) и нумерованные (U).

*Информационный формат* используется для передачи данных конечных пользователей между двумя станциями. *Супервизорный формат* выполняет управляющие функции: подтверждение (квитирование) кадров, запросы на повторную передачу кадров и ее временную задержку. *Ненумерованный формат* также используется для целей управления: инициализации или разъединения, тестирования, сброса и идентификации станции и т. д.

**Таблица 2.1**

**Команды и ответы, используемые в случае сбалансированной и несбалансированной конфигураций канала**

Конфигурация канала			
Несбалансированная (UN)		Сбалансированная (UB)	
Первичная	Вторичная	Первичная	Вторичная
Станция A		Станция B	
Команда	Ответ	Команда	Ответ
		–	REJ
		–	RR
–	RNR	–	RNR
SNRM	UA	SABM	UA
DISC	DM	DISC	DM
–	FRMR	–	FRMR

Бит P/F, или бит опроса/окончания, принимается во внимание только тогда, когда он установлен в 1. Он называется битом P, когда используется первичной станцией, и битом F, когда вторичной. Первичная станция использует бит P для опроса статуса вторичной станции. Так, P = 1 как бы говорит: «ответьте мне, потому что я хочу знать ваш статус». Вторичная станция отвечает на бит P кадром данных или кадром состояния с битом F. Бит F может также означать окончание передачи вторичной станцией.

*Информационное поле* содержит действительные данные пользователя. Оно имеется только в кадре информационного формата. *Примечание:* кадры «UI – ненумерованная информация» и «FRMR – Неприем кадра» ненумерованного формата имеют информационное поле.

*Поле CRC* (контрольная последовательность кадра) используется для обнаружения ошибок передачи между двумя станциями.

**Кодонезависимость и синхронизация HDLC.** HDLC является *кодпрозрачным протоколом*. Предусмотрена процедура *bit staffing*. Протоколу безразлично, какие кодовые комбинации находятся в потоке данных. Единственное, что требуется, – это поддерживать уникальность флагов. В HDLC используется

также два других сигнала: *сигнал аварийного завершения*, состоящий из последовательности единиц, число которых не меньше семи и не больше сорока; *состояние покоя*, которое представляется последовательностью сорока или большего числа единиц. *Межкадровое временное заполнение* сопровождается передачей между кадрами непрерывной последовательности флагов.

Управляющие (супервизорные) кадры предназначены для передачи команд и ответов: RR – *приемник готов* (Receiver Ready), REJ – *отказ* (Reject, REJ), RNR – *приемник не готов* (Receiver Not Ready), SREJ – *выборочный не прием* (Selective Reject) (в служебном поле S2 бит, рис. 2.17). Супервизорные кадры не содержат информационного поля, их назначение состоит в выполнении таких функций, как подтверждение (квитирование), опрос, временная задержка передачи данных, запрос на повторную передачу искаженных информационных кадров, начиная с номера N(R) (рис. 2.17). Супервизорный формат может быть использован и для подтверждения приема кадров от передающей станции.

Функции команд/ответов для нумерованного формата:

- SNRM (*Set Normal Response Mode – установить режим нормального ответа*) переводит вторичную станцию в NRM (режим нормального ответа). NRM предотвращает посылку вторичной станцией несанкционированных (unsolicited) кадров. Это означает, что первичная станция управляет всем потоком сообщений в канале;

- UA (*Unnumbered Acknowledgement – нумерованное подтверждение*) служит для подтверждения установления или разрыва соединения;

- DISC (*Disconnect – сброс соединения*) осуществляет запрос на разрыв соединения;

- DM (*отклик на кадр DISC*) указывает на разрыв соединения;

- SABM (*Set Asynchronous Balanced Mode*) – устанавливает асинхронный сбалансированный режим, который используют комбинированные станции. Этот режим обеспечивает двусторонний обмен потоками данных между станциями и является основным (рабочим) и наиболее часто используемым на практике;

- FRMR (*Frame Reject – не прием кадра*), используется для сообщения об ошибочной ситуации, которая не может быть устранена повторением кадра (искажение формата кадра).

**Системные параметры T1, N2, N1, K и рекомендации по их установке.**

**Таймер T1** запускается с момента передачи каждого кадра и используется для инициирования повторной передачи, в случае его завершения. Период таймера T1 должен быть больше, чем максимальное время между передачей кадра и приемом подтверждения.

**Счетчик N2** используется для определения максимального числа повторных передач, выполняемых по завершению таймера T1.

**Счетчик N1** определяет максимальную длину информационных полей (битов в I-кадре).

**Размер окна  $K$**  – максимальное число переданных, но не подтвержденных I-кадров. Оно не должно быть более 7.

Параметры  $T_1$ ,  $N_2$ ,  $N_1$  и  $K$  подлежат согласованию с администрацией на некоторый период времени.

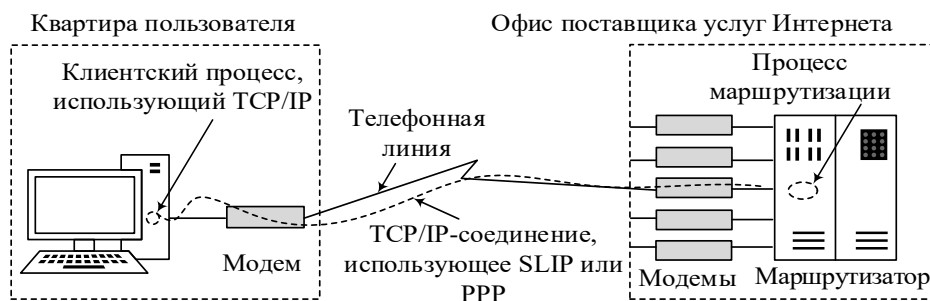
## 2.2.5. Уровень передачи данных в Интернете

На больших территориях инфраструктура строится на основе выделенных линий, соединяющих отдельные машины по принципу «точка – точка». На практике соединение «точка-точка» используется, прежде всего, в двух ситуациях.

Во-первых, вся связь с внешним миром локальных сетей осуществляется через один или два маршрутизатора, связанных выделенными линиями «точка-точка» с удаленными маршрутизаторами. Именно эти маршрутизаторы вместе с выделенными линиями образуют подсети, из которых состоит Интернет.

Во-вторых, соединения «точка-точка» связывают миллионы индивидуальных пользователей с Интернетом с помощью модемов и телефонных линий (рис. 2.19).

Как для соединения двух маршрутизаторов по выделенной линии, так и для соединения маршрутизатора с хостом требуется протокол, который бы занимался формированием кадров, обработкой ошибок и другими функциями уровня передачи данных.



**Рис. 2.19. Выход ПК в Интернет**

Широко распространенным в Интернете является PPP (Point-to-Point – протокол передачи от точки к точке).

Протокол PPP является механизмом формирования кадров, поддерживающим различные протоколы, которым можно пользоваться при модемных соединениях, в последовательных по битам линиях HDLC, сетях SONET и других физических средах. PPP поддерживает обнаружение ошибок, переговоры о параметрах, сжатие заголовков, а также, по желанию, надежное соединение с использованием кадров HDLC.

Протокол PPP описан в рекомендации RFC 1661 и доработан в некоторых более поздних документах RFC4 (например, RFC 1662 и 166).

<sup>4</sup> RFC – Requests for Comments, набор технических отчетов

## Контрольные вопросы

1. Можно ли аналоговые сигналы передавать по цифровым линиям (каналам) связи?
2. Как осуществляется «оцифрование» аналоговых сигналов?
3. Что в общем случае представляет собой канал связи?
4. Что такое широкополосность сигнала?
5. Что определяет качество передачи сигналов по линиям связи?
6. Скорости передачи данных бит/с и бод/с. В чем их отличие?
7. Для чего используется многократная фазовая модуляция?
8. Каково назначение систем синхронизации приемника с передатчиком?
9. Почему манчестерский код называют самосинхронизирующимся?
10. С какой целью используются коды типа 4В/5В?
11. Как представлены данные при синхронной и асинхронной передачах?
12. Какие методы защиты от ошибок используются в системах ПД?
13. Как передатчик определяет факт потери квитанции?
14. При перегрузке сети размер окна нужно увеличивать или уменьшать?
15. Почему протоколы канального уровня называют бит-ориентированными?



### 3. ПРОВОДНЫЕ ЛОКАЛЬНЫЕ СЕТИ

#### 3.1. Предназначение локальной сети

Целью создания локальных компьютерных сетей является совместное использование ресурсов и осуществление интерактивной связи как внутри одной организации (фирмы), так и за ее пределами. Ресурсы – это данные и приложения (программы), хранящиеся на сетевых дисках, и периферийные устройства, такие как внешний дисковод, принтер, сканер и т.д. Кроме того, возможны обмен информацией между всеми компьютерами сети и доступ пользователя с любого компьютера к ресурсам глобальных сетей.

Пример сегментированной LAN представлен на рис. 3.1. Топологии сегментов и устройства, объединяющие сегменты в единую сеть, рассмотрены в гл. 1.

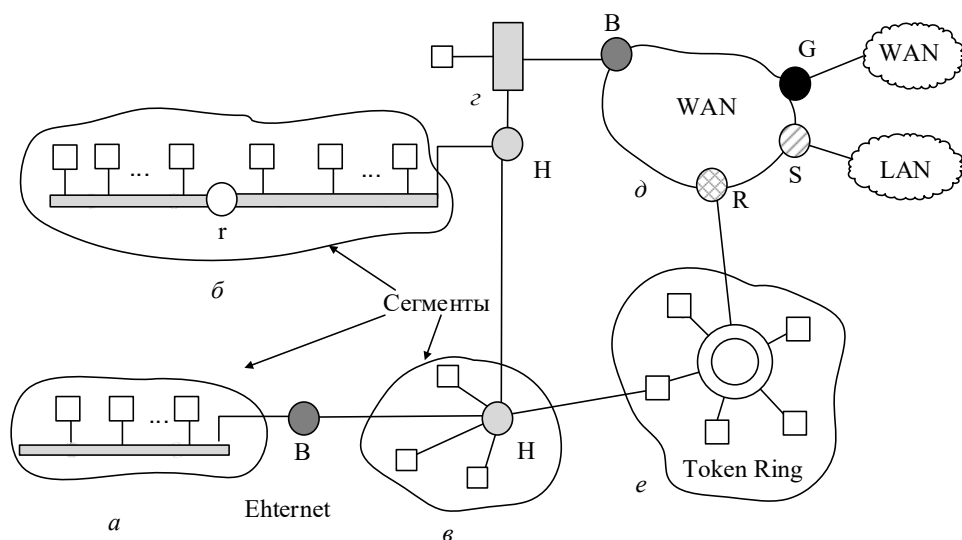


Рис. 3.1. Структура и элементы LAN:

**г** – повторитель (Repeater); **Н** – концентратор (Hub); **В** – мост (Bridge); **С** – коммутатор (Switch); **Р** – маршрутизатор (Router); **Г** – шлюз (Gateway)

Общие принципы построения LAN были рассмотрены в п. 1.5, 1.6, 1.7. Для простейших LAN характерно использование общей пассивной физической среды передачи, связывающей сразу все рабочие станции (PC), а не отдельные пары. Эта особенность технологий построения LAN связана с попыткой упростить их до минимума и не использовать в сети никаких активных систем передачи и коммутации, кроме самих PC (компьютеров). В дальнейшем технологии построения LAN стали допускать объединение автономных локальных сетей в ассоциации с помощью различных внутрисетевых коммуникационных устройств (см. рис. 3.1) как с целью уменьшения нагрузки на общие разделяемые физические среды, так и для расширения сетей, включая доступ к другим локальным и глобальным сетям.

Прототипом первых LAN послужила сеть Aloha, в которой использован случайный доступ к общей радиосреде. В 1980 году опубликован стандарт Ethernet версии II для локальной сети, построенной на основе коаксиального кабеля. «Ethernet», одна из наиболее популярных технологий локальных сетей, дословно переводится как «эфирная сеть».

### 3.2. Стандарты базовых локальных сетей

При организации взаимодействия узлов в локальных сетях основная роль отводится протоколу канального уровня.

Кабель (физическая среда) используется всеми компьютерами сети в режиме разделения времени.

Базовые технологии ЛС Ethernet и Token Ring специфицированы стандартами IEEE 802.1-802.5 (IEEE–Institute of Electronic and Electrical Engineers).

Стандарты семейства IEEE 802.x охватывают только физический и канальный уровни модели OSI. Канальный уровень ЛС разделен на два подуровня:

- подуровень управления доступом к среде (Media Access Control, MAC);
- подуровень логической передачи данных (Logical Link Control, LLC).

Функции подуровней представлены в табл. 3.1.

**Таблица 3.1**

#### **Функции подуровней**

<b>Уровень OSI</b>	<b>Элемент</b>	<b>Методы реализации</b>
Канальный – MAC	Логическая топология Доступ к среде Адресация	Шина, кольцо Состязание, маркер, опрос Физический (MAC-адрес)
Канальный – LLC	Способ передачи Сервис соединений	Асинхронная, синхронная Управление потоком данных, Контроль ошибок

Протоколы уровней MAC и LLC взаимно независимы – каждый протокол MAC-уровня может применяться с любым типом протокола LLC-уровня и наоборот.

Стандарт IEEE 802 содержит несколько разделов:

- в разделе 802.1 даются основные понятия и определения, общие характеристики и требования к локальным сетям;
- раздел 802.2 определяет подуровень управления логическим каналом LLC;
- разделы 802.3 – 802.5 регламентируют спецификации различных протоколов подуровня доступа к среде MAC и их связь с уровнем LLC:
  - стандарт 802.3 описывает коллективный доступ с опознаванием несущей и обнаружением конфликтов (Carrier sense multiple access with collision detection – CSMA/CD) – прототип стандарта Ethernet;

- стандарт 802.4 определяет метод доступа к шине с передачей маркера (Token bus network), прототип – ArcNet;
- стандарт 802.5 описывает метод доступа к кольцу с передачей маркера (Token ring network), прототип – Token Ring.

Для каждого из этих стандартов разработаны спецификации физического уровня, определяющие среду передачи данных (коаксиальный кабель, витая пара или оптоволоконный кабель), ее параметры, а также методы кодирования информации при передаче по определенной среде.

### 3.3. Протокол LLC уровня управления логическим каналом

Уровень LLC организует передачу кадров данных между узлами с различной степенью надежности, а также реализует функции интерфейса с прилегающим к нему сетевым уровнем (рис. 3.2).

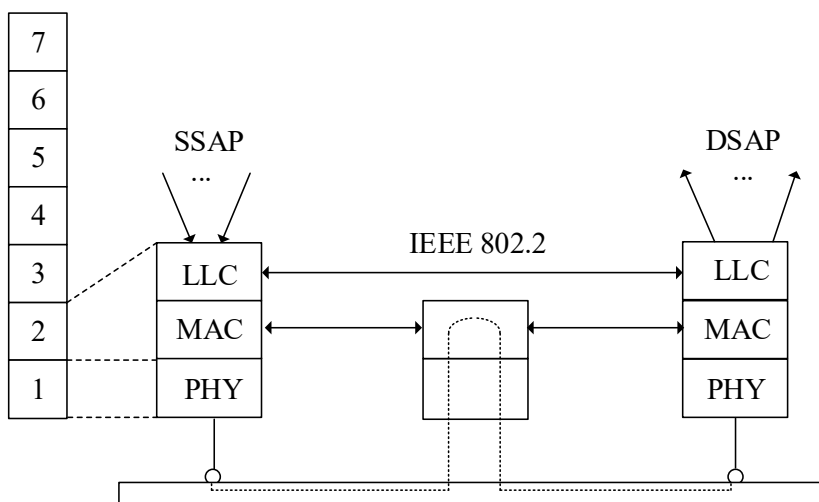


Рис. 3.2. Архитектура протокола LLC уровня

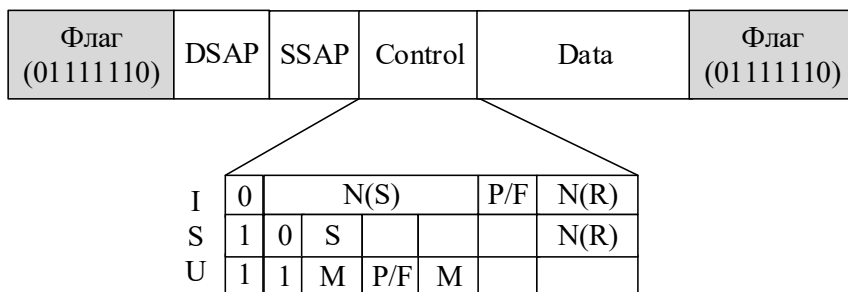
Через уровень LLC сетевой протокол запрашивает от уровня звена данных необходимую ему транспортную операцию с нужным качеством.

Протокол LLC передает свои кадры либо дейтаграммным способом, либо с помощью процедур с установлением соединения и восстановлением кадров.

Протокол LLC помещает пакет сетевого уровня (например, пакет IP, IPX) в свой кадр, который дополняется необходимыми служебными полями (рис. 3.3):

**DSAP** – *Destination Service Access Point* (адрес точки входа службы назначения), **SSAP** – *Source Service Access Point* (адрес точки входа службы источника) и **Control** (управляющее поле).

Через межуровневый интерфейс протокол LLC передает свой кадр вместе с адресной информацией об узле назначения соответствующему протоколу уровня MAC, который упаковывает кадр LLC в свой кадр.



**Рис. 3.3. Структура LLC-кадра стандарта 802-2**

Адресные поля *DSAP* и *SSAP* позволяют указать, какая служба верхнего уровня пересылает данные с помощью этого кадра.

Управляющее поле *Control* служит для реализации используемой в LAN версии высокоуровневого протокола управления звеном данных **HDLC (High-level Data Link Control)**, являющегося стандартом ISO.

Стандарт HDLC представляет собой обобщение нескольких близких стандартов **LAPx (Link Access Protocol)**, характерных для различных технологий:

**LAPB** в сетях X.25;

**LAPF** в сетях Frame Relay;

**LAPD** в сетях ISDN;

**LAPM** в сетях абонентского доступа на основе модемов.

В соответствии со стандартом 802.2 уровень управления логическим каналом LLC предоставляет верхним уровням следующие процедуры:

**LLC1** – процедура без установления соединения и подтверждения;

**LLC2** – процедура с установлением соединения и подтверждением;

**LLC3** – процедура без установления соединения, но с подтверждением.

Этот набор процедур является общим для всех методов доступа к среде, определенных стандартами 802.3–802.5, а также стандартами FDDI.

*Процедура без установления соединения и подтверждения LLC1* дает пользователю средства для передачи данных с минимумом издержек. Это дейтаграммный режим работы. Обычно он используется, когда такие функции как восстановление данных после ошибок и упорядочивание данных выполняются протоколами вышележащих уровней.

*Процедура с установлением соединений и подтверждением LLC2* дает пользователю возможность установить логическое соединение перед началом передачи любого блока данных и, если это требуется, выполнить процедуры восстановления после ошибок и упорядочивание потока этих блоков в рамках установленного соединения. Протокол LLC2 во многом аналогичен протоколам семейства HDLC (LAP-B, LAP-D, LAP-M), которые применяются в глобальных сетях для обеспечения надежной передачи кадров на зашумленных линиях.

Для случаев, когда временные издержки установления логического соединения перед отправкой данных неприемлемы, а подтверждение о коррект-

ности приема переданных данных необходимо, предусмотрена *процедура без установления соединения, но с подтверждением – LLC3*.

Чаще всего в локальных сетях используются протоколы LLC1, поскольку кабельные каналы локальных сетей обеспечивают низкую вероятность искажений бит и потери кадров. Например, когда принтеры подключаются к сети Ethernet непосредственно, с помощью встроенных сетевых адаптеров.

При сервисе без установления соединения функции уровня LLC сведены к минимуму – он используется только как интерфейс старших уровней к MAC-уровню. При использовании этого типа сервиса применяются только нумерованные блоки.

По своему назначению все кадры уровня LLC (называемые в стандарте 802.2 *протокольными блоками данных*, ПБД – *Protocol Data Unit, PDU*) подразделяются на три типа (рис. 3.4):

I – информационные (Information);

S – управляющие или супервизорные (Supervisory);

U – нумерованные (Unnumbered).

### 3.4. Архитектура и технологии построения сетей Ethernet

Типовая структура и элементы сети Ethernet показаны на рис. 3.1.

Ethernet – самый распространенный на сегодняшний день стандарт локальных сетей.

На основе стандарта Ethernet II был разработан стандарт IEEE 802.3.

В 1995 г. был принят стандарт ***Fast Ethernet***, который во многом не является самостоятельным стандартом. Его описание – дополнительный раздел к основному стандарту 802.3 – раздел 802.3u. Аналогично, принятый в 1998 г. стандарт ***Gigabit Ethernet*** описан в разделе 802.3z основного документа. А в июне 2002 г. комитетом IEEE 802 был принят новый стандарт ***10GE (10 Gigabit Ethernet)***, который уже является технологией построения не только локальных, но и глобальных сетей (на основе коммутируемых дуплексных волоконно-оптических каналов – ***DWDM (Dense Wave Division Multiplexing*** – плотное волновое мультиплексирование)).

Рассмотрим особенности названных стандартов и некоторых других популярных технологий локальных сетей.

#### 3.4.1. Ethernet. Стандарт IEEE 802.3

Стандарт IEEE 802.3 определяет топологию, тип физической среды, формат кадра, метод доступа к среде и метод передачи данных.

Топология Ethernet: шина, реже звезда. Сеть может «удлиниться» с помощью репитеров, если при максимальном расстоянии между АС затухание в кабеле слишком велико (рис. 3.1, а). «Общая шина» вместе с репитерами может быть уложена внутрь концентратора (хаба) (рис. 3.1, в). При этом пространственно сеть имеет радиальную структуру, а при использовании нескольких концентраторов – древовидную.

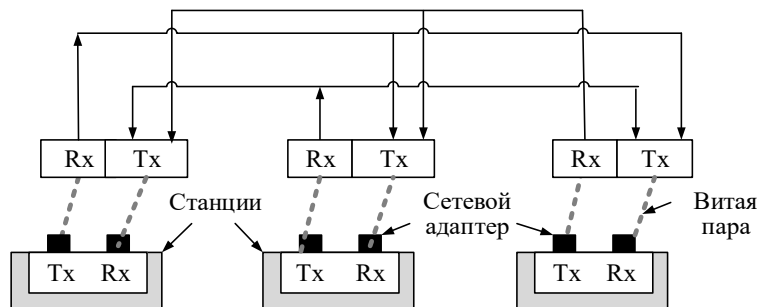
Для расширения сети или уменьшения нагрузки на общую физическую среду сеть Ethernet может разделяться на отдельные сегменты (*домены коллизий*) мостами и коммутаторами. Наряду с одноранговыми сетями, в которых все АС равны, могут строиться также сети с сервером, который может играть роль администратора сети, а также выполнять функции маршрутизатора, обеспечивающего подключение локальной сети через модем или выделенный цифровой канал к глобальной сети и удаленным локальным сетям (рис. 3.1, з).

Архитектура протоколов Ethernet 802.3 включает два подуровня: управления доступом к среде передачи (Physical Media Access, MAC) и управления логической передачей данных (Logical Link Control, LLC), физический уровень (Physical Media Access, PMA) и среду передачи.

В настоящее время в качестве кабельной физической среды используются следующие типы кабелей:

- неэкранированная витая пара (Unshielded Twisted Pair, UTP). На ее основе образуют звездообразную топологию с концентратором. Расстояние между концентратором и конечным узлом не более 100 м (рис. 3.4). Спецификация 10Base-T;
- оптоволоконный кабель. Топология аналогична стандарту на витой паре. Спецификация 10Base-F. Имеется несколько вариантов этой спецификации FOIRL (расстояние до 1000 м), 10Base-FL и 10Base-FB (расстояние до 2000 м).

В спецификации 10Base-T число 10 обозначает битовую скорость ПД – 10 Мбит/с, Base – метод передачи на базовой частоте 10 МГц, т. е. без модуляции.



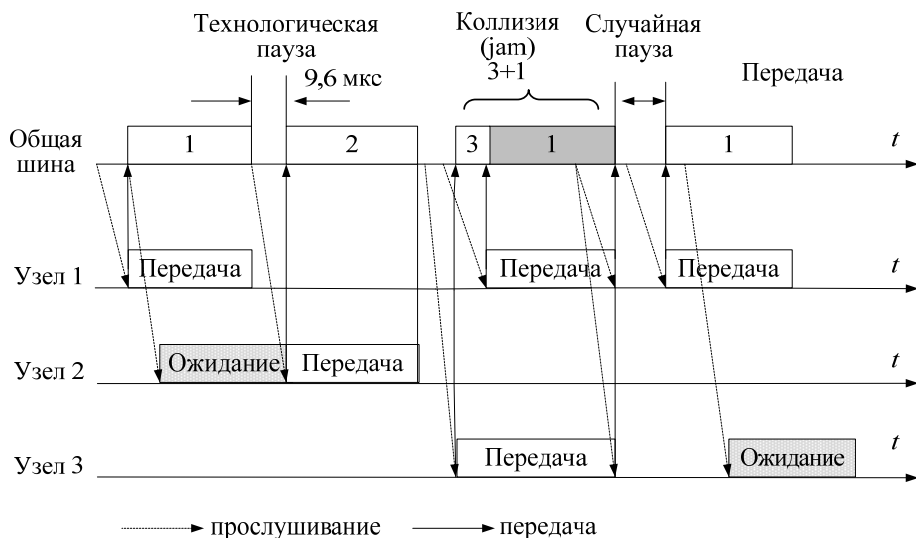
**Рис. 3.4. Сеть 10Base-T – один домен коллизий; Tx – передатчик, Rx – приемник**

Для передачи двоичной информации по кабелю для всех вариантов физического уровня технологии Ethernet используется манчестерский код, передача осуществляется кадрами (frame), скорость передачи – 10 Мбит/с. Принципы работы: на логическом уровне топология «Шина», все устройства (рабочие станции – PC) равноправны, данные, передаваемые одной PC, доступны всем остальным PC.

В технологии Ethernet независимо от применяемого стандарта физического уровня существует понятие домена коллизий. *Домен коллизий (collision*

*domain*) – это часть сети Ethernet, все узлы которой распознают коллизию независимо от того, в какой части этой сети она возникла. Сеть Ethernet, построенная на повторителях, всегда образует один домен коллизий. Он соответствует одной разделяемой среде. Мосты, коммутаторы и маршрутизаторы делят сеть Ethernet на несколько доменов коллизий.

Для доступа к среде передачи используется метод CSMA/CD (*carrier sense multiply access with collision detection*) – метод коллективного доступа с опознаванием несущей и обнаружением коллизий (рис. 3.5).



**Рис. 3.5. Метод случайного доступа CSMA/CD**

Этот метод применяется исключительно в сетях с логической общей шиной. Простота схемы доступа – один из факторов, определивших успех стандарта Ethernet. Кабель, к которому подключены все станции, работает в режиме *коллективного доступа (Multiply Access, MA)*.

Данные, передаваемые по сети, помещаются в кадры определенной структуры и снабжаются уникальным адресом станции назначения. Затем кадр передается по кабелю. Все станции, подключенные к кабелю, могут распознать факт передачи кадра. Та станция, которая узнает собственный адрес в заголовках кадра, записывает его содержимое в свой внутренний буфер, обрабатывает полученные данные и посылает по кабелю кадр-ответ. Адрес станции-источника также включен в исходный кадр, поэтому станция-получатель знает, кому нужно послать ответ.

Чтобы иметь право передавать кадр, АС должна убедиться, что разделяемая среда свободна. Признаком незанятости среды является отсутствие на ней несущей частоты, т. е. сигнала данных, уже передаваемых в данный момент какой-либо станцией.

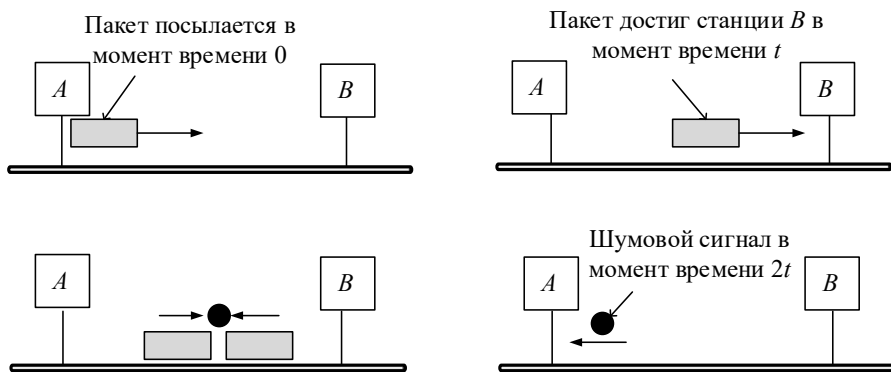
Механизм прослушивания среды и пауза между кадрами не гарантируют, что не возникнет ситуация, когда две или более станций одновременно решат,

что среда свободна, и начнут передавать свои кадры. Произойдет **коллизия** (*collision*): сигналы кадров от разных передающих станций сталкиваются на общем кабеле, происходит их искажение и, соответственно, искажение передаваемой информации. Методы кодирования, используемые в Ethernet, не позволяют выделять сигналы каждой станции из общего сигнала.

Коллизия – это нормальная ситуация в работе сетей Ethernet. Она возникает из-за того, что один узел начинает передачу раньше другого, но до второго узла сигналы первого просто не успевают дойти к тому времени, когда второй узел решает начать передачу своего кадра (рис. 3.6).

Если размер кадра будет слишком мал, отправитель закончит передачу прежде, чем получит шумовой сигнал, и будет считать, что его кадр успешно принят. Для предотвращения такой ситуации все кадры должны иметь такую длину, чтобы время их передачи было больше двойного оборота сигнала между крайними станциями.

Чтобы корректно обработать коллизию, все станции одновременно наблюдают за возникающими на кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется *обнаружение коллизии* (*collision detection, CD*). Для увеличения вероятности немедленного обнаружения коллизии всеми станциями сети, ситуация коллизии усиливается посылкой в сеть станциями, начавшими передачу своих кадров, специальной последовательности битов, называемой *jam-последовательностью*.



**Рис. 3.6. Схема возникновения коллизии в методе случайного доступа CSMA/CD ( $t$  – задержка распространения сигнала между станциями A и B)**

После этого обнаружившая коллизию передающая станция обязана прекратить передачу и сделать паузу в течение короткого случайного интервала времени. Затем она может снова предпринять попытку захвата среды и передачи кадра. Случайная пауза  $T_{\text{пауза}}$  выбирается по следующему правилу:

$$T_{\text{пауза}} = LT_{\text{отср}},$$

где  $T_{\text{отср}} = 512bt$  – интервал отсрочки (slot time) ( $bt$  – битовый интервал, соответствующий времени между появлением двух последовательных бит данных)



на кабеле; для скорости 10 Мбит/с битовый интервал равен 0,1 мкс или 100 нс);  $L$  – случайное число, выбираемое с равной вероятностью из диапазона  $[0, 2^N]$ ,  $N = \min(10, \Pi)$ ,  $\Pi \leq 16$ . Если 16 последовательных попыток ( $\Pi$ ) передачи кадра вызывают коллизию, то передатчик должен прекратить попытки и отбросить этот кадр.

**Параметры Ethernet.** Все параметры протокола Ethernet подобраны таким образом, чтобы при нормальной работе узлов сети коллизии всегда четко распознавались.

Для надежного распознавания коллизий должно выполняться следующее соотношение:

$$T_{\min} \geq PDV, \quad (3.1)$$

где  $T_{\min}$  – время передачи кадра минимальной длины; PDV (Path Delay Value) – время, за которое сигнал коллизии успевает распространиться до самого дальнего узла сети.

Так как в худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга станциями сети (в одну сторону проходит неискаженный сигнал, а на обратном пути распространяется уже искаженный коллизией сигнал), то это время называется временем двойного оборота, или окном коллизий.

Минимальная длина поля данных кадра должна быть не менее 46 байт (что вместе со служебными полями дает минимальную длину кадра в 72 байта, или 576 бит).

Длина кабельной системы выбирается так, чтобы за время передачи кадра **минимальной длины** сигнал коллизии успел бы распространиться до самого дальнего узла сети. Поэтому для скорости передачи данных 10 Мбит/с, используемой в стандартах Ethernet, а также с учетом затухания в кабеле и задержек в репитерах в реализациях Ethernet разработчики ограничили максимальное количество сегментов в сети пятью, с длиной каждого не более 500 м. Последнее определило общую длину сети 2500 м.

После окончания передачи кадра все узлы сети обязаны выдержать *технологическую паузу* (*IPG – Inter Packet Gap*) в 9,6 мкс. Эта пауза, называемая также *межкадровым интервалом*, нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одной станцией.

Параметры Ethernet сведены в табл. 3.2.

**Таблица 3.2**

**Параметры Ethernet**

Параметр	Значение
Битовая скорость	10 Мб/с
Интервал отсрочки	512 bt
Межкадровый интервал	9,6 мкс
Максим. число попыток передачи	16
Максимальное число возрастания диапазона паузы	10

Продолжение табл. 3.2

Параметр	Значение
Длина jam-последовательности	32 бита
Максимальная длина кадра (без преамбулы)	1518 байтов
Минимальная длина кадра (без преамбулы)	64 байта (512 бит)
Длина преамбулы	64 бита
Минимальное расстояние между узлами	2 м

Максимально возможная пропускная способность Ethernet

$$V = \frac{1}{(57,6 + 9,6)10^{-6}} = 14\,880 \text{ кадр/с.}$$

Сети Ethernet должны удовлетворять двум ограничениям, связанным с методом доступа:

- максимальное расстояние между двумя любыми узлами не должно превышать 2500 м;

- в сети не должно быть более 1024 узлов.

Полезная пропускная способность протокола CSMA/CD:

- для кадров минимальной длины  $C_{\text{п}} = 14880 \cdot 46 \cdot 8 = 5.48 \text{ Мбит/с}$ ;

- для кадров максимальной длины  $C_{\text{п}} = 813 \cdot 1500 \cdot 8 = 9.76 \text{ Мбит/с}$ .

Использование Ethernet в локальных сетях показывает, что она является самой недорогой и эффективной сетевой технологией.

### 3.4.2. Fast Ethernet как развитие классического Ethernet'a

Осенью 1995 г. комитет IEEE 802.3 принял спецификацию Fast Ethernet в качестве стандарта 802.3u, который является дополнением к существующему стандарту 802.3.

Уровни MAC и LLC в Fast Ethernet остались теми же. Все отличия технологии Fast Ethernet от Ethernet сосредоточены на физическом уровне. Обеспечить скорость передачи 100 Мбит/с сложнее, чем 10 Мбит/с.

Коаксиальный кабель не применяется. На небольших расстояниях витая пара категории 5 позволяет передавать данные с той же скоростью, что и коаксиальный кабель, но сеть дешевле и удобней в эксплуатации.

На больших расстояниях оптическое волокно обладает гораздо более широкой полосой пропускания, чем коаксиальный кабель. Стоимость сети ненамного выше.

В Fast Ethernet меняется как количество проводников, так и методы кодирования. В Fast Ethernet детально определены подуровни физического уровня, зависящие и не зависящие от физической среды.

Официальный стандарт 802.3u установил для разных сред передачи три различных спецификации физического уровня Fast Ethernet:

- 100Base-TX – для двух пар кабеля (прием/передача) UTP (Unshielded Twisted Pair – неэкранированная витая пара) категории 5 или STP (Shielded

Twisted Pair – экранированная витая пара) Type 1 (используемые коды – 4В/5В и MLT-3);

- 100Base-T4 – для четырех пар кабеля UTP категории 3, 4 или 5;
- 100Base-FX – для двух волокон (прием/передача) многомодового оптического кабеля.

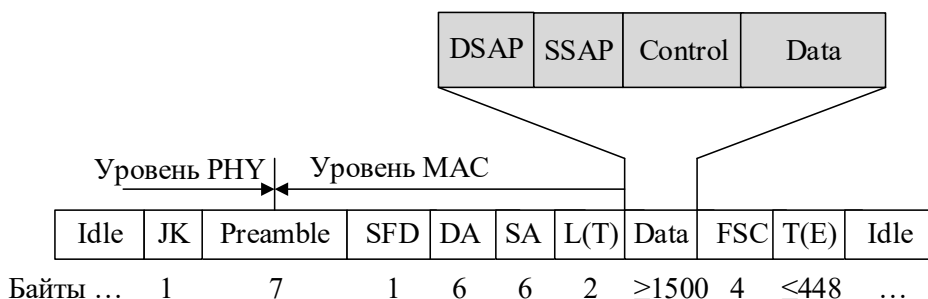
Для передачи данных используется метод кодирования 4В/5В: 4-битовый код представляется 5-битовым кодом. Коды 4В/5В построены так, что гарантируют не более трех нулей подряд при любом сочетании бит в исходной информации. За счет достаточно частой смены знака передаваемых бит обеспечивается синхронизация приемника с передатчиком. Избыточный бит вынуждает передавать биты кода 4В/5В со скоростью 125 Мб/с.

Спецификация 100Base-T4 была разработана для того, чтобы можно было использовать имеющуюся проводку (в частности, телефонную) на витой паре категории 3.

Вместо кодирования 4В/5В в этом методе используется кодирование 8В/6Т. Каждые 8 бит кодируются 6 троичными сигналами. Группа из 6 троичных цифр (посылок) передается на одну из трех передающих витых пар независимо и последовательно. Четвертая пара используется для прослушивания несущей частоты в целях обнаружения коллизии. Скорость передачи данных по каждой из трех передающих пар равна 33.3 Мб/с, поэтому общая скорость протокола 100Base-T4 составляет 100 Мб/с. В то же время из-за принятого способа кодирования скорость изменения сигнала на каждой паре равна всего 25 Мбод, что и позволяет использовать витую пару категории 3.

Уровни MAC и LLC в стандарте Fast Ethernet не претерпели изменений. Их описывают стандарты 802.3 и 802.2. Подуровень LLC обеспечивает интерфейс протокола Ethernet с протоколами вышележащих уровней, например, с IP или IPX. Подуровень MAC ответственен за формирование кадра Ethernet, получение доступа к разделяемой среде ПД и за отправку кадра по физической среде узлу назначения. MAC-подуровень каждого узла сети получает от физического уровня информацию о состоянии разделяемой среды.

Формат кадров технологии Fast Ethernet (рис. 3.7) почти не отличается от формата кадров технологий 10-мегабитного Ethernet.



**Рис. 3.7. Формат кадра Fast (Giga) Ethernet**

Отличие проявляется в использовании перед началом кадра комбинации символов Start Delimiter (пара символов J (11000) и K (10001) кода 4B/5B (применяемого в 100Base-FX/TX), а после завершения кадра символа T (End Delimiter). Данные служебные поля нужны для отделения кадра от постоянной заполняющей последовательности символов Idle (не используемой в сети Ethernet 10 Мбит/с, в которой общая среда должна в паузах «молчать»).

Сети Fast Ethernet имеют иерархическую древовидную структуру, построенную на концентраторах. Основным отличием конфигураций сетей Fast Ethernet является сокращение диаметра сети примерно до 200 м, что объясняется уменьшением времени передачи кадра минимальной длины в 10 раз за счет увеличения скорости передачи в 10 раз.

При использовании коммутаторов протокол Fast Ethernet может работать в полнодуплексном режиме, в котором нет ограничений на общую длину сети, а остаются только ограничения на длину физических сегментов, соединяющих соседние устройства (адаптер – коммутатор или коммутатор – коммутатор). Поэтому при создании магистралей локальных сетей большой протяженности технология Fast Ethernet активно применяется в полнодуплексном варианте на основе коммутаторов.

### 3.4.3. Протокол Gigabit Ethernet

Стандарт IEEE 802.3z (1998 г.). Применяется та же стратегия, что и в сети Fast Ethernet: сохранены протокол CSMA/CD и формат кадра.

Изменен носитель. Спецификация IEEE 802.3z включает варианты:

- 1000 BASE-LX. Длинноволновый вариант: оптоволоконный многомодовый кабель  $\varnothing$  62,5  $\mu$ м или 50  $\mu$ м длиной до 550 м или одномодовый кабель  $\varnothing$  10  $\mu$ м длиной до 5 км; дуплексные линии;
- 1000 BASE-SX. Коротковолновый вариант: оптоволоконный одномодовый кабель  $\varnothing$  62,5  $\mu$ м длиной до 275 м или  $\varnothing$  50  $\mu$ м длиной до 550 м; дуплексные линии;
- 1000 BASE-CX. Специализированные экранированные кабели из витых пар протяженностью не более 25 м (медные перемычки). Каждая линия состоит из отдельной витой пары, по которой данные передаются в обе стороны;
- 1000 BASE-T. Используются 4 неэкранированные витые пары категории 5 для связи с устройствами до 100 м.

25 м – неприемлемо малая длина. Применяют пакетную передачу кадров – аппаратное решение «расширение носителя» до 512 байт, что увеличивает максимальную длину сегмента до 200 м.

Кодирование носителя 8B/10B. Правила выбора кодовых слов:

- кодовые слова не должны иметь более 4-х одинаковых битов подряд;
- в кодовых словах не должно быть более 6 нулей или 6 единиц.

Почему так?

Во-первых, достаточное количество изменений в потоке данных обеспечивает синхронизацию приемника с передатчиком;

Во-вторых, сбалансированному количеству нулей и единиц удастся держать постоянную составляющую сигнала на как можно более низком уровне – тогда она (постоянная составляющая) проходит через преобразователи без изменений.

Предусмотрен контроль потока – посылка специального служебного кадра, сообщающего о том, что передающей стороне необходимо приостановиться.

Непрекращающийся рост трафика способствовал появлению 10-гигабитной сети Ethernet (стандарт IEEE 802.3 ac).

Вначале 10-гигабитную сеть Ethernet использовали в качестве ЛС. По мере увеличения спроса на широкополосную связь технология 10-гигабитной сети Ethernet распространяется на серверные фермы, магистрали и ЛС, охватывающие комплексы зданий. Эта технология позволяет создавать региональные сети, соединяющие удаленные ЛС.

### 3.5. Стандарт Token Ring

Стандарт Token Ring был принят комитетом IEEE 802.5 в 1985 году.

Передающая среда – сегменты экранированной или неэкранированной витой пары, или оптоволокну, сопрягаемые в кольцо – общий разделяемый ресурс. Право на доступ к кольцу передается с помощью кадра специального формата, называемого *маркером или токеном*. В любой момент времени только одна станция в сети обладает правом доступа. Доступ передается циклически по логическому кольцу. Это *детерминированный* алгоритм доступа.

На рис. 3.8 доступ к среде иллюстрируется временной диаграммой, на рисунке показана передача пакета *A* в кольце от станции 1 к станции 3.

Получив маркер, станция анализирует его, при необходимости модифицирует и при отсутствии у нее данных для передачи транслирует к следующей станции. Станция, имеющая данные для передачи, при получении маркера изымает его из кольца, затем выдает в кольцо кадр данных установленного формата последовательно по битам. Переданные данные проходят по кольцу всегда в одном направлении от одной станции к другой.

При поступлении кадра данных к адресуемым одной или несколькими станциями эти станции копируют для себя этот кадр и вставляют в этот кадр подтверждение приема (либо сообщение принято, либо принято с ошибкой, либо станция для приема не доступна). Станция, выдавшая кадр данных в кольцо, при обратном его получении с подтверждением приема изымает этот кадр из кольца и возвращает маркер для возможности другим станциям сети передавать данные. Время удержания одной станцией маркера ограничивается *тайм-аутом удержания маркера*, после истечения которого станция обязана передать маркер далее по кольцу.

Сети Token Ring работают с двумя битовыми скоростями – 4 Мб/с и 16 Мб/с. Сети Token Ring, работающие со скоростью 16 Мб/с, имеют некоторые усовершенствования в алгоритме доступа по сравнению со стандартом 4 Мб/с (алгоритмом *раннего освобождения маркера*). Смещение станций, работающих на различных скоростях, в одном кольце не допускается.



**Прерывающая последовательность** состоит из двух байтов, содержащих начальный ограничитель и конечный ограничитель. Прерывающая последовательность может появиться в любом месте потока битов и сигнализирует о том, что текущая передача кадра или маркера отменяется.

В сети Token Ring на уровнях MAC и LLC применяются процедуры без установления соединения, но с подтверждением получения кадров.

В конфигурации выделяют станции двух типов:

- станции, подключаемые к кольцу через концентратор. Обычно такими станциями являются компьютеры с установленными в них сетевыми адаптерами. Концентраторы Token Ring делятся на активные и пассивные. Активные концентраторы поддерживают большие расстояния до станции, чем пассивные.

- станции сети, соединенные в кольцо непосредственными связями, называются магистральными (trunk cable). Обычно связи такого рода используются для соединения концентраторов друг с другом для образования общего кольца. Порты концентраторов, предназначенные для такого соединения, называются портами Ring-In и Ring-Out.

Кроме экранированной витой пары существуют сетевые адаптеры и концентраторы Token Ring, поддерживающие неэкранированную витую пару и оптоволокно.

Максимальное количество станций в одном кольце – 250.

Максимальная длина кольца Token Ring составляет 4000 м. Ограничения на максимальную длину кольца и количество станций в кольце в технологии Token Ring не являются такими жесткими, как в технологии Ethernet. Здесь эти ограничения во многом связаны со временем оборота маркера по кольцу. Однако сети Token Ring можно настраивать, что позволяет построить сеть Token Ring с большим количеством станций и с большей длиной кольца.

### 3.6. Стандарт FDDI

Технология *Fiber Distributed Data Interface* – первая технология локальных сетей, которая использовала в качестве среды передачи данных оптоволоконный кабель (1986÷1988 гг). Тогда же появилось первое оборудование – сетевые адаптеры, концентраторы, мосты и маршрутизаторы, поддерживающие этот стандарт.

FDDI остается наиболее отработанной высокоскоростной технологией.

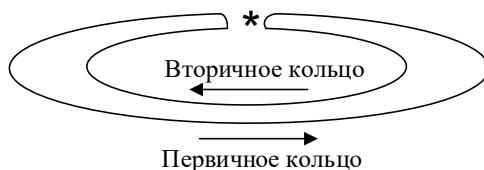
Технология FDDI во многом основывается на технологии Token Ring. Разработчики технологии FDDI ставили перед собой следующие цели:

- повысить битовую скорость передачи данных до 100 Мб/с;
- повысить отказоустойчивость сети после отказов различного рода;
- максимально эффективно использовать пропускную способность сети как для асинхронного, так и для синхронного трафика.

Сеть FDDI строится на основе двух оптоволоконных колец, которые образуют основной и резервный пути передачи данных между узлами сети. Узлы должны быть подключены к обоим кольцам. В нормальном режиме работы се-

ти данные проходят через все узлы и все участки кабеля первичного (Primary) кольца, поэтому этот режим назван режимом Thru – «сквозным» или «транзитным». Вторичное кольцо (Secondary) в этом режиме не используется.

В случае какого-либо вида отказа (например, обрыв кабеля или отказ узла), первичное кольцо объединяется со вторичным (рис. 3.9), образуя вновь единое кольцо. Этот режим работы сети называется Wrap, то есть «свертывание» или «сворачивание» колец.



**Рис. 3.9. Реконфигурация колец FDDI при отказе**

Операция свертывания производится силами концентраторов и/или сетевых адаптеров FDDI. Данные по первичному кольцу всегда передаются против часовой стрелки, а по вторичному – по часовой. Сеть FDDI может полностью восстанавливать свою работоспособность в случае единичных отказов ее элементов. При множественных отказах сеть распадается на несколько не связанных сетей.

Кольца в сетях FDDI рассматриваются как общая разделяемая среда передачи данных. Метод очень близок к методу доступа сетей Token Ring и также называется методом маркерного (или токенового) кольца – Token ring.

В сети FDDI у каждой станции есть предшествующий сосед (upstream neighbour) и последующий сосед (downstream neighbour), определяемые ее физическими связями и направлением передачи информации.

Если станция захватила токен и передает свои собственные кадры, то на протяжении этого периода времени она не транслирует приходящие кадры, а удаляет их из сети.

Если же адрес кадра совпадает с адресом станции, то она копирует кадр в свой внутренний буфер, проверяет его корректность (в основном, по контрольной сумме), передает его поле данных для последующей обработки протоколу, лежащего выше FDDI уровня (например, IP), а затем передает исходный кадр по сети последующей станции. В передаваемом в сеть кадре станция назначения отмечает три признака: *распознавания адреса, копирования кадра и отсутствия или наличия в нем ошибок*.

После этого кадр продолжает путешествовать по сети, транслируясь каждым узлом. Станция, являющаяся источником кадра для сети, ответственна за то, чтобы удалить кадр из сети, после того как он, совершив полный оборот, вновь дойдет до нее. При этом исходная станция проверяет признаки кадра, дошел ли он до станции назначения и не был ли при этом поврежден. Процесс восстановления информационных кадров не входит в обязанности протокола FDDI, этим должны заниматься протоколы более высоких уровней.



### 3.7. Технология Fibre Channel

Сеть представляет собой набор точек доступа с программной структурой протоколов, обеспечивающей обмен данными. Элементами сети **Fibre Channel** являются конечные элементы, называемые узлами (N – nodes), и набор коммуникационных элементов, называемый каркасом (F – fabric (рис. 3.10)).

Интерфейс **Fibre Channel** разработан для объединения лучших качеств обеих технологий – простоты и скорости каналов ввода-вывода и гибкости и взаимосвязанности сетевых технологий. Канал ввода-вывода – прямая двухточечная линия связи, аппаратно реализованная, высокая скорость ПД на короткие расстояния.

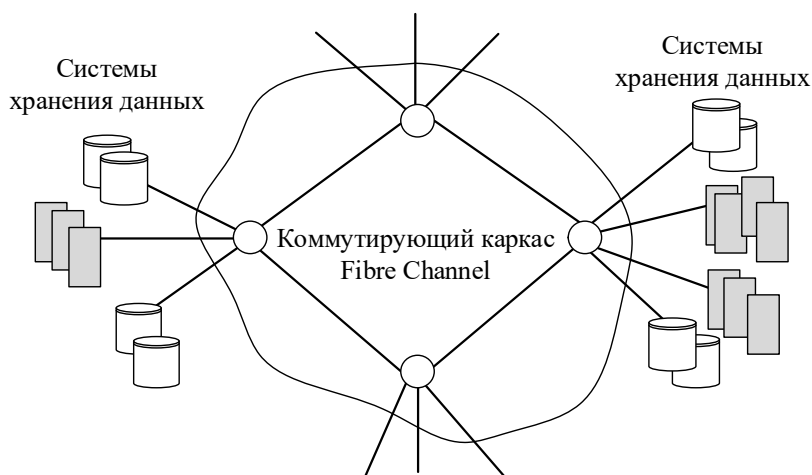


Рис. 3.10. Сеть Fibre Cannel

Элементы соединены двухточечными линиями между портами индивидуальных узлов и коммутаторов. Взаимодействие состоит из передачи кадров по двунаправленным двухточечным линиям между портами.

Вся маршрутизация кадров между узлами осуществляется каркасом. Каркас может буферизировать кадры, что позволяет узлам общаться на разных скоростях ПД. Сеть Fibre Channel основана на коммутирующей сети (не возникает вопроса доступа к несущей). Сеть легко масштабируется.

Архитектура протоколов FCh включает 5 уровней.

**FCh-0.** Физический носитель – оптоволоконный кабель, ПД на большие расстояния; коаксиальный кабель для высоких скоростей на короткие расстояния; экранированная витая пара для низких скоростей и коротких расстояний.

Скорости ПД от 100 Мбит/с до 3,2 Гбит/с. Расстояния от 33 м до 10 км.

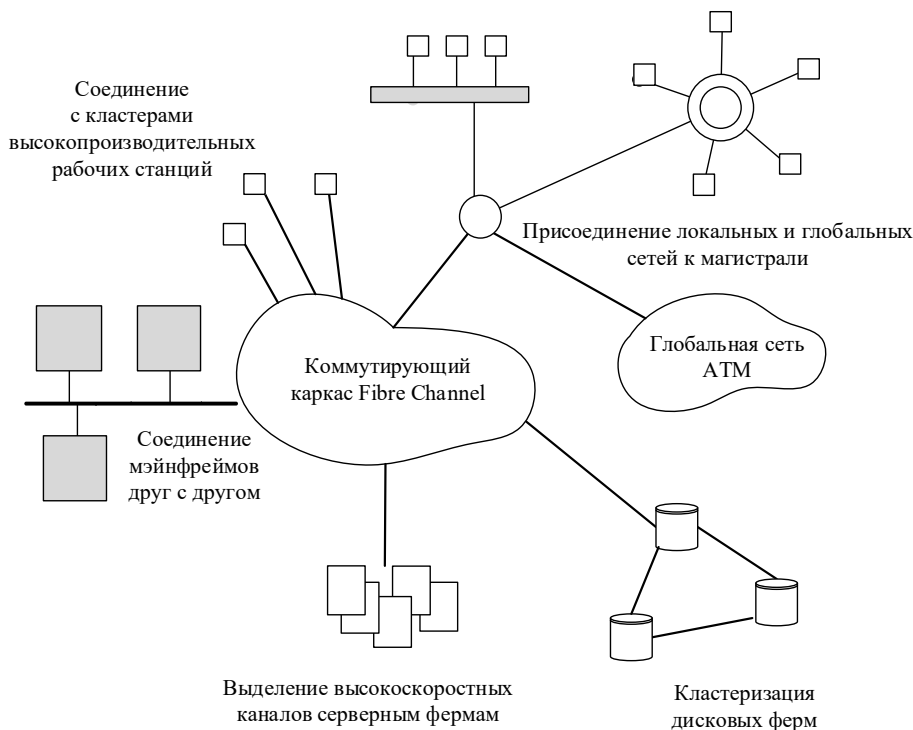
**FCh-1.** Протокол ПД. Определяет схему кодирования сигнала (NRZI – потенциальный код с инверсией при единице).

**FCh-2.** Кадровый протокол. Имеет дело с определением форматов кадров, управлением потоком, контролем ошибок, группированием кадров в логические объекты, называемые последовательностями и обменами.

**FCh-3. Общие службы.** Сюда относят групповую рассылку.

**FCh-4. Отображение.** Определяет отображение на протоколы Fibre Channel различных канальных и сетевых протоколов, включая IEEE802, ATM, IP и интерфейс SCSI (Small Computer Systems Interface – интерфейс малых компьютерных систем).

Технологии, носители данных и скорости ПД могут комбинироваться, формируя оптимальную конфигурацию для данного сайта. На рис 3.11 показан пример применения сети Fibre Channel в качестве сети хранения данных SAN (Storage Area Network).



**Рис. 3.11. Пять вариантов применения сети Fibre Channel**

Fibre Channel все чаще применяется в качестве опорной инфокоммуникационной сети.

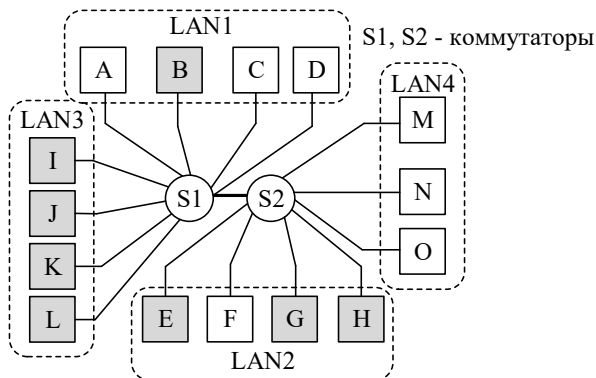
### **3.8. Виртуальные локальные сети**

Виртуальной сетью называется группа узлов сети, трафик которой, в том числе и широковещательный, на канальном уровне полностью изолирован от других узлов сети. Это означает, что передача кадров между разными виртуальными сегментами на основании адреса канального уровня невозможна, независимо от типа адреса – уникального, группового или широковещательного. В то же время внутри виртуальной сети кадры передаются по технологии

коммутации, то есть только на тот порт, который связан с адресом назначения кадра.

Виртуальная сеть образует домен широковещательного трафика (broadcast domain).

Назначение технологии виртуальных сетей состоит в облегчении процесса создания независимых сетей, которые затем должны связываться с помощью протоколов сетевого уровня.



**Рис. 3.12. Четыре физические LAN объединены в две VLAN**

При использовании технологии VLAN решаются две задачи:

- повышение производительности в каждой из виртуальных сетей, так как коммутатор передает кадры в такой сети только узлу назначения;
- изоляция сетей друг от друга для управления правами доступа пользователей и создания защитных барьеров на пути широковещательных штормов.

Приписывание отдельного порта к любому из внутренних сегментов производится *программным путем*. Программное приписывание порта сегменту часто называют *статической* или *конфигурационной* коммутацией.

При изменении состава сегментов (переход пользователя в другую сеть, дробление крупных сегментов) при обычном подходе приходится производить физическую перекоммутацию разъемов на передних панелях повторителей или в кроссовых панелях. Это не очень удобно в больших сетях – много физической работы, к тому же высока вероятность ошибки.

Поэтому для устранения необходимости физической перекоммутации узлов стали применять коммутаторы и конфигурационную коммутацию.

Чтобы VLAN функционировали корректно, необходимо *наличие конфигурационных таблиц*.

Существует несколько способов построения виртуальных сетей:

- группировка портов;
- группировка MAC-адресов;
- использование меток в дополнительном поле кадра – частные протоколы и спецификации IEEE 802.1 Q/p.

**VLAN на базе портов.** Используется механизм группирования портов коммутатора. При этом каждый порт приписывается той или иной виртуальной сети. Кадр, пришедший от порта, принадлежащего одной виртуальной сети, никогда не будет передан порту, который не принадлежит этой виртуальной сети. В некоторых коммутаторах один порт можно приписать нескольким виртуальным сетям.

Достоинство: простота настройки (достаточно каждому порту, находящемуся в одной VLAN, присвоить один и тот же идентификатор VLAN ID). Возможность изменения логической топологии сети без физического перемещения.

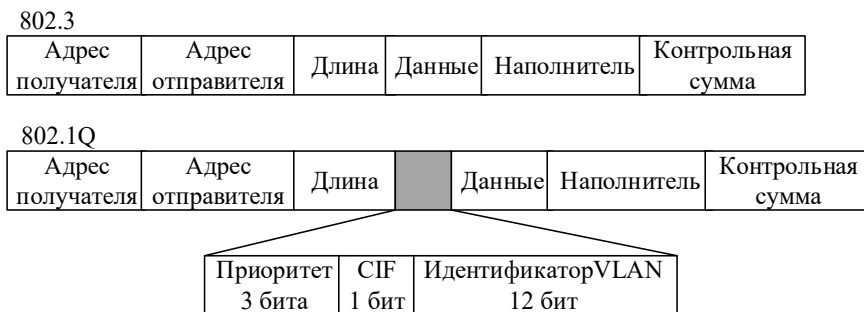
**VLAN на базе MAC-адресов.** Способ основан на группировании MAC-адресов. Каждый MAC-адрес, который изучен коммутатором, приписывается той или иной виртуальной сети. При существовании в сети множества узлов этот способ требует выполнения большого количества ручных настроек от администратора сети. Однако при построении виртуальных сетей на основе нескольких коммутаторов он оказывается более гибким, чем группирование портов.

**VLAN на основе меток в дополнительном поле кадра (стандарт IEEE 802.1Q).**

Стандарт IEEE 802.1Q определяет изменения в структуре кадра Ethernet, позволяющие передавать информацию о VLAN по сети.

К кадру Ethernet добавлены два байта. Эти 16 бит содержат информацию по принадлежности кадра Ethernet к локальной сети и о его приоритете. Три бита кодируется до восьми уровней приоритета, 12 бит позволяют различать трафик большого числа (до 4096) LAN, а один бит зарезервирован для обозначения кадров сетей других типов (Token Ring, FDDI), передаваемых по магистрали Ethernet (рис. 3.13).

Добавление двух байтов к максимальному размеру кадра Ethernet ведет к возникновению проблем в работе многих коммутаторов, обрабатывающих кадры Ethernet аппаратно. Чтобы избежать их, группы по стандартизации предложили сократить на два байта максимальный размер полезной нагрузки в кадре.



**CIF (Canonical Format Indicator)** – индикатор классического формата зарезервирован для обозначения кадров сетей других типов (Token Ring, FDDI), передаваемых по магистрали Ethernet

**Рис. 3.13. Форматы кадров Ethernet (802.3) и стандарта IEEE 802.1Q**

Так как максимальный размер кадра Ethernet остался неизменным, то пакеты всех VLAN могут обрабатываться традиционными коммутаторами и маршрутизаторами внутренней части сети.

Назначение технологии виртуальных сетей состоит в облегчении процесса создания независимых сетей, которые затем должны связываться с помощью протоколов сетевого уровня.

Отметим, технологии виртуальных сетей широко используется при построении корпоративных сетей.

### **Контрольные вопросы**

1. Чем отличается физическая топология локальной сети от логической?
2. Каковы функции MAC-уровня? Уровня LLC ?
3. Как пакет сетевого уровня передается через сеть Ethernet?
4. В чем состоят функции преамбулы и начального ограничителя кадра в стандарте Ethernet?
5. Какие процедуры уровень управления LLC предоставляет верхним уровням?
6. Из каких соображений выбирается длина кабельной системы в технологии Ethernet? Длина сегментов сети?
7. Что такое домен коллизий? Почему коллизия возникает и где?
8. Из каких соображений выбрана максимальная длина физического сегмента в стандартах Ethernet?
9. Если один вариант технологии Ethernet имеет более высокую скорость передачи данных, чем другой, то какой из них поддерживает большую максимальную длину сети и почему?
10. С какой целью в стандарте Gigabit Ethernet применено решение «расширение носителя» до 512 байт?
11. Имеет ли место случайность во времени доступа в технологии Token Ring и если да, то в чем она проявляется?
12. Чем отличаются режимы работы сети Token Ring на скорости 4 Мб/с и 16 Мб/с?
13. Чем обеспечивается повышенная отказоустойчивость технологии FDDI?
14. Чем отличается доступ к среде передачи в технологии Ethernet от технологии Fibre Channel?
15. Какова цель создания виртуальных локальных сетей?

## 4. ГЛОБАЛЬНЫЕ СЕТИ

Глобальные компьютерные сети «покрывают» территории государства или нескольких государств, к примеру, всемирная сеть Internet.

В таких объединенных сетях для управления обменом сообщениями между пользователями применяются различные протоколы.

Поскольку в Internet доминирует TCP/IP, а сеть Internet широкого пользования, то рассмотрим особенности стека протоколов TCP/IP, а в Приложении 1 приведем краткие характеристики архитектур сетей X.25, ISDN, Frame Relay, АТМ.

Функциональную организацию глобальной сети удобно представить, используя соответствующую функциональную модель.

### 4.1. Функциональная модель глобальной сети

Взаимодействие удаленных процессов лежит в основе функционирования ИС. Каждый компьютер в сети работает под управлением своей локальной операционной системы (ОС). Такая ОС отличается от операционной системы автономного компьютера наличием дополнительных сетевых средств (программной поддержкой для сетевых интерфейсных устройств и доступа к удаленным ресурсам).

Во многих случаях информация между удаленными процессами в сети передается не напрямую, а через ряд процессов-посредников (рис. 4.1). Процессы-посредники выполняются на вычислительных комплексах, которые не являются компьютерами отправителя и получателя. Эти комплексы работают под управлением собственных операционных систем, но и при отсутствии процессов-посредников удаленные процесс-отправитель и процесс-получатель часто функционируют под управлением различных ОС.

*Взаимодействие удаленных процессов* сводится к обмену сообщениями. Промежуток времени, в течение которого взаимодействуют процессы, называется *сеансом* (сессией).

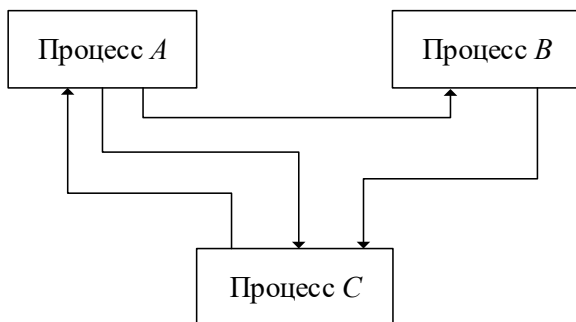


Рис. 4.1. Взаимодействие процессов

**Адресация в сети.** При организации взаимодействия удаленных процессов участники этого взаимодействия должны иметь *уникальные адреса* уже в

рамках всей сети. Практически каждый сетевой пакет информации должен быть снабжен адресами получателя и отправителя.

Полный сетевой адрес процесса или промежуточного объекта для хранения данных складывается из двух частей: адреса вычислительного комплекса, на котором находится процесс или объект в сети (удаленного адреса), и адреса самого процесса или объекта на этом вычислительном комплексе (локального адреса). Уникальность полного адреса будет обеспечиваться уникальностью удаленного адреса для каждого компьютера в сети и уникальностью локальных адресов объектов на компьютере.

**Локальная адресация. Понятие порта.** При удаленной связи для локальной адресации процессов и промежуточных объектов организуется новое специальное адресное пространство.

Каждый процесс, желающий принять участие в сетевом взаимодействии, после рождения закрепляет за собой один или несколько адресов в этом адресном пространстве.

Каждому промежуточному объекту при его создании присваивается свой адрес из этого адресного пространства.

Подобные адреса получили название портов. *Порт* – это логические (программно-организованные) точки, через которые производится ввод сообщений в процесс и их вывод из него.

Таким образом, процесс как объект представляется совокупностью портов, через которые он взаимодействует с другими процессами.

При получении данных от удаленного процесса операционная система «смотрит», на какой порт они были отправлены, и определяет процесс, который заявил этот порт в качестве своего адреса, или объект, которому присвоен данный адрес, и доставляет полученную информацию адресату.

Виды адресного пространства портов (т. е. способы построения локальных адресов) определяются, как правило, *протоколами транспортного уровня*.

**Полные адреса. Понятие сокета (socket).** Полный адрес удаленного процесса или промежуточного объекта определяется парой адресов: *⟨числовой адрес компьютера в сети, порт⟩*. Подобная пара получила наименование *socket* (*сокет*) («гнездо»), а сам способ их использования называется организацией связи с помощью сокетов.

В случае *непрямой адресации* с использованием промежуточных объектов сами эти объекты также принято называть сокетами. Поскольку разные протоколы транспортного уровня требуют разных адресных пространств портов, то для каждой пары надо указывать, какой транспортный протокол она использует, т. е. тип сокетов.

**Удаленная адресация и разрешение адресов.** Инициатором связи процессов друг с другом всегда является человек. Человеку свойственно думать словами, он легче воспринимает символическую информацию. Поэтому каждый компьютер в сети получает символическое, часто даже содержательное имя.

Компьютеру проще оперировать числами, желательно одного и того же формата, поэтому каждый компьютер в сети получает *числовой адрес*. Способы

построения удаленных адресов обычно определяются *протоколами сетевого уровня* эталонной модели.

Возникает проблема отображения пространства символьных имен (или адресов) вычислительных комплексов в пространство их числовых адресов. Она получила наименование *проблемы разрешения адресов*. Чтобы установить связь между двумя идентификаторами хоста – именем и числовым адресом, используется *система доменных имен* (Domain Name System, DNS).

*DNS* – это база данных, распределенная между иерархически структурированными серверами имен, а также протокол прикладного уровня, организующий взаимодействие между хостами и серверами имен для выполнения операции преобразования. Протокол DNS работает поверх транспортного протокола UDP.

Так решается проблема удаленных адресов, т. е. проблема получения числового удаленного адреса нужного компьютера.

**Проблемы маршрутизации в сетях.** Одна из отличительных особенностей взаимодействия удаленных процессов состоит в использовании в большинстве случаев *процессов-посредников*, расположенных на аппаратно-программных комплексах, не являющихся комплексами отправителя и получателя.

Возникают вопросы: как организовать работу операционных систем на комплексах – участниках связи (это могут быть конечные или промежуточные узлы) для определения маршрута передачи данных? По какой из нескольких линий связи (или через какой сетевой адаптер) нужно отправить пакет информации? Для решения этих проблем применяются протоколы одношаговой маршрутизации.

При *одношаговой маршрутизации* каждый компонент сети, принимающий участие в передаче информации, самостоятельно определяет, какому следующему компоненту, находящемуся в зоне прямого доступа, она должна быть отправлена. Решение принимается на основании анализа содержащегося в пакете адреса получателя. Полный маршрут передачи данных складывается из одношаговых решений, принятых компонентами сети.

Для работы *алгоритмов одношаговой маршрутизации* на каждом компоненте сети, имеющем возможность передавать информацию более чем одному компоненту, обычно строится специальная *таблица маршрутов*.

В простейшем случае каждая запись такой таблицы содержит адрес адресуемого узла получателя; адрес компонента сети, напрямую подсоединенного к данному и которому следует отправить пакет, предназначенный для этого получателя; указание линии связи (сетевого адаптера), по которой должен быть отправлен пакет.

По способам формирования и использования таблиц маршрутизации алгоритмы одношаговой маршрутизации можно разделить на два класса:

- алгоритмы фиксированной маршрутизации;
- алгоритмы динамической и адаптивной маршрутизации.

При *фиксированной маршрутизации* все записи в таблице являются статическими. Обычно линии выбирают так, чтобы минимизировать полное время



доставки данных. Преимуществом этой стратегии является простота реализации. Однако при отказе выбранной линии связи данные не будут доставлены, даже если существует другой физический путь для их передачи.

Более гибкими являются алгоритмы *динамической и адаптивной маршрутизации*, которые умеют обновлять содержимое таблиц маршрутов. Обновление происходит на основе обработки специальных сообщений, приходящих от других компонентов сети, занимающихся маршрутизацией в соответствии с определенным протоколом.

Такие алгоритмы принято делить на два вида. Первый вид – сетевые протоколы (*routed protocols* – протоколы направления), реализующие продвижение пакетов через сеть. Другой вид – протоколы маршрутизации (*router protocols*), собирают информацию о характеристиках («расстояниях») межсетевых соединений, на основе которых рассчитываются или корректируются таблицы маршрутизации.

**Связь с установлением логического соединения и без него.** В основе всех средств связи между удаленными процессами лежит передача сообщений. Применяются две модели передачи: потоковая и отдельные сообщения. Реализация происходит на основе *протоколов транспортного уровня*.

Транспортные протоколы связи удаленных процессов, которые предназначены для обмена отдельными сообщениями, получили наименование протоколов *без установления логического соединения (connectionless)* или *протоколов обмена датаграммами*. С точки зрения операционных систем все датаграммы – это независимые протокольные единицы, не имеющие ничего общего с другими датаграммами, которыми обмениваются эти же процессы.

С точки зрения процессов, обменивающихся информацией, датаграммы могут быть связаны по содержанию друг с другом. Однако ответственность за установление и поддержание семантической связи между датаграммами лежит не на сетевых частях операционных систем, а на самих пользовательских взаимодействующих процессах (вышележащие уровни эталонной модели).

Транспортные протоколы, которые поддерживают потоковую модель, получили наименование протоколов, требующих установления логического соединения (*connection-oriented*). В их основе лежит передача данных с помощью пакетов информации. Операционные системы сами «нарезают» эти пакеты из передаваемого потока данных, организуют правильную последовательность их получения и снова объединяют полученные пакеты в поток. Эти протоколы должны обеспечивать надежную связь.

Таким образом, многоуровневая модель сети, семейство сетевых протоколов и интерфейсов, двухуровневая адресация, разрешение имен, протоколы маршрутизации, применение датаграмм или потока данных при их передаче есть те составляющие функциональной структуры ИКС, которые обеспечивают удаленное взаимодействие процессов.

Ценность предложенной эталонной модели заключается в том, что она показывает направление, в котором должны двигаться разработчики новых ИС. Наиболее распространенные семейства протоколов лишь до некоторой степени

согласуются с эталонной моделью. На сегодняшний день наблюдается тенденция к всеобщему переходу на протокол TCP/IP.

## 4.2. Архитектура и технологии построения сетей TCP/IP

Подходу к описанию сети как транспортного средства полностью соответствует концепция сетей TCP/IP.

Причина, по которой TCP/IP столь важен сегодня, заключается в том, что он позволяет самостоятельным сетям подключаться к Internet или объединяться для создания корпоративных интрасетей. Уровневая модель TCP/IP отличается от модели OSI (табл. 4.1).

Таблица 4.1

**Сравнение уровней моделей TCP/IP и OSI**

Уровневая модель	
OSI	TCP/IP
Прикладной	Прикладной
Уровень представления	Не присутствует
Сеансовый	Не присутствует
Транспортный	Транспортный
Сетевой	Межсетевой
Канальный	От хоста к сети
Физический	

Различия моделей заключаются в следующем:

- 1) количество уровней разное;
- 2) модель OSI на сетевом уровне поддерживает возможности использования связи с установлением соединения и без установления соединения, на транспортном уровне – только на основе установления соединения;

Модель TCP/IP – на сетевом уровне только без установления соединения, на транспортном – оба режима, предоставляя пользователю выбор.

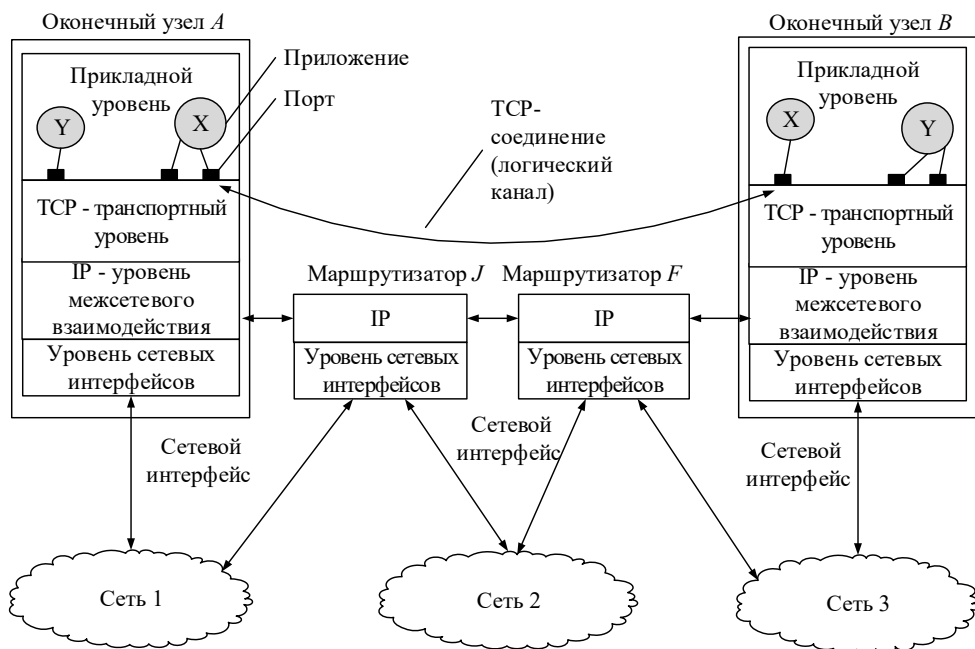
В уровневой модели TCP/IP не описывается подробно, что располагается ниже меж сетевого уровня. Сообщается только, что хост соединяется с сетью при помощи какого-нибудь протокола, позволяющего ему посылать по сети IP-пакеты. Таким протоколом обычно является протокол канального уровня, использующий услуги физического уровня.

### 4.2.1. Концептуальная модель сети TCP/IP

TCP/IP – это промышленный стандарт стека протоколов, используемый в глобальных и локальных сетях. Аббревиатура TCP/IP включает обозначение двух базовых протоколов, на основе которых строится сеть: *TCP (Transmission Control Protocol)* – *протокол управления передачей* и *IP (Internet Protocol)* – *межсетевой протокол*. В целом же стек протоколов TCP/IP включает протоколы четырех уровней (прикладного, транспортного, сетевого и уровня доступа, или уровня сетевых интерфейсов, – по сути, это канальный и физический уров-

ни). Данный стек использует в качестве транспортной среды между узлами коммутации (шлюзами/маршрутизаторами) другие сети или выделенные каналы. Стек ТСР/ІР является самым популярным средством организации ассоциативных (объединенных, составных) сетей.

Концептуальная модель взаимодействия двух прикладных процессов представлена на рис. 4.2.



**Рис. 4.2. Концептуальная модель ТСР/ІР**

Основными элементами сети ТСР/ІР являются оконечные устройства (компьютеры) и узлы коммутации, а также связывающие их каналы физической среды передачи.

Узлы коммутации представляют собой маршрутизаторы ІР-пакетов, а в роли физической среды обычно выступают различные сети, построенные на других сетевых технологиях (LAN или WAN). В ПБД таких сетей от уровня звена данных и выше инкапсулируются ІР-пакеты для передачи от одного маршрутизатора к другому (или от оконечного устройства к ближайшему маршрутизатору). Для соединения маршрутизаторов между собой и с оконечными устройствами используются также выделенные (или коммутируемые) цифровые каналы в сочетании с вспомогательными протоколами пакетной передачи данных типа «точка–точка», в ПБД которых можно инкапсулировать ІР-пакеты.

В сети Internet, например, различают не только отдельные сети, но и более крупные объединения – автономные системы. *Автономная система* – это совокупность сетей под единым административным управлением, обеспечива-

ющим общую для всех входящих в автономную систему маршрутизаторов политику маршрутизации.

Сегодня Internet представляет собой объединение равноправных автономных систем с произвольной топологией связей.

Вся эта конструкция «склеивается» благодаря протоколу сетевого уровня IP (Internet Protocol – протокол сети Internet). Его работа заключается в транспортировке дейтаграмм от отправителя к получателю независимо от того, находятся эти машины в одной и той же сети или нет.

Транспортный уровень берет сегмент из потока данных и разбивает его на датаграммы. Теоретически размер каждой датаграммы может достигать 64 Кбайт, однако на практике они обычно не более 1500 байт (укладываются в один кадр Ethernet). Каждая датаграмма пересылается по Internet, возможно, разбиваясь при этом на более мелкие фрагменты, собираемые сетевым уровнем получателя в исходную дейтаграмму. Затем эта датаграмма передается транспортному уровню, вставляющему ее во входной поток получающего процесса. На пути следования дейтаграммы может оказаться несколько промежуточных сетей.

Протокол TCP отслеживает передаваемые блоки данных. У каждого хоста должен быть *уникальный глобальный адрес* (межсетевой). У каждого *процесса* хоста должен быть уникальный адрес в пределах данного хоста – порт. Совокупность номеров сети, узла и порта в узле образует гнездо. Такая адресация позволяет транспортному протоколу (TCP) доставить данные нужному процессу.

Хост *A* передает сообщение протоколу TCP с указанием переслать его хосту *B* (см. рис. 4.2).

Протокол TCP ориентирован на *виртуальное соединение* – некий логический (виртуальный) канал связи между двумя оконечными устройствами сети. Это соединение идентифицируется специальным числовым кодом, присваиваемым коммутаторами, участвующими в процессе установления соединения. Более детально на понятии «виртуальный канал» остановимся далее.

Протокол TCP использует для транспортировки IP-датаграммы (уровень 3), которые пересылаются посредством протокольных кадров второго уровня. Между двумя партнерами может быть прямое соединение, а может располагаться большое число сетевых приборов второго и третьего уровней.

Протоколу IP не нужно знать адрес (номер) порта *B*. Он передает сообщение уровню доступа к сети 1 (например логическому уровню Ethernet) с указанием переслать его маршрутизатору *J*. Управляющая информация передается вместе с данными в заголовках протокольного блока.

Заголовок сегмента (TCP) содержит номер порта приемника, номер сегмента, контрольную сумму. Заголовок датаграммы (IP) хранит адрес хоста получателя. Заголовок уровня доступа содержит информацию, необходимую для передачи пакета по подсети (в данном примере адрес маршрутизатора *J*, запрос характеристик).

На маршрутизаторе *J* заголовок пакета удаляется, изучается заголовок IP. IP-модуль маршрутизатора направляет дейтаграмму по подсетям 2, 3 и марш-

рутизатору *F* хосту *B*. Для этого в маршрутизаторах к дейтаграмме добавляется заголовок доступа к сети. На хосте *B* происходит обратный процесс.

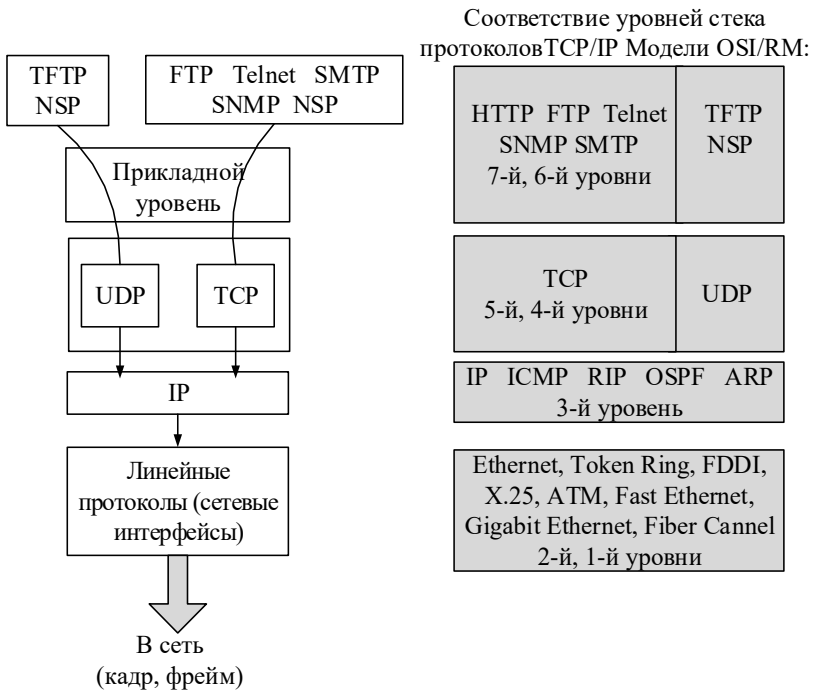
Блок данных на каждом протокольном уровне называют *протокольным модулем данных* (Protocol Data Unit, PDU).

Протоколы, такие как TCP и IP, задают правила для передачи сообщений, описывают детали форматов сообщений и указывают, как обрабатывать ошибки. Самое важное то, что они позволяют рассматривать стандарты взаимодействия вне зависимости от того, на оборудовании какого производителя они реализуются.

### 4.2.2. Стек протоколов TCP/IP

Стек протоколов TCP/IP включает транспортные протоколы TCP/IP; протоколы сбора маршрутной информации RIP и OSPF; протоколы управления сетью ICMP и ARP; прикладные протоколы HTTP, FTP, Telnet, SNMP, SMTP, TFTP, NSP.

Соответствие уровней стека протоколов TCP/IP семиуровневой модели OSI показано на рис. 4.3.



**Рис. 4.3. Уровни стека протоколов TCP/IP**

Краткое описание основных протоколов семейства TCP/IP с расшифровкой аббревиатур приведено в Приложении 1.

Рассмотрим особенности протоколов стека TCP/IP на каждом из четырех уровней: прикладном, транспортном, сетевом и уровне доступа.

#### 4.2.2.1. Прикладной уровень

Прикладной уровень (*уровень приложений – application layer*) объединяет службы, предоставляющие телекоммуникационные услуги различным пользовательским приложениям. Протоколы прикладного уровня ориентированы на конкретные службы. Они определяют как процедуры по организации взаимодействия конкретного типа между прикладными процессами, так и форму представления информации при таком взаимодействии.

Протоколы TCP и IP являются «стержнем» Internet. С точки зрения пользователя Internet TCP/IP – набор прикладных программ, использующих сеть для выполнения полезных коммуникационных задач. Эти программы имеют свои собственные протоколы обмена информацией, например HTTP для WWW, FTP (передача файлов), SMTP (электронная почта), SSH (безопасное соединение с удаленной машиной), DNS (преобразование символьных имен в IP-адреса) и мн. др.

В массе своей эти протоколы работают поверх TCP или UDP. Перед тем как обменяться информацией, клиент и сервер должны сначала установить соединение TCP/IP. Чтобы отличать протоколы, приложения используют для каждого из них уникальные номера. Общие протоколы, такие как FTP и HTTP, используют «хорошо известные» *номера портов*. Стандартным значением для порта HTTP является 80, хотя сервер и клиент могут работать и по другому номеру. Наиболее известные протоколы Web и Internet используют следующие номера портов:

File Transfer Protocol (FTP) .....	21
TELNET Protocol .....	23
Simple Mail Transfer Protocol (SMTP) .....	25
Trivial File Transfer Protocol .....	69
Gopher Protocol .....	70
Finger Protocol .....	79
HTTP Protocol .....	80

Эти порты определены Агентством по выделению имен и уникальных параметров протоколов (IANA). Все порты с номером менее 1024 называются *привилегированными*, и только их используют в качестве стандартных. Невозможно создать свой порт с номером меньше чем 1024.

#### 4.2.2.2. Транспортный уровень

Транспортный уровень обеспечивает обмен сообщениями между прикладными процессами. Идентификация процесса получателя осуществляется по адресу, состоящему из двух частей: IP-адреса, идентифицирующего оконечное устройство, и номера порта, идентифицирующего прикладной процесс. В данном случае «порт» – это не физический разъем или канальный вход/выход, а условный номер прикладного процесса или службы прикладного уровня.

На транспортном уровне стека TCP/IP поддерживаются два режима: с установлением соединения (при использовании протокола TCP) и без него (при применении протокола UDP).

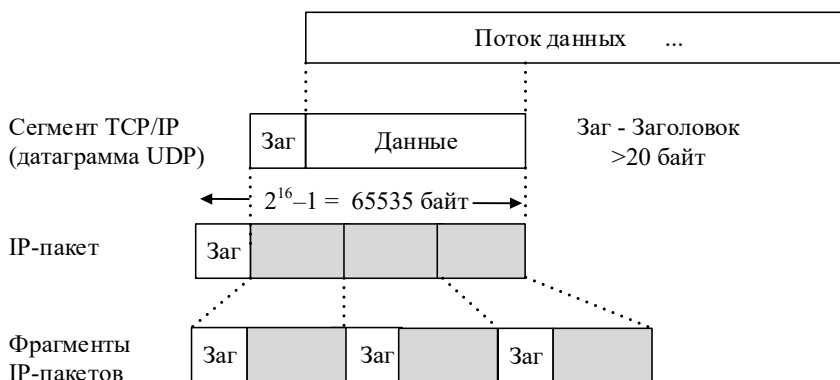
**Протокол TCP** реализован только на оконечных системах. Обеспечивает надежную доставку блоков данных с помощью установления между источником и адресатом логического (виртуального) канала.

Установление логического соединения позволяет нумеровать сегменты, подтверждать их прием квитанциями, в случае потери организовывать повторные передачи, распознавать и уничтожать дубликаты, доставлять прикладному уровню сегменты в том порядке, в котором они были отправлены. Протокол TCP дает возможность объектам одного ранга на компьютере-отправителе и компьютере-получателе поддерживать обмен данными в дуплексном режиме.

Протокол TCP предоставляет полноценную транспортную службу, которая обеспечивает обмен потоками данных. При этом он не накладывает ограничений на состав потока, освобождая прикладной процесс от функции структурирования данных. Передача данных протоколу TCP аналогична их записи в неструктурированный файл.

Целостность потока данных обеспечивается квитированием, при этом для управления потоком используется механизм окна. Управление шириной окна позволяет защищать от перегрузок как промежуточные узлы сети, так и буферную память, принимающую данные. Первую задачу решают маршрутизаторы, направляя протоколам оконечных станций требования об уменьшении размера окна. Вторая задача решается непосредственно протоколом TCP, который декларирует выбранную ширину окна, используя при необходимости и нулевую ширину, т. е. запрещая передачу.

Форматы ПБД транспортного и сетевого уровней TCP/IP показаны на рис. 4.4.



**Рис. 4.4. Формат протокольных блоков данных TCP/IP**

Сегмент TCP состоит из полей заголовка и данных. Длина информационного поля данных сегмента может меняться в широких пределах с учетом двух ограничений. Во-первых, каждый сегмент, включая его заголовок и заголовок

IP-пакета, не должен превышать  $65\,535 = (2^{16}-1)$  байт. Во-вторых, в каждой сети есть соответствующая используемым ПБД максимальная единица передачи (MTU – *Maximum Transfer Unit*), в которую должен помещаться сегмент (вместе с заголовком IP-пакета). Впрочем, если сегмент (пакет) проходит через последовательность сетей и попадает в сеть, MTU-единица которой оказывается меньше него, то пограничный маршрутизатор фрагментирует такой сегмент на несколько частей. Сегменты могут и не содержать поля данных при передаче квитанций и управляющих сообщений.

Формат заголовка сегмента приведен на рис. 4.5. В заголовке сегмента 11 полей и 6 служебных бит.

Поля «номер порта отправителя» и «номер порта получателя» (по 16 бит каждое) идентифицируют службы прикладного уровня или прикладные процессы отправителя и получателя.

Поля «номер последнего переданного байта» и «номер первого ожидаемого на приеме байта» (по 32 бит каждое) используются для управления потоком данных. Нумеруются не ПБД (сегменты), а байты в сегменте, причем цикл нумерации составляет  $2^{32}$  байт = 4 Гбайта, т.е. он вполне может применяться не для относительной нумерации (по кругу), а для абсолютной (например, для последовательной нумерации байт в передаваемых файлах).



**Рис. 4.5. Формат заголовка сегмента TCP**

Поле «Длина заголовка» (4 бита) определяет длину заголовка сегмента TCP, измеренную в 32-битовых блоках (словах, соответствующих типовой адресуемой единице в 32-разрядных компьютерах). Длина заголовка не фиксирована, она зависит от числа бит в поле «Дополнительная информация», которое округляется с помощью пустого поля «Выравнивание» в большую сторону до кратной 32 битам величины. Минимальный размер заголовка при отсутствии дополнительной информации – 20 байт. Максимальный размер, который можно записать в поле из 4 бит, составляет  $32(2^4 - 1) = 480$  бит = 60 байт.

Следующие 6 бит зарезервированы для последующего использования.

Далее следуют 6 служебных бит (1-битных флагов). Активному состоянию служебных бит соответствует значение 1. Флаги используются в качестве следующих признаков (указателей):



- **URG** – указатель наличия срочных данных в сегменте;
- **ACK** – указатель наличия осмысленных данных в поле «Номер первого ожидаемого на приеме байта» (квитанция на ранее принятые данные). При ACK = 0 указанное поле игнорируется;
- **PSH** – указатель получателю, чтобы он доставил данные в сегменте сразу прикладному процессу, а не хранил его в буфере;
- **RST** – указатель запроса на сброс и переустановку соединения (из-за сбоя хоста или другой тупиковой ситуации);
- **SYN** – признак *служебных сегментов Connection Request* – запрос соединения (при ACK = 1) и *Connection Accepted* – согласие на соединение (при ACK = 0);
- **FIN** – признак того, что у отправителя нет больше данных для передачи. Используется для разрыва соединения.

У сегментов с битами FIN и SYN есть порядковые номера, что гарантирует правильный порядок их выполнения.

Поле «Ширина скользящего окна» (16 бит) используется для управления потоком данных в виртуальном соединении и сообщает, сколько байт может быть послано после получившего подтверждение байта. Нулевое значение данного поля является командой приостановки передачи (что напоминает служебный кадр RNR в протоколе HDLC). Командой продолжения передачи служит ненулевое значение поля «Ширина скользящего окна» при таком же значении поля «Номер первого ожидаемого на приеме байта», как у команды приостановки передачи (служебный кадр RR в протоколе HDLC).

Поле «Контрольная сумма» (16 бит) содержит проверочную последовательность, позволяющую обнаруживать ошибки в сегменте. Заполняется и проверяется данное поле без использования циклического кодирования. Алгоритм вычисления просто складывает все 16-ричные слова в дополнительном до 1 коде, затем рассчитывает дополнение для всей суммы. Получатель считает контрольную сумму всего сегмента, включая поле «Контрольная сумма», результат должен быть равен нулю.

Поле «Указатель на срочные данные» содержит смещение в байтах от текущего порядкового номера байта до места расположения срочных данных. Поле проверяется в случае, когда флаг URG установлен в 1. Так в протоколе TCP/IP реализуются прерывающие сообщения прикладного уровня.

Использование скользящего окна протоколом TCP сочетается с процедурой выборочного повтора, аналогичной процедуре РОС-АП (см. разд. 2.3), поддерживаемой стандартом HDLC.

В целом протокол TCP поддерживает множество процедур, обеспечивающих эффективное управление TCP-соединением, передачей данных, борьбой с перегрузками, таймерами.

**Модели реализации протокола TCP.** Протокол TCP функционирует нормально при выполнении ряда условий.

1. Вероятность ошибки доставки невелика, и потеря пакета вероятнее всего происходит из-за переполнения буфера. Если потеря пакета из-за его искажения существенна, уместно поискать оптимальное значение MTU.

2. Время доставки достаточно стабильно. Для его оценки можно использовать простые линейные аппроксимации и модель виртуального канала. Смена порядка прихода пакетов маловероятна.

3. Сеть имеет фиксированную полосу пропускания и не допускает скачкообразных ее вариаций.

4. Буферы сетевых устройств используют схему «первым вошел – первым вышел» (FIFO). Предполагается, что размер этих буферов соответствует произведению  $RTT \times B$ , где  $B$  – полоса пропускания;  $RTT$  – сумма времен транспортировки сегмента от отправителя к получателю и времени движения отклика от получателя к отправителю. Если это условие нарушено, пропускная способность неизбежно понизится и будет определяться размером буфера, а не полосой пропускания канала.

5. Чтобы минимизировать влияние избыточности, связанной с заголовком (20 байт IP + 20 байт TCP + MAC-заголовок), используемое поле данных должно иметь большой объем. Для узкополосных каналов, где MTU мало, нарушение данного требования делает канал низкоэффективным. По этой причине выявление допустимого MTU в начале сессии должно приветствоваться.

6. Взаимодействие с другими TCP-сессиями не должно быть разрушительным, приводящим к резкому снижению эффективности виртуального канала.

В настоящее время предложено и опробовано несколько разновидностей протокола TCP.

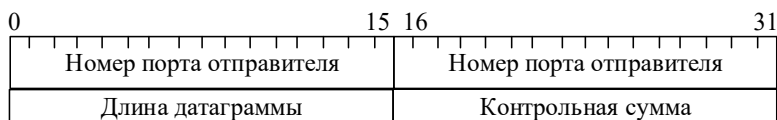
Анализируя различные модели работы протокола TCP, следует учитывать, что в сети Internet могут встречаться участки с разными протоколами L2 (Ethernet, ATM, SDH, Frame Relay, PPP и т. д.). Эти технологии имеют разные алгоритмы обработки ситуаций перегрузки (или не имеют их вовсе), а отправитель и получатель, как правило, не имеют данных о том, какие протоколы второго уровня реализуют виртуальное соединение четвертого уровня.

Модификации (модели) протокола TCP связаны с алгоритмами регулирования размера *окна* в зависимости от частоты перегрузок и времени «обращения» сегмента данных (TCP-Reno, TCP Vegas, TCP-Tahoe, TCP Westwood, модель TCP Hybla, модель BIC TCP, модель CUBIC TCP, TCP-Illinois, TCP-Veno).

**Протокол UDP** (User Datagram Protocol) предназначен для передачи дейтаграмм на уровне абонент–абонент без установления соединения, ориентирован на транзакции, не гарантирует доставку пакета или отсутствие его дубликата.

Прикладные программы через протокол UDP получают доступ к сетевому уровню почти без обработки на транспортном уровне. Многие приложения «клиент – сервер», чтобы обменяться одним запросом и ответом, предпочитают не устанавливать соединения, а пользоваться протоколом UDP (см. протоколы прикладного уровня, применяющие транспортные услуги UDP).

Как и сегмент TCP, датаграмма UDP состоит из заголовка и блока данных. Длина информационного поля данных датаграммы, как и сегмента, может достигать 65 535 байт. А заголовок датаграммы намного проще, чем заголовок сегмента, и включает всего четыре 16-битных поля (рис. 4.6).



**Рис. 4.6. Формат заголовка датаграммы UDP**

Поля заголовка «*Номер порта отправителя*» и «*Номер порта получателя*» (по 16 бит каждое) так же, как аналогичные поля в заголовке сегмента, идентифицируют службы прикладного уровня или прикладные процессы отправителя и получателя.

Поле «*Длина датаграммы*» включает суммарный размер 8-байтового заголовка и поля данных датаграммы.

Поле «*Контрольная сумма*» вычисляется и проверяется таким же способом, как и в сегменте TCP. Контрольная сумма может не рассчитываться, тогда это поле содержит нули.

Протокол UDP самостоятельно не может управлять потоком, следить за порядком следования дейтаграмм и переспрашивать искаженные или потерянные датаграммы (хотя обнаруживать искажения может).

Таким образом, основные функции протокола UDP – мультиплексирование и демультиплексирование (распределение по портам) потока дейтаграмм между приложениями. Кроме того, использование контрольной суммы позволяет контролировать достоверность данных.

#### **4.2.2.3. Сетевой уровень**

Сетевой уровень (чаще называемый *межсетевым* от слова *internet* или *уровнем межсетевого взаимодействия*) является стержнем всей архитектуры TCP/IP. Именно он обеспечивает перемещение пакетов от одних оконечных устройств (граничных узлов, хостов) к другим через маршрутизаторы в пределах всей сети.

Основной протокол сетевого уровня – *Internet protocol (IP)* – *межсетевой протокол*, в связи с чем сети TCP/IP часто именуют *IP-сетями*.

Протокол IP обеспечивает только маршрутизацию и доставку пакетов данных и полностью освобожден от задач обеспечения надежности. Функции транспортного и сетевого уровней четко разделены, что исключает их дублирование.

Протокол IP реализуется программным обеспечением оконечных устройств пользователей (в том числе граничных хостов) и маршрутизаторов и не зависит от характеристик, связывающих их (т. е. пользователей) WAN и LAN.

Важной особенностью протокола IP является его способность выполнять динамическую фрагментацию пакетов при передаче их между сетями с различными максимально допустимыми значениями поля данных кадров (пакетов) MTU. Свойство фрагментации во многом способствовало тому, что протокол IP смог занять доминирующие позиции в сложных составных сетях.

Существует прямая связь между функциональной сложностью протокола и сложностью заголовков пакетов, которые такой протокол использует. Это объясняется тем, что основные служебные данные содержатся в полях заголовка пакетов. На основании этих данных модули, реализующие данный протокол на разных сетевых устройствах, выполняют то или иное действие. Поэтому очень полезно изучить назначение каждого поля заголовка IP-пакета (рис. 4.7), что дает не только формальные знания о структуре пакета, но и объясняет все основные режимы работы протокола по обработке и передаче IP-дейтаграмм.

Поле «*Номер версии*» (Version) указывает версию протокола IP. Сейчас повсеместно используется версия 4 (IPv4), постепенно совершается переход на версию 6 (IPv6).

Длина заголовка является переменной величиной, для указания которой выделено поле «*Длина заголовка*» (IHL). Значение длины заголовка измеряется в 32-битовых словах. Минимальная (обычная) длина заголовка 20 байт (пять 32-битовых слов). При увеличении объема служебной информации эта длина может быть больше за счет использования дополнительных байт в поле «*Опции*» (Options). Наибольший заголовок занимает 60 октетов.

0		D,N,R			15	16	31	
Номер версии (4 бита)	Длина заголовка (4 бита)	Приоритет (3 бита)	Критерий (3 бита)	Резерв (2 бита)	Общая длина пакета (16 бит)			
Идентификатор фрагмента (16 бит)					Флаги (3 бита)		Смещение фрагмента (13 бит)	
						D M		
Время жизни (8 бит)		Протокол верхнего уровня (8 бит)			Контрольная сумма (16 бит)			
IP-адрес отправителя (32 бита)								
IP-адрес получателя (32 бита)								
Опции					Выравнивание			

**Рис. 4.7. Формат заголовка пакета IP**

Поле «*Приоритет*» (Precedence) задает приоритетность пакета. Приоритет может иметь значения от самого низкого – 0 (нормальный пакет) до самого высокого – 7 (пакет управляющей информации). Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь.

Поле «*Критерий*» задает критерий выбора маршрута. Выбор осуществляется между тремя альтернативами, каждой из которых соответствует свой

1-битный флаг: с малой задержкой – бит D (delay), с высокой достоверностью – бит T (true) и с высокой пропускной способностью – бит R (rate).

Поля «*Приоритет*» и «*Критерий*» вместе с полем «*Резерв*» (из 2 бит) часто объединяют в одно поле, называемое «*Тип сервиса*» (Type of Service) длиной 1 байт.

Поле «*Общая длина пакета*» (Total Length of Packets) содержит длину всей датаграммы, включая заголовок и поле данных. Максимальная длина пакета ограничена разрядностью поля и составляет 65 535 байт. В настоящее время этот предел достаточен, однако с появлением гигабитных сетей могут понадобиться датаграммы большего размера.

Поле «*Идентификатор фрагмента*» (Identification) позволяет хосту определить, какой дейтаграмме принадлежат принятые им фрагменты. Все фрагменты одного пакета содержат одно и то же значение этого поля.

Поле «*Флаги*» (Flags) содержит признаки, связанные с фрагментацией. Установленный бит D (Do not Fragment) запрещает маршрутизатору фрагментировать данный пакет, а установленный бит M (More Fragments) говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Еще 1 бит поля зарезервирован.

Поле «*Смещение фрагмента*» (Fragment Offset) указывает положение фрагмента в исходной дейтаграмме. Длина всех фрагментов, кроме последнего, должна быть кратна 8. Максимальное количество фрагментов в дейтаграмме равно 8192, что покрывает максимальную длину датаграммы 65 535 байт.

Поле «*Время жизни*» (Time to Live) содержит предельный срок, в течение которого пакет может перемещаться по сети. Время жизни этого пакета измеряется в секундах и задается источником передачи. На маршрутизаторах и в других узлах сети по истечении каждой секунды из текущего времени жизни вычитается единица; она вычитается и в том случае, когда время задержки меньше секунды. Поскольку современные маршрутизаторы редко обрабатывают пакет дольше чем за 1 с, то время жизни можно считать равным максимальному числу узлов, которые разрешено пройти такому пакету до того, как он достигнет места назначения. Если параметр времени жизни станет нулевым до того, как пакет достигнет получателя, этот пакет будет уничтожен.

Поле «*Протокол верхнего уровня*» (Protocol) указывает, какому протоколу верхнего уровня (или своего же сетевого) принадлежит информация, размещенная в поле данных пакета (например, это могут быть сегменты протокола TCP, датаграммы UDP, пакеты ICMP или OSPF).

Поле «*Контрольная сумма*» (Header Checksum) рассчитывается только по заголовку пакета. Поскольку некоторые поля заголовка меняют свое значение в процессе передачи пакета по сети (например, время жизни), контрольная сумма проверяется и повторно рассчитывается при каждой обработке IP-заголовка. Она подсчитывается как дополнение к сумме всех 16-битовых слов заголовка. При вычислении контрольной суммы значение самого поля «Контрольная сумма» устанавливается в нуль. Если контрольная сумма не верна, то пакет будет отброшен, как только ошибка будет обнаружена.

Поля «*IP-адрес отправителя*» (Source IP Address) и «*IP-адрес получателя*» (Destination IP Address) задают адресную информацию, необходимую для маршрутизации пакетов (номера сети и хоста).

Поле «*Опции*» (IP Options) является необязательным и используется обычно только при отладке сети. Оно состоит из нескольких подполей. В них можно указывать точный маршрут прохождения маршрутизаторов, регистрировать проходимые пакетом маршрутизаторы, помещать данные системы безопасности, а также временные отметки.

Поле «*Выравнивание*» (Padding) используется для дополнения поля «*Опции*» (нулями) до значения, кратного 32 битам.

**Адресация в IP-сетях.** В стеке TCP/IP используются три типа адресов: физический (*MAC-адрес*); сетевой (*IP-адрес*); символьный (*DNS-имя*).

**Сетевой IP-адрес.** Длина адреса IP (32 бита, IPv4) разделена на две части. Первая обозначает адрес сети, вторая – адрес узла (хоста). Такое деление позволяет передавать сообщения между сетями только на основании номера сети, а номер узла используется после доставки сообщения в нужную сеть.

Номер узла назначается администратором независимо от локального адреса узла. IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение (конечный узел может входить в несколько IP-сетевых связей).

Сети IP могут также быть разделены на более мелкие единицы, называемые подсетями (*subnets*). Подсети обеспечивают дополнительную гибкость для администратора сети. Число битов, занимаемых для адреса подсети, является переменной величиной.

Адреса IP записываются в формате десятичного числа с проставленными точками, например 192.228.17.57. Адресация IP обеспечивает 5 различных классов сети (табл. 4.2). Самые крайние левые биты обозначают класс сети. Наиболее распространены классы А, В и С. Классы D и Е обычно не используются конечными пользователями.

Большие сети применяют адреса класса А, средние – класса В, малого размера – класса С.

Таблица 4.2

Классы сети адресации IP

Класс	Первые биты IP-адреса	Наименьший номер сети	Наибольший номер сети	Максимальное число сетей	Максимальное число узлов в каждой сети
A	0	0.0.0.0	127.0.0.0	$2^7 - 2$	$2^{24} - 2$
B	10	128.0.0.0	191.255.0.0	$2^{14} - 2$	$2^{16} - 2$
C	110	192.0.0.0	223.255.255.0	$2^{21} - 2$	$2^8 - 2$
D	1110	224.0.0.0	239.255.255.255	–	–
E	1111	240.0.0.0	247.255.255.255	–	–

В версии 4 (IPv4) существуют определенные соглашения об использовании адресов.

1) Сеть с номером 0.0.0.0 зарезервирована для использования в служебных сообщениях, а сеть с номером 127.0.0.0 используется для петлевого соеди-

нения (пересылки пакетов самим себе), поэтому общее количество сетей класса А равно 126.

2) Маршрутизация пакета в публичной сети всегда производится на основании классического IP-адреса номера сети, согласно табл. 4.2, поэтому адрес сети не может быть назначен ни одному узлу.

3) Адрес узла со всеми двоичными «1» предназначен для адресации всем узлам соответствующей сети (широковещательная рассылка), поэтому этот адрес не может быть назначен ни одному узлу. Совместно с пунктом 2 это означает, что число узлов в любой сети уменьшается на 2.

4) В каждом классе имеется диапазон сетевых адресов для частного использования, которые в публичных сетях отсутствуют. Они используются для построения локальных либо корпоративных сетей. В классе А – это сеть 10.0.0.0, в классе В – диапазоны сетей от 172.16.0.0 до 172.31.0.0, в классе С – диапазон сетей от 192.168.0.0 до 192.168.256.256.

Основное назначение адресов класса D – распространение информации по схеме «один-ко-многим» для групповой рассылки в Интернет аудио- и видеоинформации. Адреса класса E зарезервированы для будущих применений.

Номер сети принято обозначать с помощью маски. Маска – это число, которое используется вместе с IP-адресом: двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Для классов сетей маски имеют следующий вид:

- класс А – 11111111.00000000.00000000.00000000 (255.0.0.0);
- класс В – 11111111.11111111.00000000.00000000 (255.255.0.0);
- класс С – 11111111.11111111.11111111.00000000 (255.255.255.0).

Создание подсетей обеспечивается переназначением части битов узла в качестве битов сети. Процесс заимствования части битов всегда начинается с крайнего левого бита.

Поступивший (в маршрутизатор, узел) IP-адрес в двоичном коде складывается поразрядно с маской с помощью логической операции «И»:

$$0 + 0 = 0, 0 + 1 = 0, 1 + 0 = 0, 1 + 1 = 1. \quad (4.1)$$

Результат сложения (адрес сети) сравнивается с IP-адресом, записанным в первой строке таблицы маршрутизации. При совпадении адресов поступивший пакет направляется на соответствующий интерфейс. В случае несовпадения сравнение продлевается с последующими строками маршрутной таблицы, если они имеются. Если совпадения нет, поступившая датаграмма отбрасывается (фильтруется).

*Пример 1.* Имеется глобальная сеть с 150 узлами в трех сетях (в разных городах), соединенных маршрутизатором TCP/IP. У каждой из этих трех сетей 50 узлов. Выделяем сеть класса С 192.168.123.0. Это значит, что адреса с 192.168.123.1 по 192.168.123.254 можно использовать для этих 150 узлов.

Два адреса в данном примере – 192.168.123.0 и 192.168.123.255 – использовать нельзя, поскольку двоичные адреса с составляющей узла из одних единиц и нулей недопустимы (см. п. 2 и 3). Следует просто запомнить, что первый

и последний адреса в любой сети и подсети не могут быть присвоены отдельному узлу.

Теперь осталось дать IP-адреса 254 узлам. Это несложно, если все 150 компьютеров являются частью одной сети. Однако в данном примере 150 компьютеров работают в трех отдельных физических сетях. Разбиваем сеть на подсети. С помощью маски «одалживаем» несколько разрядов, обычно применяемых для задания адреса узла, и используем их для составляющей сети в адресе. Маска подсети 256.256.256.192 позволяет создать четыре сети с 62 узлами в каждой. Это возможно, поскольку в двоичном обозначении 256.256.256.192 – то же самое, что и 1111111.11111111.1111111.11000000. Первые две цифры последнего октета становятся адресами сети, поэтому появляются дополнительные сети 00000000 (0), 01000000 (64), 10000000 (128) и 11000000 (192). В этих четырех сетях последние 6 двоичных цифр можно использовать в качестве адресов узлов. Эти четыре сети будут иметь следующие действующие адреса узлов: 192.168.123.1 – 62; 192.168.123.65 – 126; 192.168.123.129 – 190; 192.168.123.193 – 254.

Не забываем, что двоичные адреса узлов с одними только единицами и нулями недействительны, поэтому нельзя использовать адреса со следующими числами в последнем октете: 0, 63, 64, 127, 128, 191, 192 или 256.

*Пример 2.* Рассмотрим еще один пример работы маршрутизатора в качестве межсетевого узла, сопрягающего разные сети внутри корпоративной.

Пусть, например, администратор получил адрес сети 135.38.0.0 (адрес класса В, маска сети 255.255.0.0). В этой сети по 16 битов выделено на адреса сети и узла. Администратору необходимо 8000 узлов, на это нужно выделить только 13 битов на адреса узлов ( $2^{13} = 8192$ ), следовательно, оставшиеся  $16 - 13 = 3$  бита адреса узла можно переназначить как адрес сети. Тогда маска образованной подсети в двоичном коде будет иметь вид 11111111.11111111.11100000.00000000, или 255.255.224.0 (жирным шрифтом выделены заимствованные биты адреса узлов класса В). В результате такого деления получим адреса подсетей, приведенные в табл. 4.3.

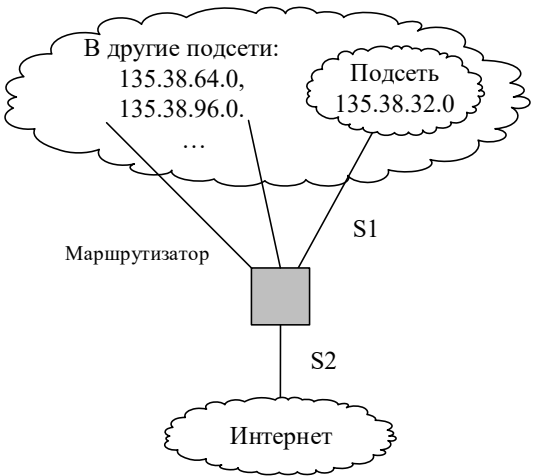
**Таблица 4.3**

Номер сети	Число узлов в подсети
<del>10000111 00100110 00000000 00000000</del> <del>135. 38. 0. 0</del>	<del>8190</del>
10000111 00100110 00100000 00000000 135. 38. 32. 0	8190
10000111 00100110 01000000 00000000 135. 38. 64. 0	8190
10000111 00100110 01100000 00000000 135. 38. 96. 0	8190
10000111 00100110 10000000 00000000 135. 38. 128. 0	8190
10000111 00100110 10100000 00000000 135. 38. 160. 0	8190
10000111 00100110 11000000 00000000 135. 38. 192. 0	8190
<del>10000111 00100110 11100000 00000000</del> <del>135. 38. 224. 0</del>	<del>8190</del>



Две полученные подсети (135.38.0.0 и 135.38.224.0) использовать нельзя, так как сетевой адрес первой подсети совпадает с адресом исходной классической сети класса В, а адрес широковещательной рассылки внутри второй подсети – с адресом широковещательной рассылки исходной классической сети класса В. Теперь одну из оставшихся шести подсетей (например подсеть 135.38.32.0) (рис. 4.8) администратор использует для своих нужд, а оставшиеся пять сетей может отдать другому администратору.

Архитектура местоположения подсети 135.38.32.0 приведена на рис. 4.8.



**Рис. 4.8. Архитектура местоположения подсети 135.38.32.0**

Для обслуживания подсети 135.38.32.0 маршрутизатор использует таблицу маршрутов (табл. 4.4).

**Таблица 4.4**

**Таблица маршрутов**

Пункт назначения (Destination)	Маска сети (Subnet Mask)	Пункт пересылки (Next Hop)	Интерфейс (Interface)	Метрика (Metric)
135.38.32.0	255.255.224.0	0.0.0.2	S1	1
Default	255.255.0.0	0.0.203.0	S2	20

Для определения дальнейшего маршрута следования поступившего пакета маршрутизатор производит операции.

1. Поступивший IP-адрес в двоичном коде с помощью логической операции (4.1) складывается поразрядно с маской сети первой строки таблицы маршрутизации.

2. Полученный в результате сложения адрес сети сравнивается с IP-адресом пункта назначения первой строки. При их совпадении поступивший пакет направляется на интерфейс S1.

3. В случае несовпадения те же операции проделываются с последующими строками маршрутной таблицы, если они имеются.

4. Все поступившие пакеты из подсети 135.38.32.0 направляются по умолчанию на интерфейс S2.

Пример вычисления маршрута пакета по маске подсети рассмотрен в приложении 3 (лабораторная работа 8).

Адреса версии IPv6, имеющие 7 байт на адрес, назначаются отдельным интерфейсам узлов, а не самим узлам. У одного интерфейса может быть несколько уникальных адресов для целевых передач. Более длинные межсетевые адреса позволяют объединять адреса по сетевым иерархиям, поставщикам услуг, географическому расположению, корпорациям. Такая кластеризация таблицы маршрутизации ускоряет процедуру поиска в таблице.

В терминологии TCP/IP под *физическим* (именуемым также *локальным* или *аппаратным*) адресом понимается такой тип адреса, который предназначен для доставки данных в пределах подсети. Если подсеть – локальная сеть, то используется MAC-адрес: сетевые адаптеры, интерфейсы маршрутизаторов. Адрес уникальный, назначается производителем (6 байт). Бывают исключения: некоторые компьютеры в локальной сети могут иметь несколько локальных адресов даже при одном адаптере. Однако протокол IP может работать и над протоколами более высокого уровня, например над IPX или X.26. В этом случае локальными адресами для протокола IP соответственно будут адреса IPX и X.26.

Глобальные порты маршрутизаторов, предназначенные для соединений «точка-точка», не имеют локальных адресов.

*Символьные доменные адреса* строятся по иерархическому принципу: простое имя конечного узла, имя группы узлов (организация), имя более крупной группы (поддомена), ..., до имени домена, объединяющего организации по географическому принципу: ru – Россия, uk – Великобритания, us – США. Domain Name System (DNS) – дает необходимое соответствие.

Пример символьного имени: cityline.spb.ru.

Между доменным именем и IP-адресом узла нет никакого алгоритмического соответствия, поэтому необходимо использовать какие-то дополнительные таблицы или службы, чтобы узел сети однозначно определялся как по доменному имени, так и по IP-адресу. В сетях TCP/IP применяется специальная распределенная *служба доменных имен DNS* (Domain Name System), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Потому доменные имена называют также *DNS-именами*.

**Отображение IP-адресов на MAC-адреса.** Важная задача уровня межсетевых интерфейсов – отображение IP-адресов в локальные MAC-адреса. Используется ARP (Address Resolution Protocol – протокол разрешения адреса). Для каждой сети, подключенной к сетевому адаптеру компьютера или порту маршрутизатора, строится отдельная таблица.

ARP – обращение: IP-адрес узла назначения известен модулю IP; требуется на его основе найти MAC-адрес узла назначения. Работа ARP начинается с просмотра ARP-таблицы. Каждая строка такой таблицы содержит соответствие между IP-адресом и MAC-адресом (как пример, табл. 4.5). Выделяется специ-

альный маршрутизатор, который ведет таблицу ARP для всех маршрутизаторов и узлов глобальной сети.

Реверсивный ARP (Reverse ARP – RARP) решает обратную задачу – нахождение IP-адреса по известному локальному адресу. Используется при старте бездисковых станций, не знающих в начальный момент своего IP-адреса, но знающих адрес своего сетевого адаптера.

**Таблица 4.5**

**Таблица ARP для всех маршрутизаторов и узлов глобальной сети**

IP-адрес	MAC-адрес	Тип записи
194.86.136.75	008048EB7EG0	динамический
194.86.60.21	008048EB7567	статический

**Динамическая настройка IP-адресов.** Для динамической настройки IP-адреса и всех его параметров может использоваться протокол DHCP (*Dynamic Host Configuration Protocol* – протокол динамической настройки узла) – сетевой протокол, позволяющий выдавать компьютерам в одном сегменте широковещания IP-адреса из некоторого указанного диапазона. Кроме адреса, клиент DHCP получает от сервера маску сети, адрес (или адреса) маршрутизаторов, адреса DNS-серверов. Протокол DHCP использует широковещательную рассылку поверх протокола UDP для обмена данными между клиентом и сервером.

Сервер может привязать выдаваемый IP-адрес к MAC-адресу клиента, а клиент попросить у сервера некоторый конкретный IP (например, полученный в последний раз).

**Трансляция сетевых адресов и портов.** Трансляция сетевых адресов (*Network Address Translation, NAT*) позволяет изолировать внутреннюю локальную сеть от внешнего мира и обойти недостаток сетевых адресов протокола IPv4 для всего мира (в протоколе IPv6 данная проблема отсутствует, однако пока что сети IPv4 очень широко используются). Более точно данная трансляция называется трансляцией сетевых адресов и портов (NAPT), поскольку затрагивает не только адреса, но и порты протоколов TCP/UDP.

Трансляция представляет собой отображение пары «адрес во внутренней сети – порт» в пару «внешний адрес маршрутизатора – порт». Отображение задается маршрутизатором в момент, например, обнаружения исходящего из локальной сети соединения.

**Фрагментация пакетов.** Все сети накладывают ограничения на размер своих пакетов. Эти пределы обусловлены различными предпосылками, среди которых следующие:

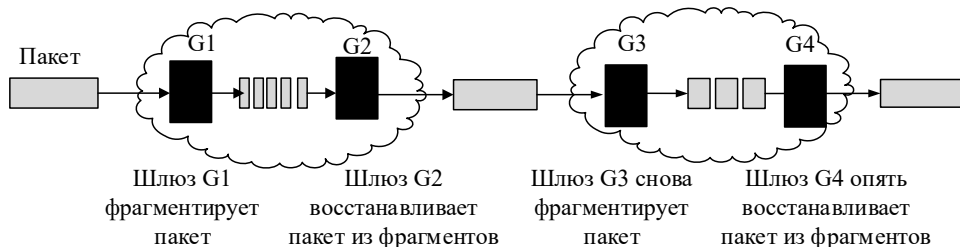
- 1) аппаратные (например, размер кадра Ethernet);
- 2) операционная система (например, все буферы имеют размер 512 байт);
- 3) протоколы (например, количество бит в поле длины пакета);
- 4) соответствие международному или национальному стандарту;
- 5) желание снизить количество пакетов, пересылаемых повторно из-за ошибок передачи;

б) желание предотвратить ситуацию, когда один пакет слишком долгое время занимает канал.

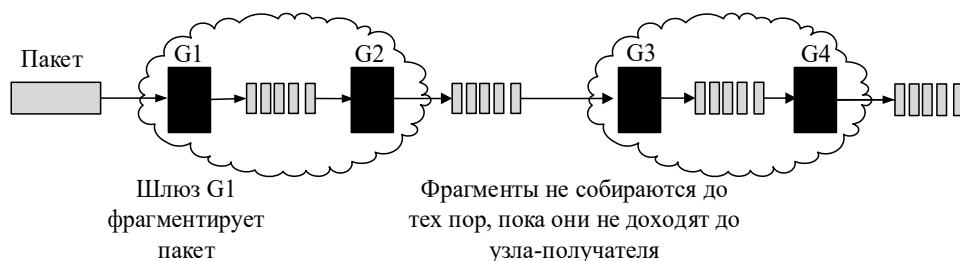
Результатом действия всех этих факторов является то, что разработчики не могут выбирать максимальный размер пакета по своему усмотрению. Максимальный размер поля полезной нагрузки варьируется от 48 байт (АТМ-ячейки) до 65 515 байт (IP-пакеты), хотя на более высоких уровнях он часто бывает больше.

Возникает проблема, когда большой пакет хочет пройти по сети, в которой максимальный размер пакетов слишком мал. Единственное решение заключается в разрешении шлюзам разбивать пакеты на фрагменты и посылать каждый фрагмент в виде отдельного межсетевого пакета. Однако возникает проблема с восстановлением пакета из фрагментов.

Для ее решения применяются две противоположные стратегии. Первая заключается в добавлении в «мелкопакетную» сеть шлюзов (специализированных маршрутизаторов), предоставляющих интерфейсы другим сетям. Когда на такой шлюз приходит пакет слишком большого размера, он разбивается на фрагменты (рис. 4.9, а.). Каждый фрагмент адресуется одному и тому же выходному шлюзу, восстанавливающему из этих фрагментов пакет. Прохождение данных через «мелкопакетную» сеть остается прозрачным для хостов.



а



б

**Рис. 4.9. Фрагментация: а – прозрачная; б – непрозрачная**

Другая стратегия фрагментации состоит в отказе от восстановления пакета из фрагментов на промежуточных маршрутизаторах. Как только пакет оказывается разбитым на отдельные фрагменты, с каждым фрагментом обращаются как с отдельным пакетом. Все фрагменты проходят через один или несколько

ко выходных шлюзов (рис. 4.9, б). Задача восстановления оригинального пакета возложена на получающий хост. Так работает IP.

Каждый фрагмент должен иметь заголовок. Фрагменты пакета нумеруются таким образом, чтобы можно было восстановить исходный поток данных. Так, например, заголовок межсетевого пакета включает два поля: порядковый номер исходного пакета и порядковый номер фрагмента.

#### 4.2.2.4. Уровень доступа (уровень сетевых интерфейсов)

Самый нижний уровень стека TCP/IP соответствует уровню звена данных ЭМВОС. Он поддерживает все популярные стандарты уровня звена данных как для локальных, так и для глобальных сетей.

Основное назначение уровня доступа к сетям (сетевого интерфейса) – обеспечение независимости функционирования протоколов TCP/IP от среды передачи.

Уровень доступа к сетям (к среде передачи) формирует кадры (фрагменты пакетов), имеющие формат, структура которого зависит от характеристик используемых сетей передачи данных или выделенных каналов.

При использовании выделенных (или коммутируемых) цифровых каналов для соединения маршрутизаторов между собой и с оконечными устройствами применяются вспомогательные протоколы пакетной передачи данных типа «точка-точка».

*PPP (Point-to-Point Protocol)* – протокол «точка-точка». При его разработке за основу был взят формат кадров протокола HDLC и дополнен собственными полями. Поля протокола PPP вложены в поле данных кадра HDLC, в кадры Frame Relay и других протоколов глобальных сетей.

Протокол PPP основан на четырех принципах: переговорном принятии параметров соединения, многопротокольной поддержке, расширяемости протокола, независимости от глобальных служб.

Одним из важных параметров PPP-соединения является *режим аутентификации*. Для целей аутентификации PPP предлагает по умолчанию протокол PAP (Password Authentication Protocol), передающий пароль по линии связи в открытом виде, или протокол CHAP (Challenge Handshake Authentication Protocol), не передающий пароль по линии связи и поэтому обеспечивающий большую безопасность сети.

Протокол PPP работает со многими протоколами сетевого уровня (IP, Novell IPX, AppleTalk, DECnet, XNS, Banyan VINES и OSI), и протоколами уровня звена данных LAN.

В PPP используется только один тип кадра HDLC – ненумерованный – информационный. Исправления очень редких ошибок, возникающих в канале, выполняют протоколы верхних уровней (тот же TCP).

В локальной сети доступ станции в среду передачи осуществляется по принципу «точка – многоточка», и технологии локальных сетей специфицированы стандартами IEEE 802.x, которые охватывают только физический и канальный уровни модели OSI (см. разд. 3).

Напомним, канальный уровень ЛС разделен на два подуровня: управления доступом к среде (Media Access Control, MAC) и логической передачи данных (Logical Link Control, LLC). Протоколы уровней MAC и LLC взаимно независимы – каждый протокол MAC-уровня может применяться с любым типом протокола LLC-уровня и наоборот.

Уровень LLC реализует функции интерфейса с прилегающим к нему сетевым уровнем. Через этот уровень сетевой протокол запрашивает от уровня звена данных необходимую ему транспортную операцию с нужным качеством.

Протокол LLC передает свои кадры либо датаграммным способом, либо с помощью процедур с установлением соединения и восстановлением кадров. Он помещает пакет сетевого уровня (пакет IP) в поле «*Data*» своего кадра, который дополняется необходимыми служебными полями *DSAP* (*Destination Service Access Point*) – адрес точки входа службы назначения, *SSAP* (*Source Service Access Point*) – адрес точки входа службы источника, *Control* – управляющее поле.

Поля *DSAP* и *SSAP* позволяют указать, какой сервис верхнего уровня пересылает данные с помощью этого кадра. Программному обеспечению узлов сети при получении кадров канального уровня необходимо распознать, какой протокол вложил свой пакет в поле данных поступившего кадра, для того, чтобы передать извлеченный из кадра пакет нужному протоколу для последующей обработки.

Поле *управления* (один байт) используется для обозначения типа кадра данных – информационный, управляющий или нумерованный. Кадр LLC обрамляется двумя однобайтовыми полями «Флаг», имеющими значение 01111110.

Флаги используются на MAC-уровне для определения границ блока. Формат кадров LLC, за исключением поля адреса точки входа сервиса источника, соответствует формату кадра протокола HDLC (разд. 2.2.4).

Через межуровневый интерфейс протокол LLC передает свой кадр вместе с адресной информацией об узле назначения соответствующему протоколу уровня MAC, который упаковывает кадр LLC в свой кадр (рис. 3.4).

Управляющее поле служит для реализации используемой в LAN версии высокоуровневого протокола управления звеном данных протокола HDLC (разд. 2.2.4).

Стандарт HDLC представляет собой обобщение нескольких близких стандартов LAPx (Link Access Protocol), характерных для различных технологий: LAPB в сетях X.25; LAPF во Frame Relay; LAPD в ISDN; LAPM в сетях абонентского доступа на основе модемов.

В соответствии со стандартом 802.2 уровень управления логическим каналом LLC предоставляет верхним уровням следующие процедуры LLC1, LLC2, LLC3.

Этот набор процедур является общим для всех методов доступа к среде, определенных стандартами 802.3–802.5, а также стандартами FDDI.

Возможности процедур рассмотрены в главе 3.

Отметим, процедура LLC2 во многом аналогична протоколам семейства HDLC (LAPB, LAPD, LAPM) и работает в режиме скользящего окна.

В подуровне LLC используются все три типа кадров протокола HDLC: информационные (information), управляющие или супервизорные (supervisory) и нумерованные (unnumbered).

Таким образом, применение того или иного вида стандарта LARx зависит от того, через какую локальную или глобальную (региональную) сеть проходит физическая составляющая логического канала взаимодействия между удаленными процессами.

Обычно появляющиеся новые технологии локальных или глобальных сетей быстро включаются в стек TCP/IP за счет разработки соответствующего RFC (Requests for Comments), определяющего метод инкапсуляции пакетов IP в ее кадры.

### **Контрольные вопросы**

1. Что отражает функциональная модель глобальной сети?
2. В чем смысл понятия порта, сокета?
3. В чем различие между динамической и адаптивной маршрутизациями?
4. Какие модели передачи сообщений применяют для связи между удаленными процессами?
5. Поясните, как рис. 4.1 отражает взаимодействие процессов.
6. Дайте определение автономной системы.
7. Как соотносятся транспортный и прикладной уровни? Сетевой и транспортный? Сопоставьте с элементами систем грузоперевозок.
8. Как используется поле «Ширина скользящего окна» в формате заголовка сегмента TCP?
9. Что такое и для чего используется маска в системе IP-адресации?
10. Для чего необходимо отображение IP-адресов на MAC-адреса?
11. Какие операции по поступившему IP-адресу выполняет маршрутизатор для определения дальнейшего маршрута?
12. Как реализуется динамическая настройка IP-адресов?
13. Какова основная задача уровня доступа?
14. В чем особенность протокола PPP (Point-to-Point Protocol)?
15. От чего зависит применение того или иного стандарта LARx в уровне доступа?

## 5. ОБЪЕДИНЕНИЕ СЕТЕЙ

Объединение разнородных сетей в единую интернет осуществляется на основе территориальных сетей. Территориальные сети решают проблему формирования глобальных сетей из локальных сетей, сетей регионов и целых стран и даже наднациональных сетей (например, E-BONE для Европы, Internet для мира). Как правило, эти сети строятся с использованием протоколов ATM, ISDN, Frame Relay или X.25, TCP/IP.

Объединяемые в единую интернет сети могут существенно различаться по протоколам (IP, IPX, SNA, ATM, MPLS и др.), адресации (плоская (802.x) или иерархическая (IP)) по размерам пакетов, по параметрам (различные тайм-ауты) и по многому другому.

Рассмотрим сложившиеся подходы к объединению сетей.

### 5.1. Устройства объединения сетей

На физическом уровне сети объединяются повторителями или концентраторами, которые просто переносят бит из одной сети в другую такую же сеть, и преобразовывают сигналы при сопряжении различных физических сред.

Мосты и коммутаторы работают на уровне передачи данных. Они могут принимать кадры, анализировать их MAC-адреса, направлять их в другие сети, осуществляя по ходу минимальные преобразования протоколов, например из Ethernet в FDDI или в 802.11.

На сетевом уровне маршрутизаторы соединяют две сети. Если сетевые уровни у них разные, маршрутизатор может обеспечить перевод пакета из одного формата в другой. Маршрутизатор, поддерживающий несколько протоколов, называется *мультипротокольным* маршрутизатором.

На транспортном уровне существуют транспортные шлюзы, предоставляющие интерфейсы для соединений этого уровня. Транспортный шлюз позволяет, к примеру, передавать пакеты из сети TCP в сеть SNA (протоколы транспортного уровня у них различаются), склеивая одно соединение с другим.

На прикладном уровне шлюзы занимаются преобразованием семантики сообщений. Например, шлюзы между электронной почтой Интернета (RFC 822) и электронной почтой X.400 должны анализировать содержимое сообщений и изменять различные поля электронного конверта.

### 5.2. Технологии межсетевого взаимодействия

Если в разных частях составной сети используются разные сетевые протоколы, то для того, чтобы она функционировала как единая сеть, все узлы которой имели бы возможность обмениваться информацией, может быть использован один из стандартных приемов – мультиплексирование, трансляция, туннелирование.

**Мультиплексирование протоколов** состоит в установке нескольких дополнительных стеков протоколов на одной из конечных машин, участвующих



во взаимодействии. Компьютер с несколькими стеками протоколов использует для взаимодействия с другим компьютером тот стек, который понимает этот другой компьютер.

Для того чтобы запрос от прикладного процесса был правильно обработан и направлен через соответствующий стек, необходимо наличие специального программного элемента – *мультиплексора протоколов*. Мультиплексор должен уметь определять, к какой сети направляется запрос клиента.

Мультиплексирование может осуществляться и на отдельных уровнях протоколов. В общем случае на каждом уровне может быть установлено несколько протоколов, и для каждого уровня может существовать собственный мультиплексор, выполняющий коммутацию между протоколами соседних уровней.

Достоинство: отсутствуют очереди к единственному межсетевому устройству, простая процедура переключения на нужный протокол.

Недостаток: высокая избыточность требует дополнительных ресурсов от рабочих станций, требует навыков работы с транспортными протоколами «чужих» сетей.

Применение ограничено начальной стадией объединения сетей.

**Трансляция протоколов** обеспечивает согласование двух протоколов путем преобразования (трансляции) сообщений, поступающих от одной сети, в формат другой сети. Транслирующий элемент (программный или аппаратный шлюз, мост, коммутатор или маршрутизатор) размещается между взаимодействующими сетями и служит посредником в их «диалоге». Часто транслятор протоколов называют шлюзом в широком смысле, независимо от того, какие протоколы он транслирует. Этим подчеркивается тот факт, что трансляция осуществляется выделенным устройством, соединяющим две разнородные сети.

Трансляция сетевых протоколов является более сложной задачей, чем трансляция канальных протоколов, хотя бы потому, что в отличие от канального уровня, на котором имеется единая система уникальных адресов узлов, каждый сетевой протокол имеет собственный, свойственный только ему, формат адресов. Так, преобразование протокола Ethernet в протокол Token Ring сводится к нескольким несложным действиям, потому что в обоих протоколах используется единая адресация узлов. А вот трансляция протоколов сетевого уровня IP и IPX – более сложный процесс, включающий не только преобразование форматов сообщений, но и отображение адресов сетей и узлов, различным образом трактуемых в этих протоколах.

Кроме различий в системе адресации, в каждом сетевом протоколе имеется еще много других специфических особенностей, которые могут выражаться в различии как количественных параметров (например, для разных протоколов могут быть определены разные величины тайм-аутов, времен жизни пакета или максимальных размеров пакетов), так и в структуре пакетов. Протоколы могут отличаться и функциональными возможностями, например, одни из них реализованы с установлением соединений, а другие – без установления соединений, в одних предусмотрена возможность фрагментации, в других – нет.

При использовании техники трансляции не требуется устанавливать дополнительное программное обеспечение на рабочих станциях, сохраняется привычная среда пользователей и приложений, транслятор полностью прозрачен для них. Проблемы межсетевого взаимодействия локализованы, следовательно, упрощается администрирование, поиск неисправностей, обеспечение безопасности.

Однако транслятор замедляет работу из-за относительно больших временных затрат на сложную процедуру трансляции, а также из-за ожидания запросов в очередях к единственному элементу, через который проходит весь межсетевой трафик. При увеличении числа пользователей и интенсивности обращений к ресурсам другой сети резко снижается производительность – плохая масштабируемость.

**Инкапсуляция протоколов.** Инкапсуляция (encapsulation) или туннелирование (tunneling) применяется, когда две сети с одной транспортной технологией необходимо соединить через сеть, использующую другую транспортную технологию. Промежуточная сеть используется только как *транзитная транспортная система*.

Пограничные маршрутизаторы, которые подключают объединяемые сети к транзитной, упаковывают пакеты транспортного протокола объединяемых сетей в пакеты транспортного протокола транзитной сети.

Инкапсуляция может быть использована для транспортных протоколов любого уровня. Для согласования сетей на сетевом уровне могут быть использованы многопротокольные и инкапсулирующие маршрутизаторы, а также программные и аппаратные шлюзы.

Обычно инкапсуляция приводит к более простым и быстрым решениям по сравнению с трансляцией, так как решает более частную задачу, не обеспечивая взаимодействия с узлами транзитной сети.

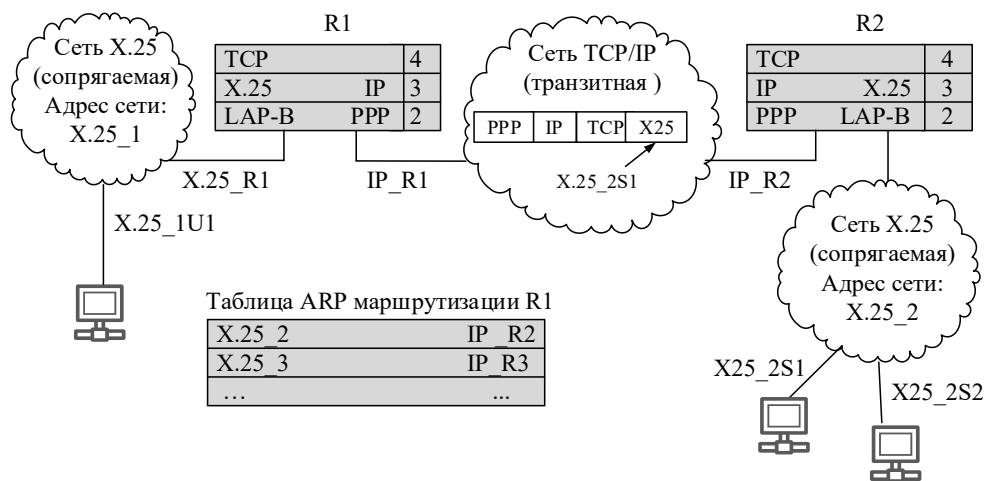
Пример: инкапсуляция на сетевом уровне: X.25 поверх TCP.

Пусть двум сетям X.25 нужно связаться между собой через сеть TCP/IP (рис. 5.1).

В таких случаях сетевой протокол транзитной сети считается протоколом более низкого уровня, чем сетевой протокол объединяемых сетей. Поэтому пакеты сопрягаемых сетей X.25 инкапсулируются в пакеты TCP транспортного уровня транзитной сети TCP/IP пограничным маршрутизатором R1, и переносятся в пакетах TCP по транзитной сети до следующего пограничного маршрутизатора R2. Для переноса по сети TCP/IP пакеты TCP в соответствии с технологией этой сети помещаются в пакеты IP, которые инкапсулируются в кадры протокола канального уровня, например, PPP (Point-to-Point). Маршрутизатор R2 извлекает пакет X.25 из пакета TCP и, предварительно установив виртуальное соединение с узлом назначения по адресу X.25\_2S1, отправляет по этому виртуальному соединению прибывший пакет.

Реализация метода требует процедур нахождения адреса пограничного маршрутизатора R2 в транзитной сети TCP/IP по адресу сети назначения протокола X.25. Такие протоколы называют протоколами разрешения адресов – Address Resolution Protocol, ARP. Такой протокол должен оперировать с ARP-

таблицей (рис. 5.1) Эта таблица содержит для каждого адреса сети назначения X.25 соответствующий IP-адрес пограничного маршрутизатора, через который эту сеть можно достичь.



**Рис. 5.1. Соединение сетей X.25 через транзитную сеть TCP/IP методом инкапсуляции**

### 5.3. Средства согласования протоколов на физическом уровне

Большинство базовых технологий локальных сетей допускает использование различных спецификаций физического уровня в одной сети. Эти спецификации отличаются используемой кабельной системой, а также способом физического кодирования сигналов в кабелях. Например, технология Ethernet имеет 6 вариантов реализации физического уровня: 10Base-5, 10Base-2, 10Base-T, FOIRL, 10Base-FL и 10Base-FB.

Согласование различных физических уровней одной и той же технологии выполняется концентраторами, имеющими порты с приемопередатчиками (трансиверами) различных типов. В стандартах новых технологий для работы с различными вариантами физической среды физический уровень обычно делится на две части: часть, зависящую от физической среды и часть, не зависящую от физической среды. В стандартах детально описывается интерфейс между этими подуровнями, что дает возможность использовать в концентраторах трансиверы третьих фирм.

Концентратор с несколькими портами различного физического уровня реализует метод трансляции протоколов, а компьютер с несколькими сетевыми адаптерами – метод мультиплексирования протоколов.

Иногда концентраторы выполняют и более сложные функции, нежели замена метода физического кодирования сигнала. Например, при объединении физического уровня 100Base-TX и 100Base-T4 в сетях Fast Ethernet концентратор должен выполнять преобразование логического кода 4В/5В в логический

код 8В/6Т. Такой концентратор называется транслирующим. Операция трансляции логических кодов занимает гораздо больше времени, чем простое преобразование электрических импульсов в оптические, как это происходит при объединении сегментов 100Base-TX и 100Base-FX, использующих один и тот же метод логического кодирования 4В/5В. Из-за этого в одном домене коллизий Fast Ethernet допускается использование максимум одного транслирующего концентратора, тогда как нетранслирующих концентраторов может быть два.

#### 5.4. Согласование протоколов канального уровня

По принципу передачи пакетов между сетями с разными канальными протоколами мосты и коммутаторы подразделяются на *инкапсулирующие* (encapsulating) и *транслирующие* (translational).

**Инкапсулирующие мосты и коммутаторы** применяются тогда, когда необходимо соединить два сегмента сети, в которых используется один и тот же канальный протокол, через промежуточную сеть, использующую другой канальный протокол. Максимальный размер инкапсулируемого кадра не должен превышать максимального размера поля данных кадра, в который он вкладывается. Инкапсуляция сейчас редко применяется для объединения локальных сетей с различными канальными протоколами.

**Транслирующие мосты и коммутаторы** выполняют преобразование из одного протокола канального уровня в другой, например, Ethernet в FDDI, Fast Ethernet в Token Ring и т.п. Преобразование заключается в изменении формата кадра, в вычислении нового значения контрольной суммы.

Трансляцию адресной информации в данном случае выполнять не нужно. Все конечные узлы локальных сетей имеют уникальные адреса одного и того же формата, независимо от поддерживаемого протокола.

Единственное отличие: порядок бит в байте, если согласуется сеть Ethernet с сетью Token Ring или FDDI. В сетях Ethernet принята каноническая форма передачи адреса по сети, когда сначала передается самый младший бит самого старшего байта адреса. В сетях FDDI и Token Ring всегда передается сначала самый старший бит самого старшего байта адреса.

Другие возможные операции при согласовании.

- Вычисление длины поля данных кадра и помещение этого значения в поле Length при передаче кадра из сети FDDI или Token Ring в сеть Ethernet 802.3 (в кадрах FDDI и Token Ring поле длины отсутствует).

- Заполнение полей статуса кадра при передаче кадров из сети FDDI или Token Ring в сеть Ethernet. Кадры FDDI и Token Ring имеют два бита – бит распознавания адреса А и бит копирования кадра С.

- Отбрасывание кадров, передаваемых из сетей FDDI или Token Ring в сеть Ethernet с размером поля данных большим, чем 1500 байт.

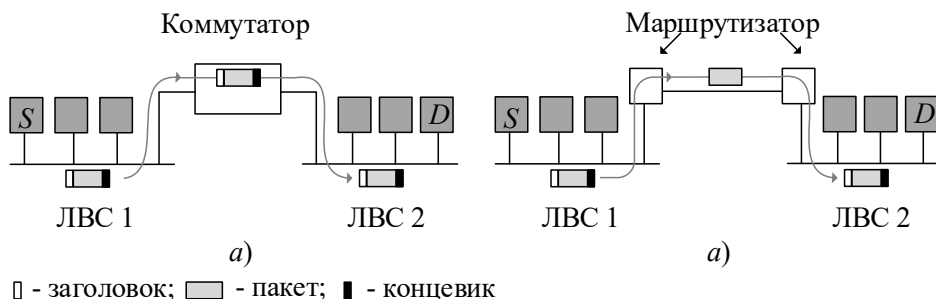
- Пересчет контрольной суммы кадра в соответствии со сформированными значениями служебных полей кадра.

Преимущества трансляции на уровне канальных протоколов: меньше накладные расходы – не нужно передавать два заголовка канального уровня.

Недостатки: транслирующие мосты и коммутаторы вносят дополнительную задержку при преобразовании форматов кадров, при новом вычислении контрольной суммы кадра, максимальный размер кадров у сетей, соединяемых транслирующими мостами и коммутаторами, должен быть одинаковым.

## 5.5. Объединение сетей на сетевом уровне

Объединение сетей на сетевом уровне имеет свои особенности и отличается от объединения на уровне передачи данных. На рис. 5.2, а источник *S* пытается послать пакет приемнику *D*. Эти две машины работают в разных сетях Ethernet, соединенных коммутатором. Источник *S* вставляет пакет в кадр и отправляет его. Кадр прибывает на коммутатор, который по MAC-адресу определяет, что его надо переслать в ЛВС 2. Коммутатор просто снимает кадр с ЛВС 1 и передает его в ЛВС 2.



**Рис. 5.2. Две сети Ethernet, объединенные: а – коммутатором; б – маршрутизаторами**

Допустим, две сети Ethernet объединены не коммутатором, а парой маршрутизаторов (рис 5.2, б). Маршрутизаторы между собой соединены выделенной двухточечной линией. Кадр принимается маршрутизатором, из его поля данных извлекается пакет. Маршрутизатор анализирует содержащийся в пакете адрес (например, IP-адрес). Этот адрес нужно отыскать в таблице маршрутизации. В соответствии с ним принимается решение об отправке пакета (возможно, упакованного в кадр нового вида – это зависит от протокола, используемого линией) на удаленный маршрутизатор. На противоположном конце пакет вставляется в поле данных кадра Ethernet и поступает в ЛВС 2.

В чем заключается основная разница между случаем коммутации (установки моста) и маршрутизации? Коммутатор (мост) пересылает весь кадр, обосновывая свое решение значением MAC-адреса. При применении маршрутизатора пакет извлекается из кадра, и для принятия решения используется адрес, содержащийся именно в пакете. Коммутаторы не обязаны вникать в подробности устройства протокола сетевого уровня. А маршрутизаторы обязаны.

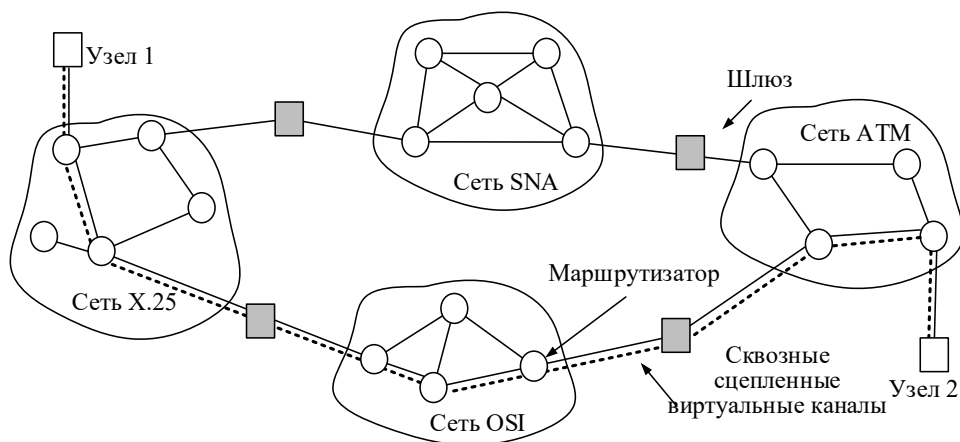
Если в разных частях составной сети используются разные сетевые протоколы, то для того, чтобы она функционировала как единая сеть, все узлы ко-

торой имели бы возможность обмениваться информацией, может быть использован стандартный прием – трансляция.

Специфические особенности, определенные в разд. 5.2, делают задачу трансляции сетевых протоколов нетривиальной, требующей привлечения программных средств. Устройство, реализующее трансляцию одного сетевого протокола в другой, называется шлюзом. (Некоторые шлюзы решают и более сложную задачу согласования стеков протоколов, включающих протоколы всех уровней.)

Оправдано применение на сетевом уровне инкапсуляции. Наибольшую гибкость при инкапсуляции своих пакетов в пакеты других сетевых протоколов демонстрирует протокол IP. Для него разработано семейство протоколов ARP, каждый из которых предназначен для выполнения процедуры инкапсуляции пакетов IP в определенный протокол – Ethernet, X.25, Frame Relay, ATM и т.п.

Туннели могут быть проложены и через несколько последовательно соединенных подсетей. Для этого используется ориентированное на соединение сцепление виртуальных каналов подсетей.



**Рис. 5.3. Объединение сетей с помощью сцепленных виртуальных каналов**

В модели сцепленных виртуальных каналов, показанной на рис. 5.3, соединение с хостом в удаленной сети устанавливается способом, близким к рассмотренному ранее. Подсеть «видит», что адресат является удаленным, и создает виртуальный канал к ближайшему маршрутизатору из сети адресата. Затем строится виртуальный канал от этого маршрутизатора к внешнему шлюзу (многопротокольному маршрутизатору). Шлюз запоминает существование созданного виртуального канала в своих таблицах и строит новый виртуальный канал к маршрутизатору в следующей подсети. Процесс продолжается до тех пор, пока не будет достигнут хост-получатель.

Поскольку речь идет о сцеплении виртуальных каналов, то целесообразно рассмотреть особенности коммутации с их использованием.

## 5.6. Коммутации с использованием техники виртуальных каналов

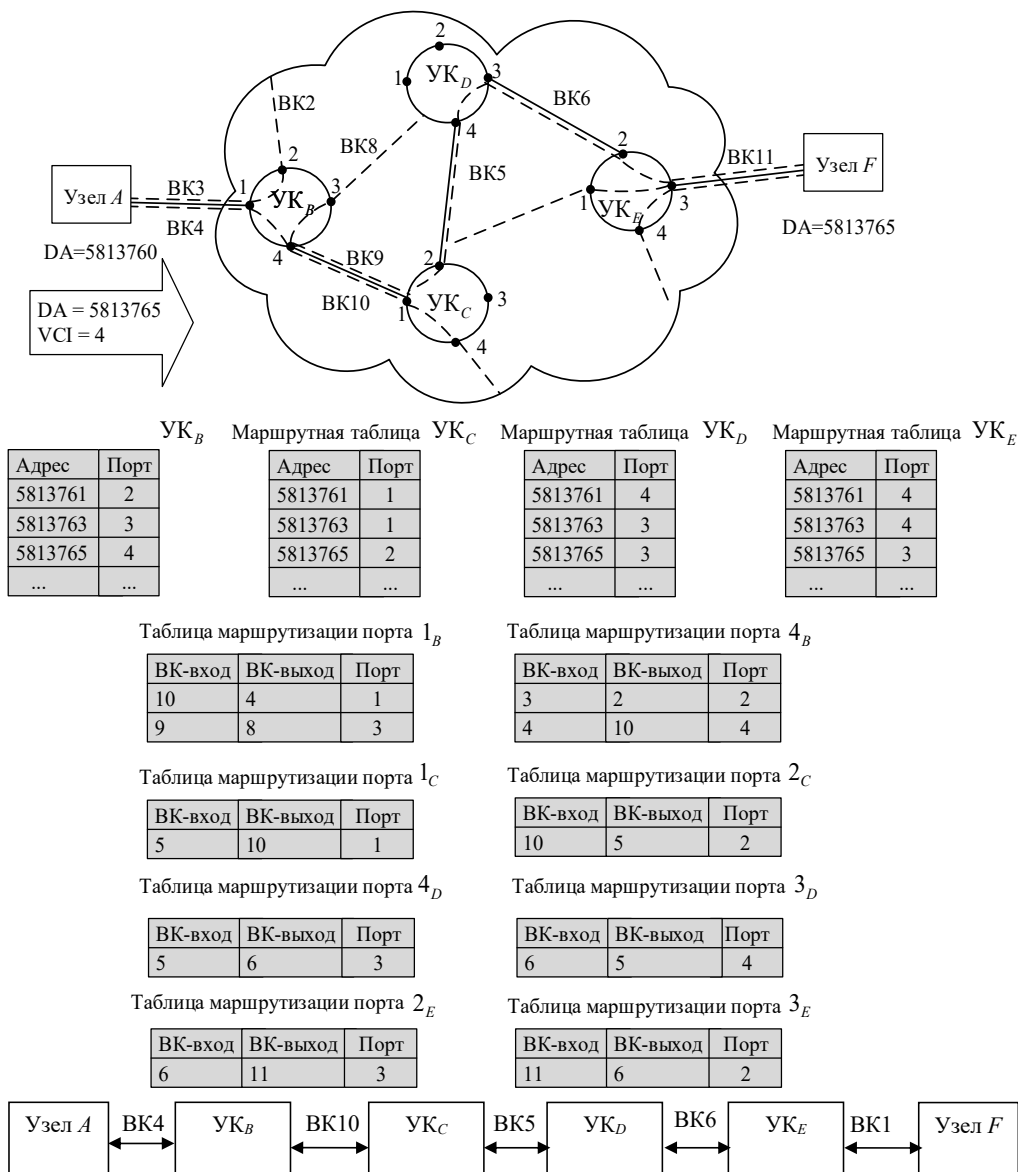
Наряду с маршрутизаторами и пограничными шлюзами для организации виртуальных каналов на сетевом уровне широко используются коммутаторы. При этом при прокладке маршрута для виртуализации проложенного соединения таблицы маршрутизации заменяются таблицами коммутации. Рассмотрим, что же дает применение виртуальных каналов (ВК) в сети.

Смысл создания ВК заключается в следующем. Маршрутизация пакетов между УК сети на основе таблиц маршрутизации происходит только один раз при создании коммутируемого ВК. Для установления виртуального соединения посылается запрос – специальный пакет Call Request, содержащий сетевые адреса источника и получателя. Адреса конечных узлов в глобальных сетях имеют длину 14, 15 десятичных цифр (до 8 байт) в служебном поле пакета. При прохождении Call Request через УК резервируются и затем фиксируются номера входящего и исходящего портов в каждом УК, между которыми будет коммутироваться поток передаваемых данных между корреспондирующими оконечными устройствами в установленном виртуальном соединении. Номер ВК составляет 12 бит (4 бита – номер группы, 8 бит номер – канала в группе). В результате после установления виртуального соединения пакеты не несут длинных адресов, и работа сети маршрутизации пакетов ускоряется за счет двух факторов: решение по продвижению пакетов принимается быстрее из-за меньшего размера таблицы коммутации и уменьшается доля служебной информации.

Режим продвижения пакетов на основе готовой таблицы коммутации портов называют не маршрутизацией, а коммутацией. Таблица маршрутизации в оконечных узлах состоит из записей глобального адреса DA и номера порта, на который нужно переслать пакет из УК, к которому имеет доступ данное оконечное устройство. Адрес следующего коммутатора (УК) не нужен, так как связи между коммутаторами типа «точка – точка», а множественных связей между коммутаторами нет.

Пусть конечный узел с адресом 5813760 начинает устанавливать виртуальное соединение с узлом 5813765 и посылает запрос на установление соединения Call Request. Установление требуемого соединения и формирование коммутационных таблиц на каждом УК, входящем в данный ВК, показаны на рис. 5.4.

Техника ВК позволяет реализовать два режима продвижения пакетов – стандартный режим маршрутизации пакета на основании адреса назначения и режим коммутации пакетов на основании номера ВК. Эти режимы применяются поэтапно, причем первый этап состоит в маршрутизации всего одного пакета – запроса на установление соединения. При использовании виртуальных каналов очень эффективно передаются через сеть долговременные потоки, но для кратковременных этот режим не очень подходит, так как на установление соединения уходит значимое время.



**Рис. 5.4. Установка виртуального канала между оконечными устройствами: А и F;  $Y_i$ ,  $Y = B, C, D, E$  – коммутаторы (УК),  $i = 1, 2, 3, 4$  – номера портов**

## 5.7. Корпоративные сети

Инкапсуляция и техника виртуальных каналов лежат в основе организации виртуальных сетей – VLAN (*Virtual Local Area Network*).



VLAN применяются, например, при построении многих корпоративных сетей. Почему?

Корпорация является сложной многопрофильной структурой и состоит из множества предприятий и организаций, обладающих весьма высокой степенью самостоятельности. В то же время в своей деятельности она ориентируется на вполне конкретные цели. Чтобы обеспечить их достижение, корпорация нуждается в исключительно четко организованной координации деятельности входящих в ее состав предприятий и организаций. Такая координация возможна только на основе эффективной *системы централизованных коммуникаций*.

Концепция корпоративной сети как базовой несущей конструкции современной организации ориентирована на крупномасштабные организации, имеющие распределенную инфраструктуру. Поэтому корпоративная сеть, как правило, является территориально распределенной, т. е. объединяющей офисы, подразделения и другие структуры, находящиеся на значительном удалении друг от друга.

Термин «корпоративная» отражает, с одной стороны, размер сети, так как корпорация – это крупное, большое предприятие. С другой стороны, этот термин имеет смысл объединения, т. е. корпоративная сеть – это сеть, получившаяся в результате объединения нескольких сетей.

С функциональной точки зрения корпоративная сеть – это среда, обеспечивающая передачу информации между различными приложениями, используемыми в системе корпорации. В узком смысле сеть – программно-аппаратный комплекс, организующий надежную и быструю доставку сообщений между взаимодействующими приложениями. Сеть является универсальной транспортной платформой, которая берет на себя выполнение рутинных коммуникационных задач.

Первая проблема, которую приходится решать при создании корпоративной сети, – организация каналов связи. За исключением некоторых крупных ведомств (МЧС, железнодорожный транспорт, газо- и нефтепроводы) территориально распределенные сети используют арендованные линии связи.

В пределах одного города можно рассчитывать на аренду выделенных линий, в том числе высокоскоростных. При переходе к географически удаленным узлам естественным решением этой проблемы является использование уже существующих глобальных сетей. В этом случае достаточно обеспечить каналы от офисов до ближайших узлов глобальной сети. Задачу доставки информации между узлами глобальная сеть при этом возьмет на себя. В качестве такой глобальной сети предприятиями широко используется среда Internet.

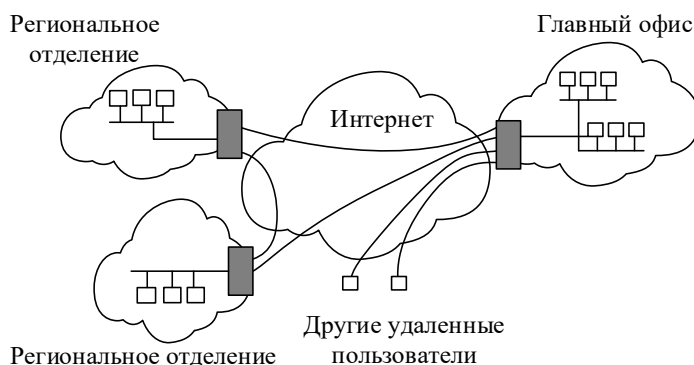
Использование Internet в качестве транспортной среды передачи данных при построении корпоративной сети имеет как преимущества, так и недостатки. К преимуществам относятся готовые коммутационные каналы, низкая абонентская плата, простота реализации. Однако бурный рост услуг, предоставляемых Internet, приводит к перегрузке узлов и каналов связи. Это резко снижает скорость и надежность передачи информации, поэтому в «чистом виде» рекомендовать Internet как основу для систем, в которых требуется надежность и закрытость, никак нельзя. В рамках корпоративной сети Internet находит применение

как *виртуальная частная сеть (VPN)*, представляющая собой защищенные виртуальные каналы сетей пакетной коммутации. Основные достоинства такого подхода – универсальность, гибкость, безопасность.

VPN – обобщенное название технологий, позволяющих реализовывать одно или несколько сетевых соединений (логическую сеть) поверх другой сети.

VPN представляет собой объединение отдельных машин или локальных сетей в виртуальной сети, которая обеспечивает целостность и безопасность передаваемых данных. Она обладает свойствами выделенной частной сети и позволяет передавать данные между компьютерами через промежуточную сеть (internetwork). Главной отличительной чертой данной технологии является использование сети Internet в качестве такой промежуточной магистрали для передачи корпоративного IP-трафика (рис. 5.5).

VPN состоит из «внутренней» (подконтрольной) сети (таких сетей может быть несколько) и «внешней», по которой проходит инкапсулированное соединение (Internet). Возможно также подключение к виртуальной сети отдельного компьютера. Когда данные передаются через VPN, они исчезают «с поверхности» в точке отправки и вновь появляются только в точке назначения. Этот процесс и принято называть «туннелированием».



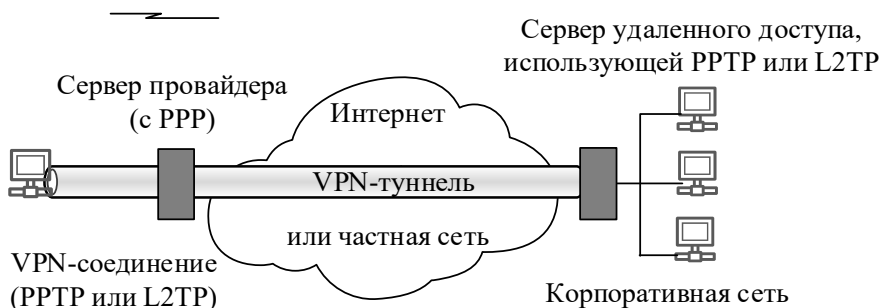
**Рис. 5.5. VPN для корпоративной сети**

При туннелировании протокольные пакеты сети одного типа для передачи вставляются или инкапсулируются в протокольные пакеты («пакеты-переносчики») другой сети. В конце туннеля пакеты деинкапсулируются и передаются получателю.

Логический путь передвижения инкапсулированных пакетов в транзитной сети называется *VPN-туннелем*.

Присоединение удаленного пользователя к VPN производится посредством сервера доступа, который подключен как ко внутренней, так и к внешней (общедоступной) сетям. Существует два типа VPN-подключений: удаленного доступа и типа «сеть-сеть».

В качестве примера на рис. 5.6 показано VPN-подключение удаленного доступа.



**Рис. 5.6. VPN на основе подключения к интернет-провайдеру**

С точки зрения пользователя VPN-подключение представляет собой подключение типа «точка – точка» между клиентским компьютером и сервером организации. Реальная инфраструктура общей или публичной сети не имеет значения. Данные передаются подобно тому, как если бы они передавались по выделенному частному каналу.

Связь через Internet подвержена потенциальным нарушениям защиты и конфиденциальности. Для формирования криптозащищенных VPN-туннелей используются специальные механизмы – протоколы IPSec, PPTP, L2TP.

Каждый криптозащищенный туннель представляет собой соединение, проведенное через открытую сеть. Для защиты от повтора, удаления и задержек пакетов сообщений, передаваемых по туннелю VPN, перегрузки сети и используются возможности стеков сетевых протоколов.

## 5.8. Транспортная сеть

В объединенных сетях роль *транспортной* сети (ТС) выполняет специальная внутренняя (*первичная*) сеть, не имеющая непосредственного выхода на абонентские системы. В первичных сетях сообщения передаются и принимаются посредством передачи и приема электрических сигналов, т. е. с помощью электросвязи. Первичные сети основаны на понятии группового канала и коммутации таких каналов.

### 5.8.1. Распределение группового канала

По физической цепи (линии связи) с помощью каналообразующего оборудования организуются *каналы связи* (КС). В некоторых случаях линия связи и канал связи совпадают. В других случаях узкополосные каналы могут вкладываться (уплотняться) в широкополосный, образуя *групповой канал*. В нем сигналы уплотняемых каналов объединяются, формируя *групповой сигнал*. Организация группового канала на передающей стороне осуществляется с помощью *аппаратуры уплотнения каналов*, а на приемной стороне выделение отдельных каналов из группового выполняется с помощью *аппаратуры разделения кана-*

лов. Создание нескольких каналов на одной линии связи обеспечивается с помощью разнесения их по частоте, времени, длине волны. Ресурс группового канала между парциальными пользователями может быть распределен статически и динамически.

**Статическое распределение канала.** Ресурс КС характеризуется частотной полосой пропускания  $\Delta F$  и временем  $T$ , на которое канал может быть предоставлен для передачи сообщений. Для совместного использования такого канала  $N$  автономными пользователями общую полосу пропускания  $\Delta F$  можно разделить на  $N$  частотных подполос:  $\Delta F_i, i=1, \dots, N, \sum_i \Delta F_i = \Delta F$  – и за каждым

пользователем жестко закрепить отдельную составляющую  $\Delta F_i$ . Такое коллективное использование общего ресурса канала называется **частотным уплотнением канала** – Frequency Division Multiplexing (FDM). Аналогично, если для каждого отдельного канала циклически в жесткой последовательности предоставлять квант времени  $\Delta T_i$ , то такой способ совместного использования общего канала называется **временным уплотнением канала** – Time Division Multiplexing (TDM). В оптоволоконных каналах осуществляется волновое мультиплексирование по длине волны (Wave Division Multiplexing – WDM). При таких методах распределения общего ресурса широкополосного канала конфликтов между пользователями не возникает.

**Динамическое распределение канала.** При большом и меняющемся числе пользователей трафик в сети крайне неравномерен (пульсирующий трафик) и статические методы оказываются неэффективными.

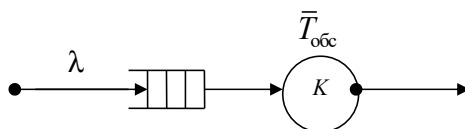
Неравномерность трафика характеризуется, например, коэффициентом пульсаций  $K_{\text{п}}$ :

$$K_{\text{п}} = \frac{\text{Пиковая нагрузка}}{\text{Средняя нагрузка}} = (50 - 100) \text{ и более.}$$

Откажемся от «жесткого» (статического) распределения ресурса канала между пользователями. Будем предоставлять полный ресурс канала каждому отдельному пользователю по требованию с его стороны. Оценим эффективность такого подхода.

Если число каналов меньше числа пользователей, то возможно ожидание доступа и, наоборот, при числе каналов, превышающем число пользователей, возникают простои каналов.

Рассмотрим один канал и его модель в виде экспоненциальной СМО с ожиданием (рис. 5.7).



**Рис. 5.7. Одноканальная однородная экспоненциальная СМО**

Введем обозначения:

$C$  – пропускная способность канала, бит/с;

$1/\mu$  – количество битов в кадре, бит/кадр;

$\lambda$  – интенсивность поступления кадров на вход канала, кадр/с;

Тогда среднюю скорость передачи кадров найдем как  $\frac{C}{1/\mu} = C\mu$ , кадр/с.

Среднее время передачи кадра по каналу  $\bar{T}_{\text{пр}}$  с учетом возможного ожидания есть время пребывания заявки в СМО с ожиданием. Для экспоненциальной одноканальной СМО

$$\bar{T}_{\text{пр}} = \frac{\bar{T}_{\text{обс}}}{1-\rho} = \frac{1}{C\mu} \cdot \frac{1}{1-\rho} = \frac{1}{C\mu} \cdot \frac{1}{1-\lambda/C\mu} = \frac{1}{C\mu - \lambda}.$$

Пусть  $C=100$  Мбит/с;  $1/\mu=10000$  бит/кадр;  $\lambda=5000$  кадр/с, тогда  $\bar{T}_{\text{пр}} = 200$  мкс. Если не учитывать ожидание в очереди, то для передачи кадра потребовалось бы 100 мкс.

Теперь разделим канал на  $N$  подканалов. Соответственно, у каждого подканала пропускная способность есть  $C/N$  бит/с. Интенсивность поступления кадров на вход отдельного канала равна  $\lambda/N$  кадр/с. Тогда среднее время передачи кадра  $\bar{T}_{\text{пр}}^{FDM}$  при частотном уплотнении канала

$$\bar{T}_{\text{пр}}^{FDM} = \frac{1}{\mu(C/N) - \lambda/N} = \frac{1}{\mu C - \lambda} = N\bar{T}_{\text{пр}}.$$

Следовательно, при FDM значение средней задержки стало в  $N$  раз больше в случае, если бы все кадры могли быть организованы в одну общую очередь на входе группового (широкополосного) канала. Те же аргументы применимы и к временному (TDM) методу уплотнения широкополосного канала.

Если разделить 100-мегабитную сеть физически на 10 мегабитных сетей, то средняя задержка возрастет с 200 мкс до 2 мс.

Вывод: ни один статический метод распределения широкополосного (широкополосного) канала не годится при пульсирующем трафике.

Однако если при статическом распределении ресурса группового канала адресом каждого абонентского канала является либо полоса  $\Delta F_i$  при FDM, либо местоположение слота времени  $\Delta T_i$  на повторяющихся циклах при TDM, то при динамическом распределении ресурса группового канала каждой структурной единице передаваемых абонентских сообщений (кадру, пакету) требуется приписывать соответствующий идентификатор (адрес). Это дополнительные «накладные» расходы при динамическом распределении ресурса канала. Тем не менее, при пульсирующем трафике такой подход эффективен.

Статическое распределение ресурса канала используется при построении магистральных каналов первичных сетей, динамическое – в компьютерных се-

тах. Например, в Ethernet используется метод коллективного доступа к среде передачи с опознаванием несущей и обнаружением коллизий (см. разд. 3.4.1).

## 5.8.2. Первичные сети

*Первичные* сети являются основой вторичных компьютерных и телефонных сетей. Обеспечение качества обмена в телекоммуникационных сетях зависит от производительности первичных сетей. Такие сети называют также *опорными* и *базовыми*. Современные первичные сети основаны на коммутации каналов. Для создания абонентского канала коммутаторы первичных сетей поддерживают один из методов статического мультиплексирования. К настоящему времени сформировалось два поколения таких первичных цифровых сетей:

- плезиохронная цифровая иерархия (Plesiochronous Digital Hierarchy, PDH);
- синхронная цифровая иерархия (Synchronous Digital Hierarchy – SDN). В США технология SDN называется SONET.

### **Технология плезиохронной цифровой передачи.**

Плезиохронная цифровая иерархия предназначалась для передачи голосовой информации через каналы связи. «Плезо» означает «почти», т.е. почти синхронная передача. Для мультиплексирования абонентских каналов в PDH используются:

- техника частотного мультиплексирования – FDM;
- техника временного мультиплексирования – TDM;
- мультиплексирование по длине волны – WDM.

Цифровые представления аналоговых сигналов осуществляются с использованием индивидуальных ИКМ-кодеков. Сигнал ограничивается полосой частот 4 кГц; осуществляется его дискретизация с интервалом 125 мс ( $f_d = 8$  кГц); отсчеты сигнала квантуются по уровню; 8 бит используется для кодирования каждого отсчета. В итоге и получают стандартную скорость передачи на один телефонный канал в 64 кбит/с.

Данная технология использует статический способ образования группового канала. Группообразование в каналообразующей аппаратуре PDH осуществляется на основе посимвольного мультиплексирования составляющих потоков.

При применении этой технологии формируется иерархия из цифровых каналов (Digital Stream, DS), каждому из которых назначен уровень и номер. Цифровые потоки с меньшими номерами мультиплексируются в потоки с большими с определенным сдвигом частоты.

В настоящее время в мире применяются европейская, американская и японская иерархии PDH, использующие основной цифровой канал (ОЦК) со скоростью 64 кбит/с.

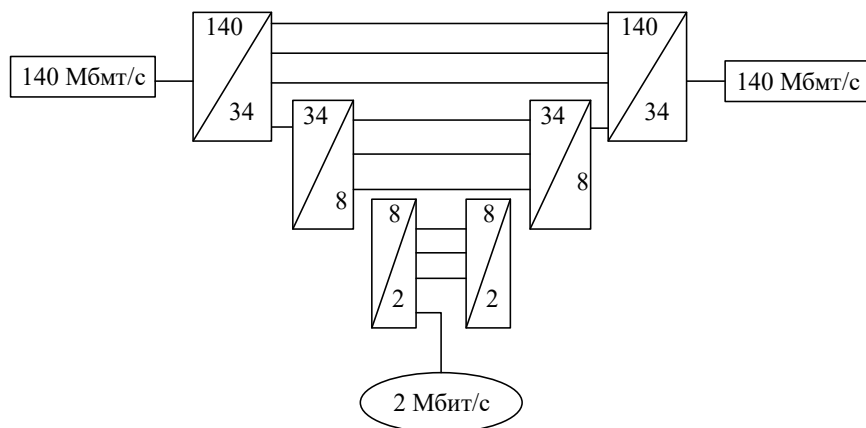
В европейской PDH первичная система передачи E1 со скоростью 2,048 кбит/с образуется объединением 32-канальных интервалов (30 оперативных, 2 служебных), периодически повторяющихся в виде цикла протяженностью 125 мс.

Дальнейшее развитие привело к появлению еще ряда стандартизированных систем E2, E3, E4, E5. Вторичная система передачи E2 со скоростью 8448 кбит/с образуется объединением четырех цифровых сигналов E1; третичная система передачи E3 со скоростью 34 368 кбит/с формируется объединением четырех цифровых сигналов E2; четверичная система E4 со скоростью 139 264 кбит/с – объединением четырех цифровых сигналов E3 и пятеричная система E5 со скоростью 564 992 кбит/с – объединением четырех цифровых сигналов E4.

В США, Канаде, Японии применяется стандарт T1. В линии T1 собираются вместе 24 цифровых канала по 64 Кбит/с, т. е. в итоге пропускная способность составляет 1.544 Мбит/с. Четыре канала T1 объединяются в канал следующего уровня – T2 (6.312 Мбит/с). Семь каналов T2 образуют канал T3 (44.736 Мбит/с) и т. д.

На практике используются в основном каналы: T1/E1 и T3/E3.

Каналы PDH, основанные на согласовании «скоростей», оказались негибкими. При вводе и выводе из общего асинхронного потока приходится выполнять большое число операций мультиплексирования и демultipлексирования. Пример подобных операций показан на рис. 5.8.



**Рис. 5.8. Схема вывода и ввода потока 2 Мбит/с из четверичного потока 140 Мбит/с**

При нарушении синхронизации группового сигнала PDH восстановление синхронизации первичных цифровых потоков занимает существенное время. Затруднен контроль, обеспечивающий функционирование сети с требуемым качеством. Отсутствует единая синхронизация для большой сети. Плезиохронная сеть удобна для строительства отдельных каналов, но вызывает лишние сложности при создании глобальных сетей. Более современные технологии практически полностью вытеснили PDH с оптических коммуникаций. В этой ситуации удачным решением стала разработанная в 1980-х годах синхронная оптическая сеть SONET и синхронная цифровая иерархия SDH, которые часто

рассматриваются как единая технология SONET/SDH. Однако на медных кабелях PDH используется как для телефонии, так и для передачи данных.

### Синхронная цифровая иерархия.

Синхронная цифровая иерархия задумана как скоростная информационная автострада для транспортировки цифровых потоков с разными скоростями (от единиц мегабит до десятков гигабит в секунду). Эта сеть явилась развитием технологии PDH (T1/T3, E1/E3). Основная область применения – первичные сети операторов связи.

Преобразование и передача данных в этой системе достаточно сложны, в рамках данного пособия рассматриваться не будут. Отметим лишь несколько отличительных моментов.

Сети SDH – сети с коммутацией каналов. Для уплотнения применяется мультиплексирование с разделением времени TDM. В иерархии объединяются и разъединяются цифровые потоки со скоростями 155.520 Мбит/с (базовый уровень скорости) и выше. Способ объединения – синхронный.

Первый уровень иерархии SDH известен как STM-1 (Synchronous Transport Module), который передается со скоростью 155.52 Мбит/с.

STM-1 представляет собой фрейм (кадр) размером  $9 \cdot 270 = 2430$  байт с периодом повторения 125 мкс (рис. 5.9).



**Рис. 5.9. Формат кадра STM-1**

Первые 9 байт каждой строки отводятся под служебные данные заголовков, а из остальных 261 байта – 260 предназначаются для полезной информации (данных), а 1 байт используется для заголовка тракта, что позволяет контролировать соединение из конца в конец.

Чтобы определить маршрут транспортного модуля, в левой части рамки записывается секционный заголовок (Section Over Head, SON), состоящий из двух частей: RSON – секционный заголовок регенератора, где будет осуществляться восстановление потока, поврежденного помехами, и MSON – секционный заголовок мультиплексора, в котором транспортный модуль будет преформатироваться. Указатель (Pointer, PTR) определяет начало записи полезной нагрузки.

В стандарте SDH все уровни скоростей (и, соответственно, форматы кадров для этих уровней) имеют общее название STM-*n*. Иерархия скоростей кратна скорости STM-1. Так, из четырех модулей STM-1 побайтным мультиплекси-



рованием формируется модуль STM-4 (скорость 622.080 Мбит/с), четыре модуля STM-4 образуют модуль STM-16 (2488.32 Мбит/с), ..., STM-256 (39.81 Гбит/с).

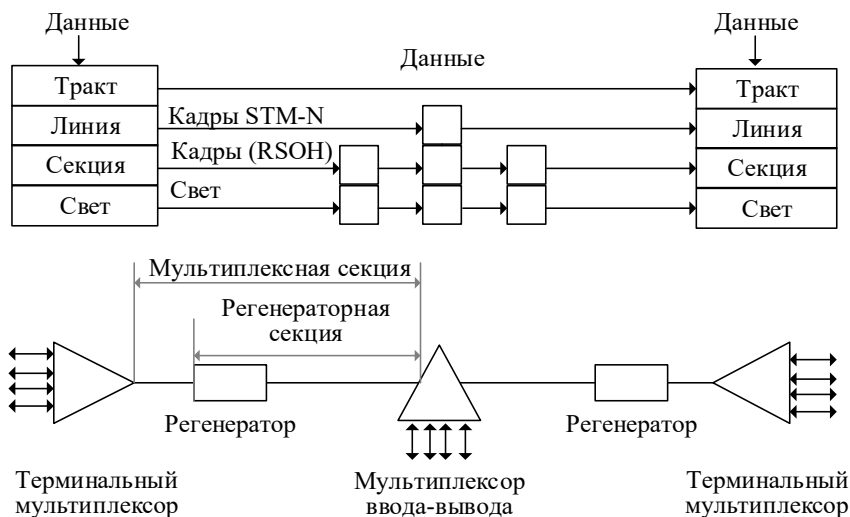
В сети SDH применяется принцип контейнерных перевозок. Подлежащие транспортировке сигналы размещаются в стандартных *виртуальных* контейнерах (*Virtual Container – VC*), которые позволяют переносить через сеть блоки PDH.

В технологии SDH стандартизовано шесть типов виртуальных контейнеров, которые хорошо сочетаются друг с другом при образовании кадра STM-*n*. Существует ряд правил, по которым контейнеры одного типа могут образовывать группы контейнеров, а также входить в состав контейнеров более высокого уровня. Информация адресуется путем временного положения внутри составного кадра.

Операции с контейнерами производятся независимо от их содержания. Этим достигается прозрачность сети SDH – способность транспортировать различные структуры сигналов, в частности PDH.

Наличие большого числа указателей (PTR) позволяет четко определять местонахождение в модуле STM-*n* любого цифрового потока. Это означает, что базовый канал со скоростью 64 кбит/с может быть выделен напрямую из уровней высшей иерархии SDH, и наоборот. Мультиплексор SDH выделяет необходимые составляющие сигнала, не разбирая весь поток. По сравнению с PDH-технологией SDH позволяет разрабатывать более гибкую структуру сети и избежать использования большого числа дорогих мультиплексирующих и демультиплексирующих устройств.

Стек протоколов SDH включает четыре уровня протоколов (рис. 5.10).



**Рис. 5.10. Стек протоколов технологии SDH**

**Физический уровень** (фотонный) имеет дело с кодированием битов. Используется потенциальный код без возвращения к нулю (Non Return to Zero, NRZ).

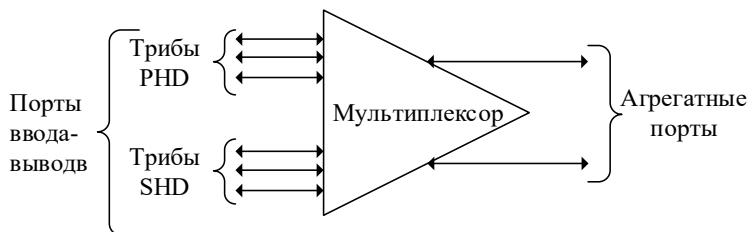
**Уровень секции** поддерживает физическую целостность системы. Секция – непрерывный отрезок оптоволоконного кабеля, который соединяет пару устройств SONET/SDN (регенератор, мультиплексор). Секцию часто называют регенераторной секцией. Ее протокол имеет дело с определенной частью заголовка кадра, называемой заголовком регенераторной секции (RSON).

**Уровень линии** отвечает за передачу данных между мультиплексорами. Протокол работает с кадрами STM-*n*, выполняя мультиплексирование и демultipлексирование, вставку и удаление пользовательских данных. Этот протокол отвечает также за проведение операций по реконфигурированию линии в случае отказа какого-либо ее сегмента – оптоволоконка, порта или соседнего мультиплексора. Линию называют и мультиплексорной секцией.

**Уровень тракта** отвечает за доставку данных между двумя конечными пользователями сети. Тракт (путь) – это составное виртуальное соединение между пользователями. Протокол тракта должен принимать данные, поступающие в пользовательском формате (например в формате E1) и последовательно преобразовывать их в синхронные кадры STM-*n*.

По своему составу и принципам функционирования транспортная сеть представляет собой совокупность пунктов ввода отдельных цифровых потоков, линий передачи с регенераторами и мультиплексорами.

Мультиплексор SDH имеет две группы интерфейсов: *пользовательскую* и *агрегатную* (рис. 5.11).



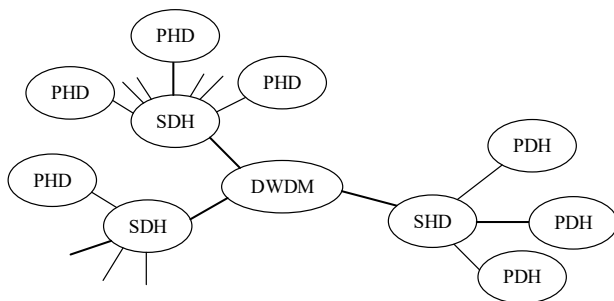
**Рис. 5.11. Структура мультиплексора SDH**

Первая группа предназначена для создания пользовательской структуры (порты PDH), вторая (SDH) – линейных межузловых соединений. Эти интерфейсы позволяют реализовать три топологии: «кольцо», «цепочку» и «ячейку». На их основе можно строить сеть мультиплексоров практически любого масштаба. В идеале такая сеть состоит из нескольких уровней. На первом осуществляется доступ пользователей к сети, которые через согласующие устройства подключаются к мультиплексорам первого уровня. На данном уровне используются, как правило, мультиплексоры STM-1. Второй уровень построен на мультиплексорах STM-4 и отвечает за сбор потоков информации от первого уровня. Третий уровень выполняет транспортные функции и строится на мультиплексорах STM-16. Он собирает потоки информации от второго уровня и транспортирует их далее.

Кроме мультиплексоров в состав сети SDH могут входить регенераторы. Они позволяют бороться с затуханием сигнала. Выполняется преобразование: свет → электрический ток → усиление → свет.

Устойчивая работа SDH-сети обеспечивается иерархией синхронизирующих источников.

Сети SDH интегрируются с сетями DWDM, обеспечивая передачу информации по оптическим магистралям со скоростями сотни гигабит в секунду за счет мультиплексирования с разделением по длине волны (рис. 5.12).



**Рис. 5.12. Иерархия базовых сетей и сетей доступа**

В сетях DWDM сети SDH выступают как сети доступа, т.е. выполняют ту же роль, что по отношению к ним сети PDH.

В настоящее время сети SDH составляют фундамент практически всех крупных телекоммуникационных сетей – региональных, национальных и международных.

### **Контрольные вопросы**

1. Почему возникает необходимость в объединении сетей?
2. Каковы особенности согласования протоколов на физическом уровне?
3. Чем отличается объединение сетей на физическом уровне от объединения на канальном уровне?
4. Каковы особенности объединения сетей на сетевом уровне?
5. В чем заключается достоинство объединения сетей посредством мультиплексирования протоколов?
6. В каких ситуациях применяется трансляция протоколов?
7. Какие очевидные недостатки трансляции?
8. Сопоставьте достоинства и недостатки мультиплексирования и трансляции протоколов.
9. В чем особенность инкапсуляции протоколов?
10. Когда используются связанные виртуальные каналы?
11. В чем отличается таблица коммутации от таблицы маршрутизации?
12. Чем обусловлена фрагментация пакетов?
13. Можно ли посредством туннелирования объединить сети Ethernet и X.25?
14. Какие преобразования выполняет сеть-посредник при инкапсуляции протоколов?
15. Какой путь называют VPN-туннелем?

## 6. БЕСПРОВОДНЫЕ ЛОКАЛЬНЫЕ СЕТИ. ИНТЕРНЕТ ВЕЩЕЙ

Беспроводная локальная сеть (англ. *Wireless Local Area Network*, **WLAN**) – локальная сеть, в которой объединение устройств в сеть происходит без использования кабельных соединений, и передача данных осуществляется через радиоэфир.

Беспроводные локальные сети (БЛС) популярны в силу ряда своих достоинств: мобильность, создание временных сетей, работа на местности, где прокладка обычных кабелей затруднена.

Применение БЛС оправдано:

- в строениях с большими открытыми участками – фабрики, торговые залы фондовых бирж и супермаркетов, складские помещения;
- в исторических зданиях, где прокладка кабелей запрещена;
- в офисах, где прокладка кабелей экономически не выгодна;
- при расширении ЛС – объединение кабельной сети и беспроводной (связь с ноутбуками).

Наиболее распространенными на сегодняшний день являются Wi-Fi, Bluetooth и WiMAX, стандарты IEEE 802.11, IEEE 802.15 и IEEE 802.16 соответственно.

### 6.1. Топологии беспроводных локальных сетей

#### 6.1.1. Стандарт IEEE 802.11

Стандарты серии IEEE 802.11, известные как Wi-Fi, относятся к созданию беспроводных локальных сетей WLAN. Оборудование Wi-Fi является наиболее массовым и в радиусе его действия позволяет обеспечить доступ к сети Интернет. На сегодняшний день действуют следующие поправки к стандарту – 802.11a, 802.11b, 802.11g и 802.11n, которые различаются пропускной способностью и используемым частотным диапазоном.

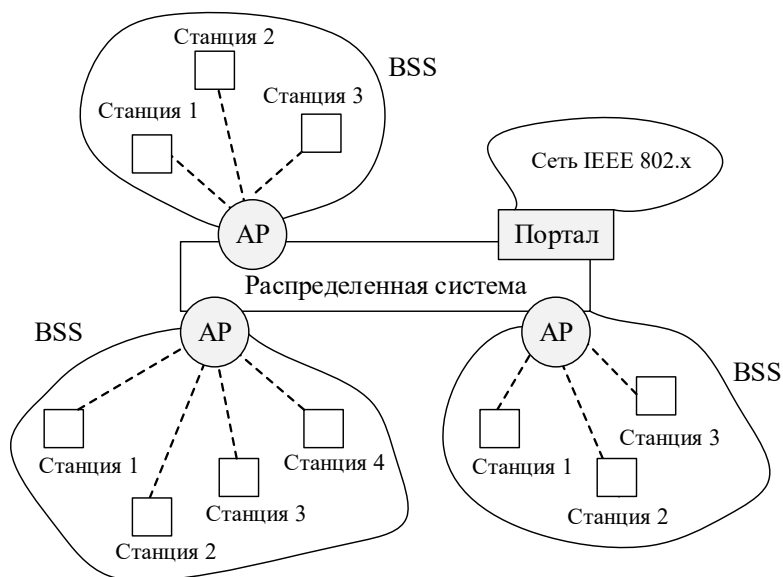
Стандарт 802.11 поддерживает два типа топологий локальных сетей: с базовым и с расширенным наборами услуг.

Сеть с базовым набором услуг (Basic Service Set, BSS) образуется отдельными станциями, базовая станция отсутствует, узлы взаимодействуют друг с другом непосредственно (рис. 6.1). Для того чтобы войти в сеть BSS, станция должна выполнить процедуру присоединения.

*Базовый набор служб* (Basic Service Set, BSS) – несколько станций, выполняющих одинаковый протокол MAC. Станции конкурируют за доступ к носителю.

BSS может быть изолирован или соединяться с *распределительной системой* (Distribution System, DS) через точку доступа (Access Point, AP) (подобна мосту). Протокол MAC может быть распределенным или централизованным, управляемым из AP.

Распределительная система может быть коммутатором, кабельной или беспроводной сетью.



**Рис. 6.1. Модель расширенного набора служб**

Станции могут отключаться, выходить из зоны приема и входить в зону приема.

Зона обслуживания BSS – Ø 100–300 м. Наборы BSS могут перекрывать-ся, а могут находиться друг от друга на значительном расстоянии. Станция из одного набора BSS может переходить в другой набор BSS.

Стандартом IEEE 802.11 определена многоуровневая архитектура протоколов. На нижнем физическом уровне определяются частотный диапазон, скорость передачи данных. Над ним располагается каналный уровень.

Канальный уровень 802.11 состоит из двух подуровней: управления логической связью (Logical Link Control, LLC) и управления доступом к носителю (Media Access Control, MAC).

Стандарт 802.11 использует тот же LLC и 48-битовую адресацию, что и другие сети 802. Это позволяет легко объединять беспроводные и проводные сети.

Как и для Ethernet сетей 802.3, MAC уровень 802.11 поддерживает множественный доступ на общем носителе, когда пользователь проверяет носитель перед доступом к нему. Однако стандарт 802.11 предусматривает использование полудуплексных приемопередатчиков. Поэтому, в отличие от беспроводных в сетях 802.11 станция не может обнаружить коллизию во время передачи (станция себя «не слышит» во время передачи).

Чтобы учесть это отличие, 802.11 использует модифицированный протокол CSMA/CA. Для определения того, является ли канал свободным, используется специальный алгоритм оценки чистоты канала, а чтобы убедиться, что

коллизий не произошло, принимающая станция посылает пакет АСК для подтверждения того, что сообщение получено неповрежденным.

Из-за дешевизны и простоты установки Wi-Fi часто используется для предоставления клиентам быстрого доступа в Интернет различными организациями. Например, во многих кафе, отелях, вокзалах и аэропортах можно обнаружить бесплатную для посетителей точку доступа Wi-Fi.

### 6.1.2. Стандарт IEEE 802.16

Стандарты серии IEEE 802.16, известные как Wi-MAX, относятся к организации беспроводных сетей масштаба города (Wireless Metropolitan Area Network, WMAN). Оборудование Wi-MAX позволяет обеспечить доступ к сети Интернет на значительной территории – до нескольких километров.

Если 802.11 – мобильный аналог Ethernet, то 802.16 – беспроводной стационарный аналог кабельного телевидения (широкополосные беспроводные сети).

WiMAX (англ. *Worldwide Interoperability for Microwave Access*) – телекоммуникационная технология, разработанная с целью предоставления универсальной беспроводной связи на больших расстояниях для широкого спектра устройств (от рабочих станций и портативных компьютеров до мобильных телефонов).

В общем виде WiMAX сети состоят из базовых и абонентских станций и оборудования, связывающего базовые станции между собой, с устройством пользователя, с поставщиком сервисов и с Интернетом. При этом, по крайней мере, одна базовая станция подключается к сети провайдера с использованием классических проводных соединений.

Основной задачей базовой станции является установление, поддержание и разъединение радиосоединений. Кроме того, базовая станция выполняет обработку сигнализации, а также распределение ресурсов среди абонентов.

В технологии 802.16 различают фиксированный и мобильный WiMAX. Фиксированный WiMAX позволяет обслуживать только «статичных» абонентов, а мобильный ориентирован на работу с пользователями, передвигающимися со скоростью до 150 км/ч. Мобильность означает наличие функций роуминга и «бесшовного» переключения между базовыми станциями при передвижении абонента (как происходит в сетях сотовой связи). В частном случае мобильный WiMAX может применяться и для обслуживания фиксированных пользователей.

Технологии семейства 802.16 позволяют экономически более эффективно (по сравнению с проводными технологиями) не только предоставлять доступ в сеть новым клиентам, но и охватывать новые труднодоступные территории. Кроме того, WiMAX обеспечивает доступ в Интернет на высоких скоростях с гораздо большим покрытием, чем у Wi-Fi-сетей. Это позволяет использовать технологию в качестве «магистральных каналов», продолжением которых выступают локальные сети. Подобный подход способствует созданию масштабируемых высокоскоростных сетей в рамках городов.

Архитектура сетей WiMax не привязана к какой-либо определенной конфигурации. Она обладает высокой гибкостью и масштабируемостью.

WiMAX и Wi-Fi сети просты в развертывании и по мере необходимости легко масштабируемы.

Однако, современным технологиям беспроводных сетей присущи существенные ограничения частотно-временных радиоресурсов. Международными комитетами по стандартизации ведется напряженная работа с целью принятия новых спецификаций, определяющих работу гетерогенных сетей связи.

Помимо услуг мобильного доступа, которые предполагают передачу разнородного пользовательского трафика, гетерогенные сети должны обеспечить поддержку множества приложений Интернета вещей, основанных на принципе межмашинного взаимодействия.

### **6.1.3. Стандарт IEEE 802.15**

Стандарты серии IEEE 802.15, известные как Bluetooth, относятся к организации беспроводных персональных сетей (Wireless Personal Area Network, WPAN). Оборудование Bluetooth реализует обмен данными между различными устройствами на небольшие расстояния до нескольких десятков метров.

Небольшой радиус действия устройств с Bluetooth-интерфейсом позволяет развернуть работу WPAN на ограниченной площади, например в рамках квартиры, офисного рабочего места.

IEEE 802.15 описывает только два нижних уровня модели OSI: физический (PHY) и уровень доступа к среде передачи (MAC).

В семейство IEEE 802.15 входят несколько стандартов, число которых постоянно пополняется благодаря популярности WPAN в различных приложениях – сетях промышленной автоматизации, системах «умного» дома, нательных медицинских сетях и других. Если обобщить, то стандарт IEEE 802.15 определяет спецификации Bluetooth, характеристики физических устройств, на которых строится беспроводная персональная сеть, и методы доступа этих устройств к среде передачи.

## **6.2. Самоорганизующаяся беспроводная сеть**

Самоорганизующаяся беспроводная сеть не имеет определенной структуры, а функции узлов не фиксированы. Каждый раз, когда к сети подключается новое устройство, происходит перераспределение функций между узлами сети и меняются характеристики каналов связи.

Цель создания первых самоорганизующихся сетей заключалась в возможности работать и получать доступ к сети Интернет в любом месте, даже в движении, не полагаясь на инфраструктуру фиксированной сети. В настоящее время это новый тип сетей.

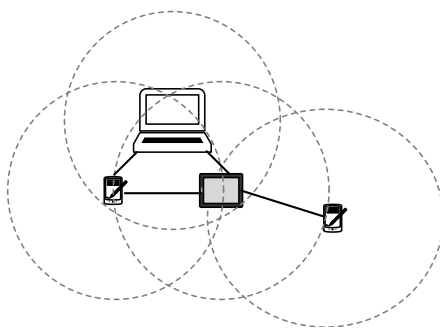
Корень «сам» в названии «самоорганизующиеся» в сетях подобного типа говорит о том, что все функции по конфигурированию, предоставлению досту-

па к каналу, устранению сбоев в работе, по организации передачи данных сеть выполняет самостоятельно.

Узлами самоорганизующейся сети могут быть любые устройства, находящиеся в пределах действия радиосигнала и оснащенные беспроводным сетевым адаптером или интерфейсом Bluetooth. Такой сетью можно организовать временный обмен данными между сотовыми телефонами, планшетами или ноутбуками.

Самоорганизующиеся сети классифицируются на целевые (ad hoc) и ячеистые (mesh) сети.

В сети ad hoc (от лат. «специально для этого случая») обмен данными происходит напрямую без промежуточных звеньев. Это так называемая одноранговая (пиринговая) коммуникация, не требующая наличия инфраструктуры локальной сети или прочих «распределителей» (рис. 6.2).



**Рис. 6.2. Пример организации ad hoc беспроводной сети**

Сети ad hoc находят применение преимущественно для мобильной передачи данных.

Объединение двух сотовых телефонов через Bluetooth в сеть ad hoc осуществляется достаточно просто. Для этого оба участника сети активируют функцию Bluetooth, создают через нее соединение между телефонами и после этого могут обмениваться файлами.

Объединение двух компьютеров в сеть ad hoc выполняется немного сложнее. Главное условие – наличие у компьютеров соответствующего интерфейса – сетевой карты WLAN или Wi-Fi USB-адаптера для беспроводных сетей. Сначала производятся настройки, одинаковые для всех компьютеров: режим «802.11 Ad Hoc», номер канала и идентификатор BSS. Последующий поиск канала происходит автоматически. После создания подключения все компьютеры в сети получают одинаковые права и имеют доступ к базам данных и каталогам друг друга.

В профессиональных сетях ad hoc конечные устройства объединяются в одну информационную систему на больших расстояниях с использованием технологии GPS<sup>5</sup>.

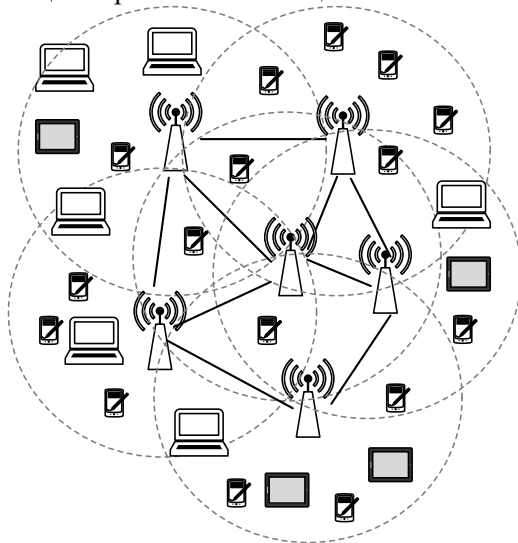
---

<sup>5</sup> **GPS** (*Global Positioning System*, система глобального позиционирования) – спутниковая система навигации, обеспечивающая измерение расстояния, времени и определяющая местоположение во всемирной системе координат



Mesh-сети (от англ. «ячейка») строятся по принципу ячеистой топологии, в которой каждая ячейка – это стационарный радиоузел, выполняющий функции маршрутизатора (рис. 6.3). Увеличение зоны действия mesh-сети достигается добавлением маршрутизаторов.

Настойка mesh-сети достаточно сложна, но это оправдывается высокой отказоустойчивостью, которую обеспечивает ячеистая топология. Каждый узел mesh-сети соединен с несколькими соседями, поэтому всегда есть широкий выбор маршрута следования трафика внутри сети. Обрыв одного соединения не нарушает функционирования сети в целом.



**Рис. 6.3. Пример организации беспроводной mesh-сети**

Топология mesh-сети относительно постоянна, отсюда и маршрут движения трафика редко меняется. Только в случаях внезапного отключения или добавления новых узлов могут быть инициированы процессы построения новых маршрутов.

Принцип транспортировки в mesh-сетях во многом напоминает проводные сети – данные передаются от одного узла к другому до тех пор, пока пакет не достигнет получателя. Промежуточные узлы не только усиливают сигнал, но и осуществляют переадресацию данных в соответствии с маршрутной таблицей.

Беспроводные mesh-сети применяются для решения широкого спектра задач – мониторинг, телеметрия объекта и другие. Особенно mesh-сети нашли применение при работе в условиях неблагоприятной окружающей среды.

В последнее время все большее распространение приобретают сенсорные сети, которые являются частным случаем беспроводных самоорганизующихся сетей. Однако, по своему назначению, параметрам, спецификациям сенсорные сети существенно отличаются от сетей связи WiFi, WiMax и Bluetooth.

### 6.3. Сенсорные сети

Беспроводная сенсорная сеть (Wireless Sensor Networks, WSN) – это распределенная сеть множества сенсорных и исполнительных устройств, объединенных между собой посредством радиоканала.

Источниками данных в сенсорной сети являются датчики (сенсоры), собирающие информацию об окружающей среде – температуре, влажности, освещении, концентрации вредных примесей и т. д. Для обработки и передачи этой информации датчик интегрируют с микроконтроллером, вместе они образуют сенсорное устройство (СУ).

Основная идея беспроводной сенсорной сети (БСС) – отказ от непосредственного участия человека в сборе информации. Это позволяет развернуть работу БСС в конкретном месте или при реализации технологического процесса.

БСС функционируют на базе стандарта IEEE 802.15.4 (ZigBee). Протокол ZigBee был разработан для объединения в сеть большого количества автономных устройств, например датчиков и выключателей с батарейным питанием.

Спецификация IEEE 802.15.4 описывает: гарантированную безопасность передачи данных при относительно небольших скоростях, возможности длительной работы сетевых устройств от автономных источников питания (батарей), возможность выбора алгоритма маршрутизации, механизмы стандартизации приложений пользователей.

#### 6.3.1. Узлы беспроводной сенсорной сети

БСС состоит из узлов следующего типа:

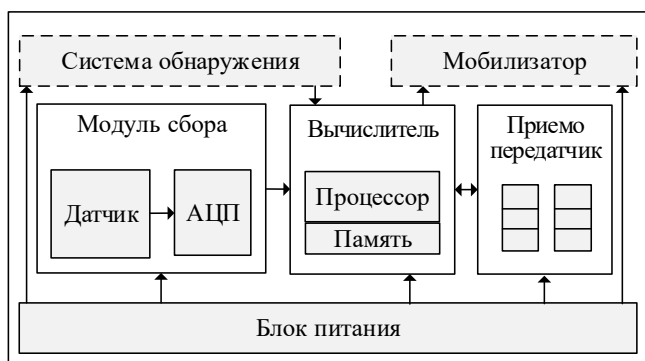
1. СУ, образующие сенсорное поле в месте действия сети;
2. исполнительные устройства, воздействующие на контролируемый сенсорными устройствами объект;
3. базовые станции (БС), выполняющие роль концентратора, маршрутизатора или шлюза в зависимости от возложенных на них функций.

Типичная структура СУ приведена на рис. 6.4. Она включает один или несколько датчиков, а также модули, позволяющие СУ самостоятельно проводить начальную обработку измеренных данных и поддерживать взаимодействие с БСС.

Модуль сбора СУ преобразует измерения датчика из аналогового вида в цифровой код с помощью аналого-цифрового преобразователя (АЦП). Вычислитель реализует отбраковку измерений, в результате чего остаются только полезные данные.

Приемопередатчик образует радиointерфейс СУ с БСС и содержит порты для приема и передачи данных.

При необходимости в структуру СУ могут быть встроены система обнаружения (система глобального позиционирования GPS) и мобилизатор (ориентация антенны при изменении местоположения или конфигурации).



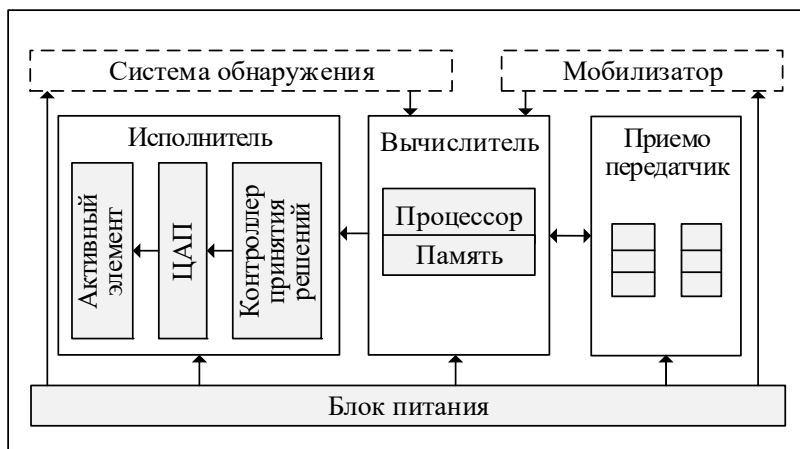
**Рис. 6.4. Структурная схема СУ**

Блок питания обеспечивает энергией работу СУ. Время «жизни» СУ зависит от срока службы элементов питания (часто обычные батареи). Также можно использовать подзаряжаемые, например солнечной энергией, батареи.

Размер корпуса СУ может быть от одного до нескольких десятков кубических сантиметров.

На базе СУ строятся сенсорные сети мониторинга контролируемых объектов, например, состояния охраняемой территории, окружающей среды и т.п. Как правило, такие сети являются *однородными*, то есть все СУ в сети одинаковы с точки зрения затрат энергии батареи и функциональных возможностей.

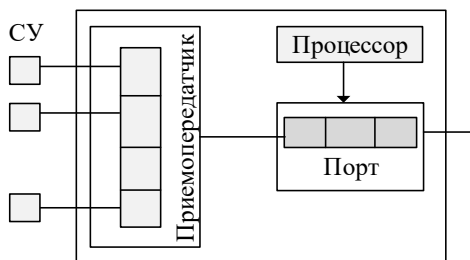
В медицинских нательных сенсорных сетях (Wireless Body Area Network, WBAN) помимо СУ, измеряющих текущее состояние человека, используются исполнительные устройства (рис. 6.5), способные воздействовать на объект измерений. В структуре исполнительного устройства присутствует активный элемент, воздействующий на внешнюю среду под управлением контроллера принятия решения, например на устройство ввода инсулина больному диабетом.



**Рис. 6.5. Структурная схема исполнительного устройства**

Медицинская нательная сенсорная сеть является примером *гетерогенной* сети – содержит два или больше типов сенсорных и исполнительных устройств, соответственно с различными энергетическими и функциональными возможностями.

На рис. 6.6 приведена структурная схема базовой станции сенсорных сетей.



**Рис. 6.6. Структурная схема БС**

В зависимости от программного обеспечения, реализуемого процессором, БС может выполнять функции:

- концентратора – агрегации данных, поступающих от СУ в выходной порт БС;
- маршрутизатора – вычисления адреса следующей БС, на которую будут отправлены данные с ее выходного порта;
- шлюза – сопряжения БСС с другой сетью, например Интернетом, при котором пакеты БСС преобразуются в формат глобальной сети.

БС функционально может сочетать функции концентратора и маршрутизации или концентратора и шлюза в одном узле.

### **6.3.2. Способы взаимодействия узлов в сенсорной сети**

Организация передачи данных от СУ зависит от масштаба беспроводной сенсорной сети.

В отличие от сетей WiFi или WiMAX время «жизни» сенсорной сети сильно зависят от расхода энергии сенсорными устройствами. По этой причине все способы передачи данных в беспроводных сенсорных сетях нацелены на снижение количества операций, выполняемых системой передачи. Расход энергии происходит во время передачи данных, их обработки, вычисления маршрута и т. д.

Все беспроводные сенсорные сети можно классифицировать на *одноранговые* и *иерархические*.

В одноранговых сенсорных сетях все СУ выполняют одинаковые задачи. Существует два способа выбора головных узлов: случайно или предопределенно. При случайном выборе создаются кластеры различных размеров.

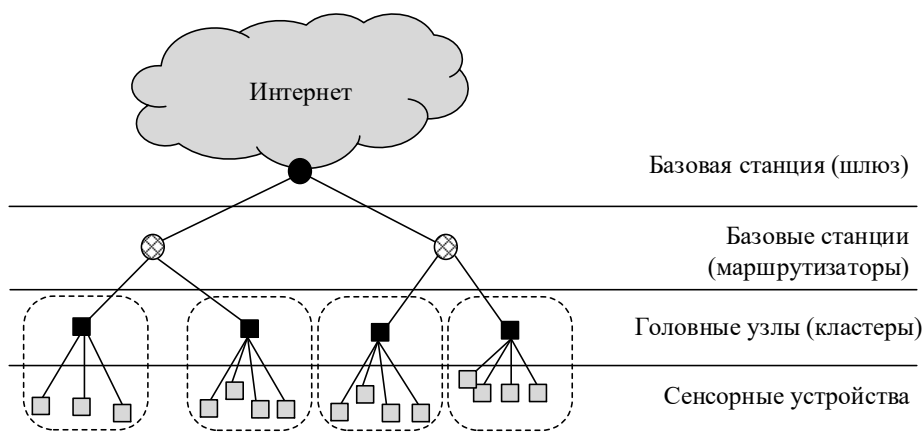
Масштабирование одноранговых сенсорных сетей образует топологию «дерево». Известны два алгоритма многошаговой маршрутизации в сенсорных сетях с древовидной топологией:

- алгоритм наводнения, при котором каждый узел сенсорной сети, получая пакет данных, передает его всем своим соседям. Этот процесс продолжается до тех пор, пока пакет не поступает на шлюз или не достигает максимально возможного количества переходов. Алгоритм реализуется просто, но имеет недостатки, связанные с дублированием сообщений, переданных в один и тот же сенсорный узел, и, как следствие, большое энергопотребление;

- алгоритм «распространения слухов» – улучшенный вариант алгоритма наводнения, при котором пакет передается единственному соседу, выбранному случайным образом из таблицы маршрутизации. Алгоритм преодолевает проблему дублирования сообщений, но увеличивает задержку их передачи.

Примером одноранговой беспроводной сенсорной сети может служить медицинская сеть удаленного мониторинга пациентов. Например, несколько пациентов носят СУ в форме браслета для записи электрокардиограммы, которая периодически передается лечащему врачу.

Все же чаще сенсорные сети покрывают большие географические районы. В целях экономии энергии мощность радиопередачи должна быть сведена к минимуму. Поэтому многошаговое взаимодействие является более распространенным вариантом беспроводной сенсорной сети и реализуется *иерархической* структурой (рис. 6.7).



**Рис. 6.7. Иерархическая структура беспроводной сенсорной сети**

СУ нижнего уровня сначала передают данные головным узлам, вокруг которых СУ объединены в кластеры. Головной узел представляет собой базовую станцию с функцией концентратора. Эти функции включают агрегацию данных, формирование пакетов сенсорной сети и их ретрансляцию до ближайшего маршрутизатора. Маршрутизаторы образуют ячеистую топологию mesh (сети), по которой транспортируются пакеты. За несколько **хопов** данные будут переданы на шлюз – выход сенсорной сети в глобальную сеть.

Таким образом, беспроводная сенсорная сеть может быть построена как совокупность кластеров, на которые разбивается создаваемое сенсорное поле.

Количество кластеров теоретически не ограничено. Это позволяет масштабировать размер сети под покрытие сенсорным полем значительных территорий.

Алгоритмы маршрутизации в иерархических беспроводных сенсорных сетях получили название – алгоритмы иерархической маршрутизации.

Примером иерархических беспроводных сенсорных сетей могут быть системы технологического учета электроэнергии, учета водоснабжения, охраны труда на вредном производстве и другие системы, контролирующие деятельность предприятий.

Из-за относительно большого количества СУ традиционная адресация на основе IP-протокола не применяется в сенсорных сетях в полном объеме. В сенсорных сетях получать данные иногда важнее, чем знать адреса СУ, с которых они отправлены. Поэтому в беспроводной сенсорной сети адреса могут иметь только базовые станции.

### 6.3.3. Механизмы кластеризации беспроводных сенсорных сетей

Выбор головных узлов сенсорной сети выполняется таким образом, чтобы обеспечить баланс расхода энергии. Практика реализации показала, что кластеризованные (иерархические) сенсорные сети «живут» дольше одноранговых.

Количество головных узлов зависит от размера беспроводной сенсорной сети и обычно не превышает 25% от общего числа СУ. Ближайшие к головному узлу сенсорные устройства образуют вокруг него кластер. Головной узел для своих СУ задает расписание передачи данных.

Известны несколько механизмов кластеризации, такие как **LEACH**, **PEGASIS**, **TEEN** и другие. Объединяет их общее правило – каждая базовая станция имеет возможность стать головным узлом.

Алгоритм LEACH (Low Energy Adaptive Cluster Hierarchy) предусматривает вероятностный выбор головного узла на основе энергетических характеристик в каждом новом цикле функционирования сенсорной сети.

По алгоритму PEGASIS (Power-Efficient Gathering Sensor Information Systems) сенсорные узлы в каждом новом цикле функционирования сети организуются в последовательные цепочки таким образом, чтобы только первые узлы цепочек передавали информацию на базовую станцию.

В соответствии с алгоритмом TEEN (Threshold-sensitive Energy Efficient Protocols) передача данных головному узлу от СУ происходит только, если количество накопленных данных достигло определенного уровня.

После назначения головных узлов остальные СУ начинают формироваться в кластеры на основе сигнала RSS (Received Signal Strength), получаемого от головного узла. Мощность RSS является параметром, позволяющим измерить расстояние от СУ до головного узла – число хопов.

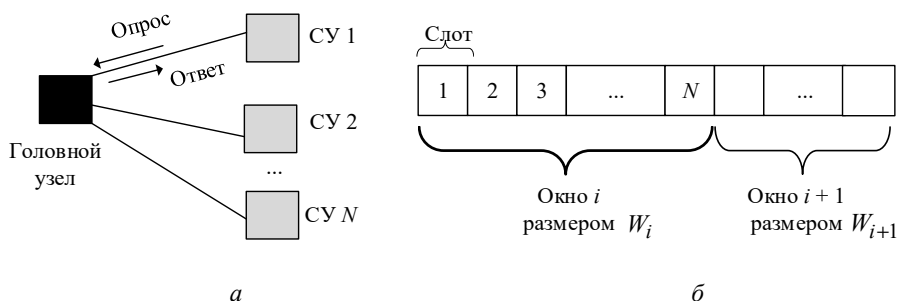
После формирования кластеров головной узел каждого из них широко-вещательной рассылкой передает на СУ сведения о себе – адрес, расстояние и т. п. СУ после передачи данных на головной узел могут перейти в спящий режим до следующего временного цикла.

### 6.3.4. Разрешение коллизий источников данных в кластере БСС

Когда множество СУ начинают передавать данные на головной узел, то могут возникнуть коллизии.

Коллизией, или конфликтом, называется наложение сигналов от разных СУ друг на друга. Существуют следующие методы разрешения коллизий в кластере БСС – опроса, прерываний и множественного доступа.

При применении метода опроса сенсорные устройства начинают передачу данных только по запросу головного узла. Если у СУ нет подготовленного пакета данных, то формируется пакет с идентификационным номером (ID) сенсорного устройства.



**Рис. 6.8. Метод опроса: а – обобщенная схема режима опроса; б – разделение времени опроса на окна и слоты**

Время отдельного опроса – временной интервал, который разделен на блоки – окна. Размер окон определяется количеством слотов, на которые они делятся. Размер слота – фиксированная величина для каждого сенсорного устройства. Так как слот – это тоже временной интервал, его размер определяется скоростью передачи данных от СУ до узла, т. е. его определяет оборудование, используемое в системе. Пример разделения времени опроса приведен на рис. 6.8.

В начале процесса узел посылает сигналы «опроса» всем СУ, которые находятся в зоне его покрытия. В этих сигналах содержится время начала доступа и продолжительность, то есть количество слотов. СУ, приняв эти сигналы, случайным образом выбирают слот, в котором будут передавать свои данные.

В процессе доступа в слоте возможно возникновение трех состояний. Пусто – в том случае, когда ни одно из СУ не выбрало текущий слот для передачи данных. Успех – когда только одно СУ передает данные в текущем слоте. Конфликт – когда более одного СУ начинают передавать данные в текущем слоте.

Опрос СУ, находящихся в зоне покрытия головного узла, заканчивается, когда в окне появляются только слоты с успешной передачей и пустые слоты.

Головной узел анализирует каждый слот и снимает информацию, переданную в них.

Продолжительность обслуживания сенсорного устройства при опросе включает время, затрачиваемое на передачу сигнала опроса и время передачи ответа.

Полный цикл взаимодействия головного узла с  $N$  сенсорными устройствами составляет случайное (из-за возможных коллизий – пока не ответят все) суммарное время  $T$ .

Связь этих переменных представлена на рис. 6.9.

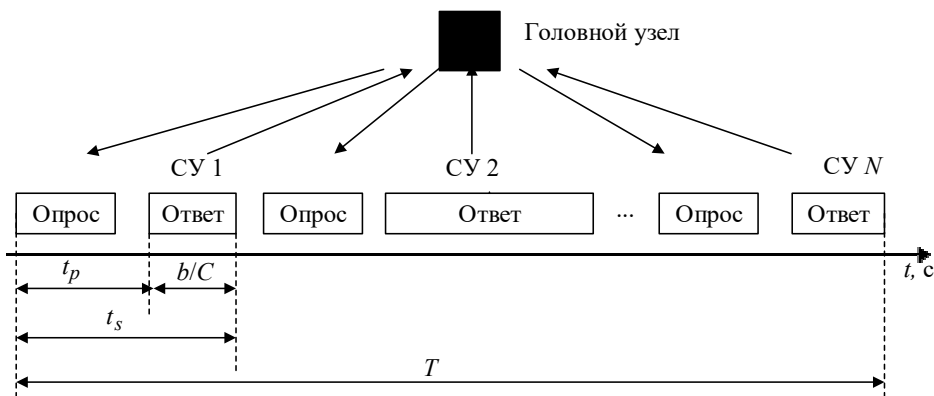


Рис. 6.9. Временная диаграмма реализации режима опроса

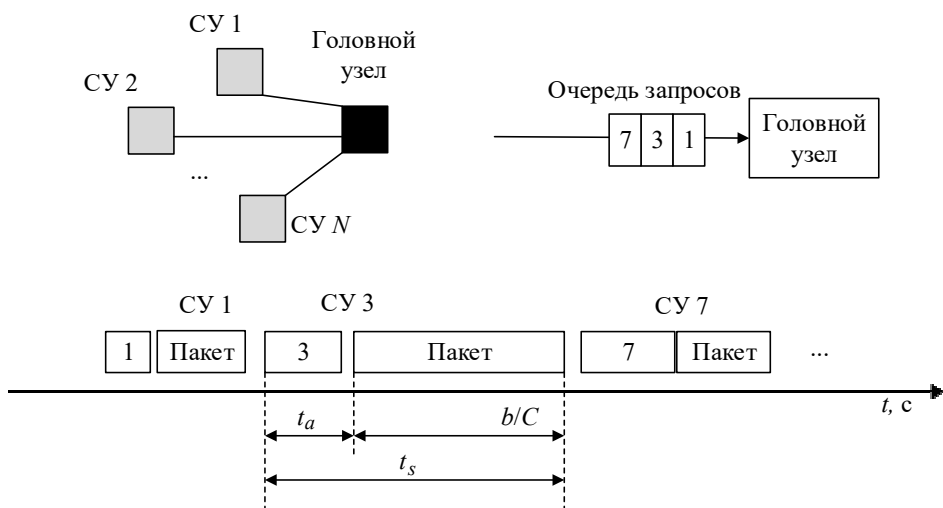
Отличие **метода прерываний** от опроса состоит в том, что вместо посылки сигналов опроса головной узел принимает и ставит в очередь сигналы от СУ о готовности начать передачу. Система, управляемая по прерываниям, предполагает соперничество за право передать данные на узел. СУ спонтанно посылают запросы на передачу данных головному узлу, который выстраивает их в очередь и направляет подтверждения. Если по истечении определенного интервала времени такое подтверждение не поступает, запрос автоматически повторяется. При свободном канале узел принимает данные от запрашивающего СУ. После завершения передачи узел переходит к приему данных от следующего в очереди запроса. На рис. 6.10 приведена схема и временная диаграмма реализации метода прерываний.

Время обслуживания сенсорного устройства в методе прерываний – случайная величина и включает время доступа, время ожидания в очереди и время передачи данных от СУ.

**Режим множественного доступа** предполагает доступ к головному узлу в соответствии с управляемым вероятностным арбитражем. При наличии данных на обработку СУ начинает передачу пакета на узел. Передача аварийно завершается и заново планируется сенсорным устройством при обнаружении пресечения с пакетами других СУ. В отсутствии пересечения пакет будет послан по назначению. Во избежание повторения конфликтов сенсорные устройства осуществляют повторную передачу в случайные интервалы времени.

Данный множественный доступ по исключению коллизий аналогичен рассмотренному в разделе 3.4.1.





**Рис. 6.10. Схема и временная диаграмма реализации метода прерываний**

## 6.4. Интернет вещей

В настоящее время уже существуют готовые интеллектуальные системы на базе беспроводных сенсорных сетей, получившие название «Интернет вещей» (Internet of Things, IoT).

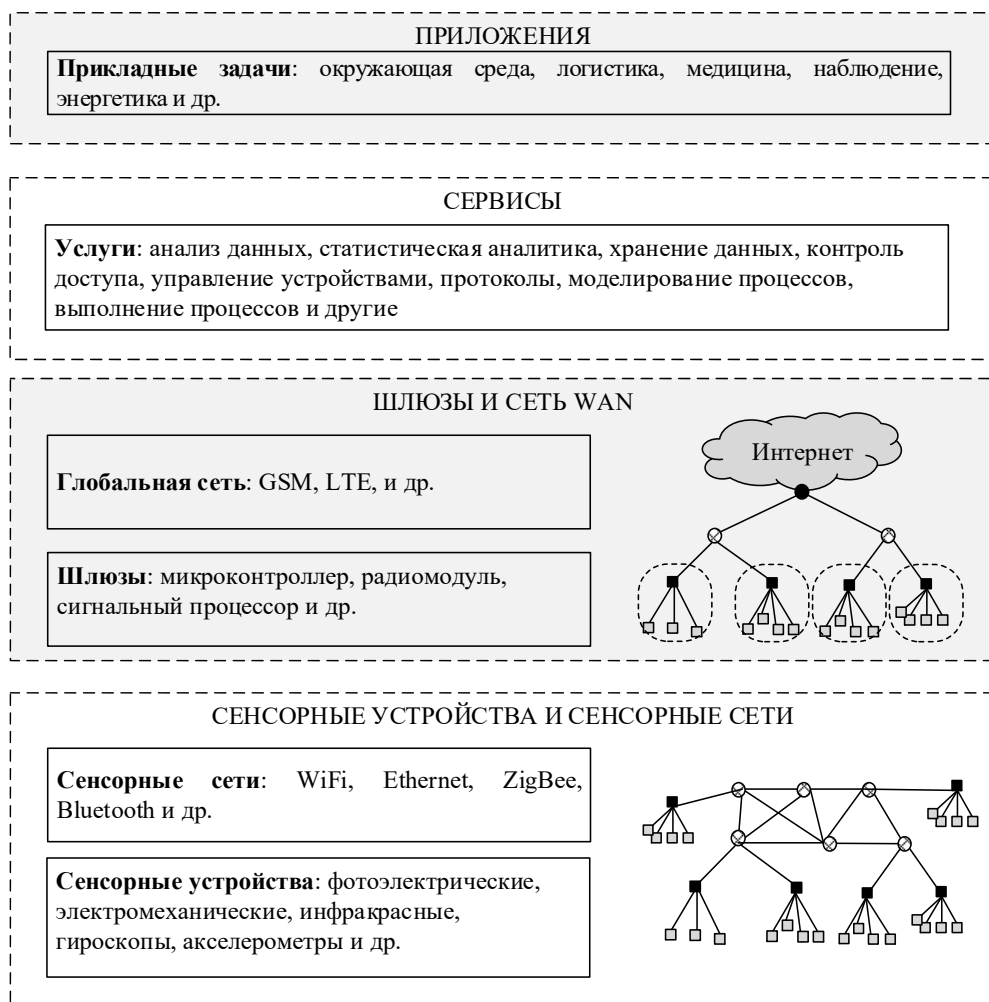
Сенсорное устройство, интегрированное в какой-либо объект, получило название – «вещь». Совместно вещи образуют некое множество объектов, способных взаимодействовать друг с другом и пользователями (владельцами вещей), создавая временные или постоянные сети.

IoT может работать как поверх сетей общего пользования, так и в изолированных инсталляциях. Управление вещами со стороны пользователей выполняется при помощи различных типов клиентских устройств и интерфейсов (графических, телефонных, SMS и др.).

### 6.4.1. Архитектура интернета вещей

Архитектура IoT включает четыре функциональных уровня (рис. 6.11).

*Уровень взаимодействия со средой* (сенсоры и сенсорные сети) – это самый нижний уровень архитектуры IoT, который состоит из «умных» (смарт) объектов (вещей), интегрированных с СУ. Сенсоры реализуют соединение физического и виртуального (цифрового) миров, обеспечивая сбор и обработку информации в реальном масштабе времени. Вещи соединяются с базовыми станциями (маршрутизаторами, шлюзами), образуя локальные вычислительные сети, такие как Ethernet, Wi-Fi или персональную сеть WPAN, стандарт которой разработан рабочей группой IEEE 802.15.



**Рис. 6.11. Архитектура интернета вещей**

*Сетевой уровень* обеспечивает транспорт данным, создаваемым вещами и их владельцами, на первом уровне IoT. Сетевая инфраструктура создается путем интеграции разнородных сетей в единую сетевую платформу.

*Сервисный уровень* содержит набор услуг, которые автоматизируют технологические операции в IoT: хранение данных, их анализ, обработку, обеспечение безопасного доступа к вещам, управление бизнес-процессами.

*Уровень приложений* включает различные готовые решения IoT в таких областях, как энергетика, транспорт, торговля, медицина, образование, и других прикладных областях. Приложения могут быть «вертикальными», когда они являются «специфическими» для конкретной области, а также «горизонтальными», которые могут использоваться в различных сферах деятельности.

## 6.4.2. Идентификация в интернете вещей

В отличие от БСС, которые большей частью сегодня имеют мониторинговые приложения, в интернете вещей каждый объект должен иметь свой идентификатор (ID). Идентификатор служит уникальным указателем на вещь, тогда как ее сетевой адрес может меняться в зависимости от физического местоположения.

Идентификатор представляет собой число или строку символов, однозначно именующие объект интернета вещей. Каждому идентификатору сопутствует набор метаданных – детальных сведений об объекте интернета вещей.

Для управления идентификацией вещей применяют набор различных технологий, которые адекватны уровням архитектуры интернета вещей:

- идентификаторы объектов;
- коммуникационные идентификаторы;
- идентификаторы приложений.

Технология идентификации объектов реализуется по принципу номерного знака и полностью заимствована из стандартов мобильного оборудования, где 15-разрядное число (идентификатор) хранится в SIM-карте мобильного устройства. Для идентификации вещи число хранится во встроенном в нее чипе. В интернете вещей этот способ используется при слиянии физического объекта с его цифровой копией. Яркий пример – RFID-системы<sup>6</sup>.

Коммуникационные идентификаторы необходимы при организации обмена данными между вещами по сети. Ими могут быть сетевые адреса и/или маршруты. Например, в системе «умный дом» каждая вещь, удаленно управляемая хозяином, может иметь свой IP-адрес. При идентификации по маршруту устройства запоминают, через какие узлы проходит к ним соединение. Маршрут остается статичным в памяти какое-то время и может служить идентификатором для обозначения вещи в сети.

Идентификаторы приложений – это URL-адреса<sup>7</sup>, которые используются для нахождения сервисов и приложений в рамках интернета вещей.

Для интернета вещей, построенного на БСС небольшого размера, поиск объектов по их идентификаторам является простой задачей.

Для масштабных сетей интернета вещей механизмы обнаружения объектов и сервисов автоматизированы подобно службе доменных имен DNS в компьютерных сетях.

Для интернета вещей существуют расширения системы DNS – ONS (Object Naming Service) для RFID-систем и ODS (Object Directory Service) для файловых систем.

---

<sup>6</sup> RFID (*Radio Frequency IDentification*, радиочастотная идентификация) – способ автоматической идентификации объектов посредством радиосигналов.

<sup>7</sup> URL (*Uniform Resource Locator*, единый указатель ресурса) – единообразный определитель местонахождения ресурса.

### 6.4.3. Способы взаимодействия в сети интернета вещей

На практике реализуют один из трех способов взаимодействия вещей и клиентов в сетях IoT [11]: прямой доступ, доступ через шлюз, доступ через сервер.

При прямом способе взаимодействия (рис. 6.12) обращение к вещи происходит по ее IP-адресу непосредственно из клиентского приложения. Интерфейс взаимодействия с вещью представляет собой web-интерфейс, управляемый посредством браузера. Недостаток способа – необходимость фиксированной адресации вещей, что вызывает зависимость от интернет-провайдера.

Альтернативой прямому доступу служит организация взаимодействия через шлюз (рис. 6.13). Внутри сенсорной сети используются собственные протоколы (не IP) взаимодействия СУ малой зоны действия. Шлюз выполняет функции ретрансляции данных из сенсорной сети в сеть Интернет. Взаимодействие посредством шлюза сегодня практически вытеснило метод прямого доступа.

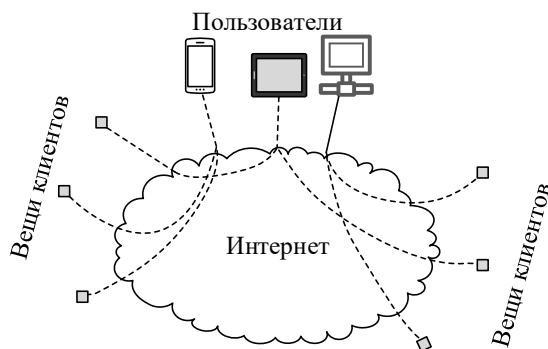


Рис. 6.12. Прямое взаимодействие клиента и вещи

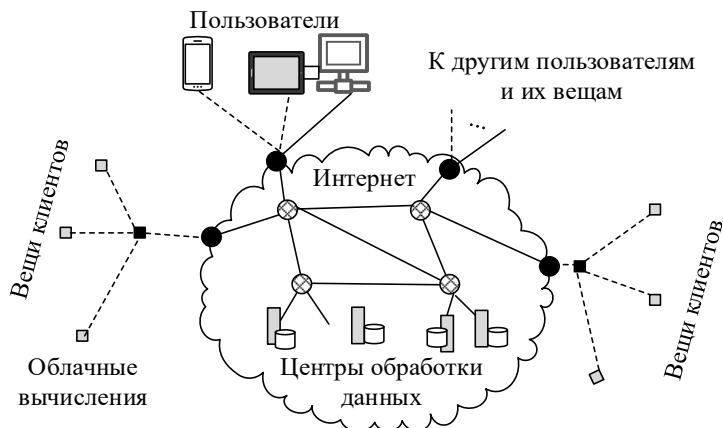
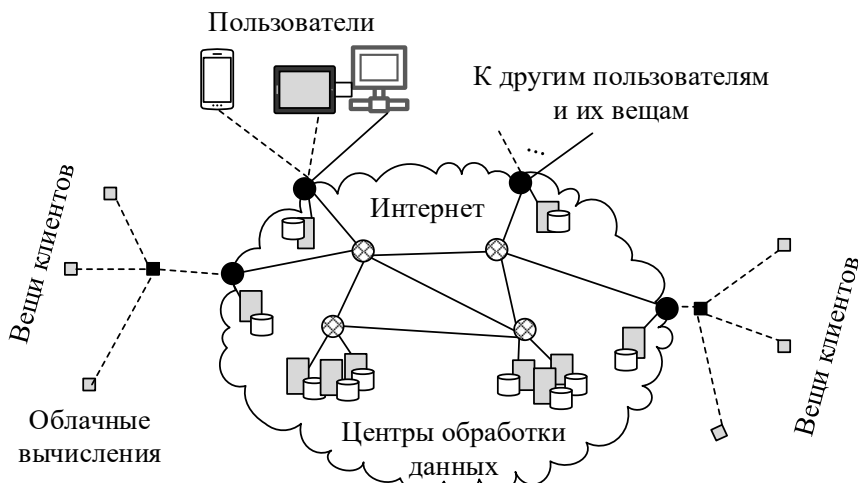


Рис. 6.13. Взаимодействие в IoT посредством шлюза

При доступе через сервер взаимодействие между вещами и пользователями аналогично технологии «клиент-сервер». Сервер включает модули обработки информации, базу данных для хранения принимаемой информации, интерфейс взаимодействия с интернет-вещами, систему контроля пользовательского доступа к вещам, управление их иерархией, параметрами и функционалом.

В практике проектирования IoT нашел применение комбинационный подход, в котором шлюзы соединяют локальные беспроводные сенсорные сети с сервером (рис. 6.14).



**Рис. 6.14. Взаимодействие в IoT посредством сервера**

Сервер в сложных системах IoT выполняет различные операции, связанные с анализом и обработкой поступающих на него сообщений. Сервер может устанавливать приоритеты сообщениям и формировать из них очереди. При недостаточном ресурсе канала связи или если получатель недоступен во время отправки сообщения, очередь хранит сообщение до тех пор, пока оно не будет доставлено.

#### **6.4.4. Облачные технологии в интернете вещей**

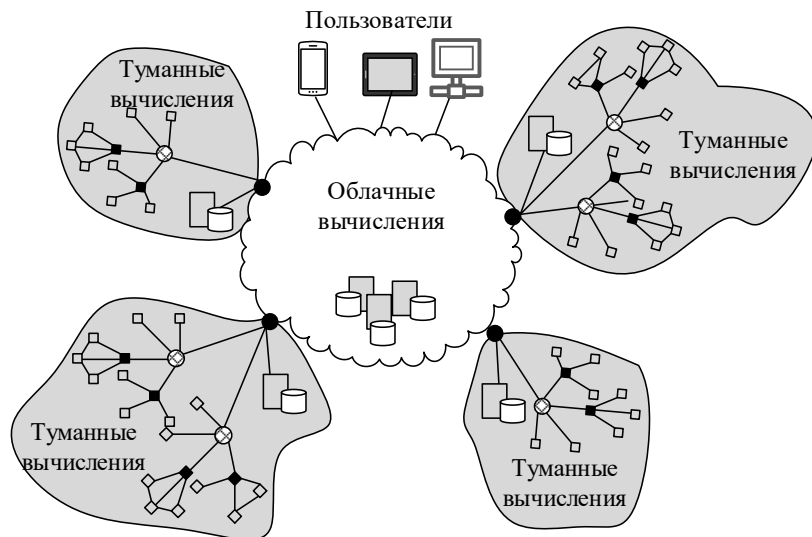
Облачные технологии в интернете вещей разделяются на облачные вычисления (Cloud Computing) и туманные вычисления (Fog Computing).

Необходимость такого разделения вызвана растущим количеством «умных» вещей, их пользователей и, соответственно, данных, которые нужно хранить и обрабатывать при работе интернета вещей.

Облачные вычисления – это модель обеспечения сетевого доступа к общему фонду конфигурируемых вычислительных ресурсов – серверам, системам хранения данных, приложениям и сервисам – как вместе, так и по отдельности. «Облако» строится на основе центров обработки данных (ЦОД).

Туманные вычисления – это разновидность облачных сервисов, расположенных не в «облаке», а в окружающей среде, например, на соседнем сервере. Туманные вычисления реализуются беспроводными сенсорными сетями, которые объединяют в IoT (рис. 6.15).

Туманные вычисления дополняют облачные за счет передачи части работы с «облака» «туману» при реализации задач, требующих значительных компьютерных ресурсов.



**Рис. 6.15. Туманные и облачные вычисления в концепции интернета вещей**

Таким образом, если говорить об интернете вещей не на уровне «умного дома», а, например, на уровне «умного города», то IoT реализуется в виде трехуровневой иерархической структуры. Верхний уровень занимают тысячи облачных ЦОД, предоставляющих ресурсы, необходимые для выполнения серьезных, например аналитических, программных приложений IoT. Уровнем ниже располагаются десятки тысяч распределенных управляющих ЦОД, в которых содержится «интеллект» Fog Computing («Обработка тумана»), а на нижнем уровне находятся миллионы вычислительных устройств умных вещей.

#### **6.4.5. Протоколы интернета вещей**

Для взаимодействия вещей и пользователей в IoT необходимы специальные протоколы. Рассмотрим их в соответствии с последовательными участками установления связи между элементами IoT:

- СУ (вещи) и пользовательское устройство (компьютер, планшет, мобильный телефон и т.д.) устанавливают друг с другом связь, назовем этот участок взаимодействия «Device-to-Device» (D2D);

- собранные данные передаются в серверную инфраструктуру (облако), назовем этот участок взаимодействия «Device-to-Server» (D2S);

- серверная инфраструктура должна совместно использовать данные, имея возможность передавать их обратно устройствам, программам анализа или пользователям. Назовем этот участок взаимодействия «Server-to-Server».

Известны следующие протоколы реализации взаимодействия элементов интернета вещей согласно выделенным участкам:

- DDS: быстрая шина для интегрирования интеллектуальных устройств (D2D);

- CoAP: протокол для передачи информации о состоянии узла на сервер (D2S);

- MQTT: протокол для сбора данных устройств и передачи их серверам (D2S);

- XMPP: протокол для соединения устройств с пользователями, частный случай D2S-схемы, когда пользователи соединяются с серверами;

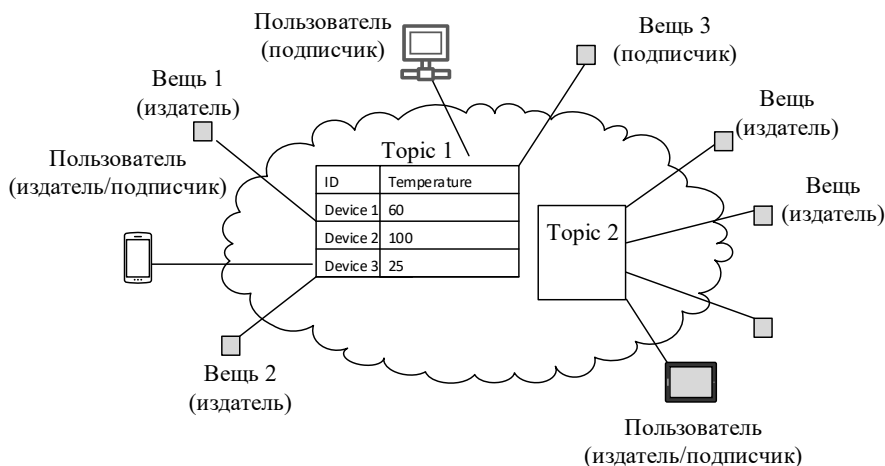
- STOMP: протокол для обмена сообщениями между устройством и сервером, реализованными на разных языках и платформах (D2S);

- AMQP: система организация очередей для соединения серверов между собой (S2S).

Каждый из перечисленных является протоколом реального времени и может варьироваться в зависимости от приложения интернета вещей. Известны десятки вариантов реализации перечисленных протоколов на практике. Объединяет их общая идея организации взаимодействия – схема «издай/подпишись» (publish/subscribe), которая позволяет соединять тысячи устройств.

DDS (Data Distribution Service) – реализует шаблон публикации-подписки для отправки и приема данных, смен состояний и команд среди конечных узлов. Узлы-издатели создают информацию – «topic» (темы, разделы: температура, местоположение, давление и т.д.) и публикуют шаблоны в виде реляционной модели данных (рис. 6.16). Узлам, предназначенным для таких разделов, DDS прозрачно доставляет созданные шаблоны и реализует прямую шинную связь между устройствами. В качестве транспорта используется протокол UDP (User Datagram Protocol), предназначенный для передачи дейтаграмм.

С помощью DDS реализуется многоадресная система между вещами и пользователями. Передача сообщений между взаимодействующими вещами и пользователями производится по методу «запрос-ответ». В отличие от других протоколов, в которых в явном виде надо указывать, что, кому и когда передавать, протокол DSS является анонимной моделью взаимодействия – вещи не знают своих пользователей, а пользователи не знают, какие конкретно сенсорные устройства являются источниками информации. Такая анонимность является основой масштабируемости и самоорганизации БСС, на базе которой построен интернет вещей, – при замене вещи или пользователя не надо переписывать связующее программное обеспечение.



**Рис. 6.16. Схема работы DDS-протокола интернета вещей**

CoAP (Constrained Application Protocol) работает на прикладном уровне и предназначен для передачи данных по линиям с ограниченной пропускной способностью. CoAP был разработан на основе протокола HTTP с учетом низкой мощности и малого потребления энергии СУ интернета вещей. В отличие от протокола HTTP, который является текстовым, CoAP – бинарный протокол, что уменьшает размер его служебных данных. Транспортируется посредством UDP.

CoAP организован в два слоя: транзакций и «Request/Response» («Запрос/Ответ»).

Слой транзакций обрабатывает обмен сообщениями между конечными точками. Сообщения обмена на этом слое могут быть четырех типов:

- «Confirmable» – требует подтверждения;
- «Non-confirmable» – не требует подтверждения;
- квитирование – подтверждает получение «Confirmable» сообщения;
- «Reset» – указывает на то, что «Confirmable» сообщение было получено, но контекст, подлежащий обработке, отсутствует.

Слой «Запрос/Ответ» представляет модель взаимодействия Клиент/Сервер для манипулирования ресурсами и передачи. «Вещь» обычно «играет роль» сервера. По запросу клиента (приложения пользователя) устанавливается флаг наблюдения, и сервер начинает отвечать, передавая измерения состояний сенсорных устройств (вещей).

Протокол CoAP может использоваться с любым протоколом прикладного уровня: SMTP, FTP, HTTP, HTTPS.

MQTT (Message Queue Telemetry Transport) осуществляет сбор данных от множества узлов и передачу их на сервер. В качестве транспорта – протокол TCP. MQTT предназначен для телеметрии и дистанционного мониторинга.

Протокол MQTT основывается на модели издатель-подписчик с использованием промежуточного сервера – брокера. Брокер решает задачу формирования очередей сообщений и их приоритезации. Таким образом, вся передаваем



мая информация разделяется по направлениям на разные каналы, число которых соответствует количеству издателей/подписчиков.

Все сенсорные или исполнительные устройства посылают данные только брокеру и принимают данные тоже только от него. То есть когда один клиент, так называемый издатель, передает сообщение  $M$  на определенную тему  $T$ , то все клиенты, которые подписываются на тему  $T$ , получают это сообщение  $M$ . Например, три клиента подключены к брокеру, клиенты  $B$  и  $C$  подписываются на topic «Temperature» (рис. 6.17). В какое-то время, когда клиент  $A$  передает значение «30» на topic «Temperature» (рис. 6.17, а), сразу после его получения брокер передает это сообщение к подписавшимся клиентам (рис. 6.17, б).

Также в MQTT предусматривается три выбора надежности обмена сообщениями, которые обеспечиваются тремя уровнями качества обслуживания (QoS, Quality of Service):

- QoS0 – сообщение передается только один раз и не требует подтверждения;
- QoS1 – сообщение отправляется до тех пор, пока не будет получено подтверждение об его успешной доставке;
- QoS2 – при однократной передаче сообщения используется четырехступенчатая процедура подтверждения доставки.

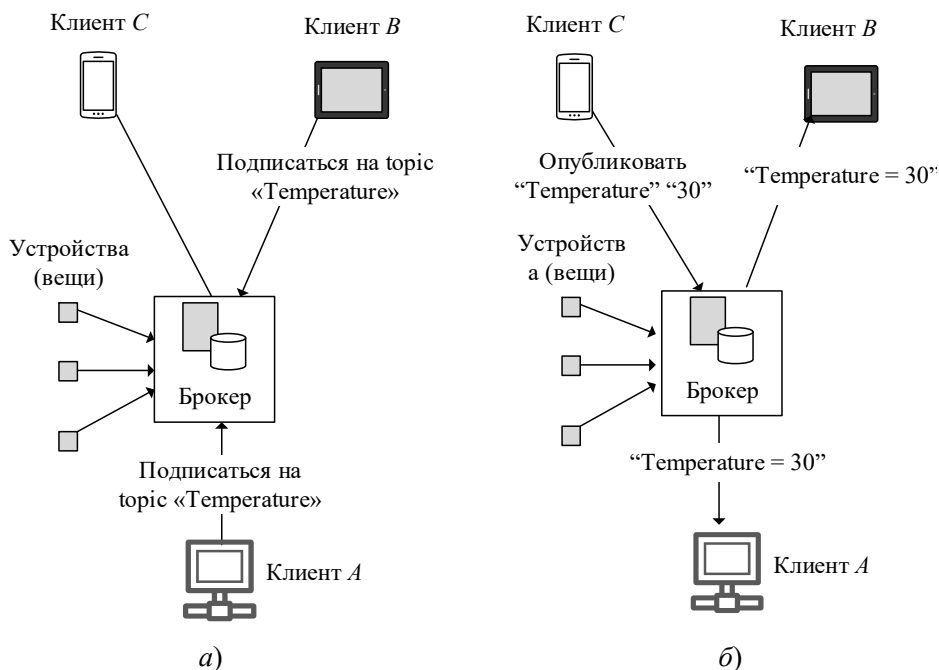


Рис. 6.17. Схема работы протокола MQTT

XMPP (Extensible Messaging and Presence Protocol) – расширяемый протокол обмена сообщениями и идентификационной информацией. XMPP давно

используется в сети Интернет для передачи сообщений в режиме реального времени. В XMPP применяется текстовый формат языка разметки ХМ, благодаря чему протокол подходит для использования в сетях IoT. XMPP работает поверх архитектур издатель-подписчик и клиент-сервер и используется для адресации устройств в небольших сетях (адресация вида name@domain.com). В качестве транспорта применяется протокол TCP.

С помощью XMPP, например, возможно подключение домашнего термостата к Web-серверу с целью получения к нему доступа с телефона. Сильными сторонами этого протокола являются безопасность и масштабируемость, что делает его идеальным для приложений Интернета вещей с ориентацией на потребителя.

Таким образом, на участке сети между СУ и брокером чаще всего применяются протоколы – CoAP, MQTT и XMPP. Выбор конкретного протокола зависит от условий реализуемости сети. Можно отметить, что XMPP нашел свое применение в системах климат-контроля и освещения, а также используется для адресации устройств в небольших персональных сетях. MQTT поддерживает качество обслуживания и проверку доставки сообщений. Протокол MQTT обеспечивает такие приложения, как мониторинг утечек и контроль за окружающей средой на территориях опасного производства. Другими приложениями для MQTT могут быть контроль потребления энергии, управление светом и даже интеллектуальное садоводство. CoAP предназначен для устройств с ограниченными ресурсами и для сетей с низким энергопотреблением. Известно применение протокола в системах датчиков умного дома.

Для сетей, использующих оборудование различных платформ, можно рекомендовать простой протокол передачи сообщений STOMP.

STOMP (Simple Text Oriented Message Protocol) – простой протокол обмена сообщениями, предполагающий широкое взаимодействие со многими языками, платформами и брокерами. Так, STOMP согласует взаимодействие сервера, описываемого на одном языке программирования, и клиентом, программное обеспечение которого разработано на другом языке. Поддерживает большое количество совместимых клиентских библиотек, связанных языков.

Если протокол MQTT обеспечивает «сквозную» связь, как от брокера к сенсорным узлам, так и от брокера к серверу, то протокол STOMP ориентирован только на взаимодействие брокера с сервером.

Для соединения серверов между собой (S2S) разработан протокол AMQP.

Протокол AMQP (Advanced Message Queuing Protocol) – усовершенствованный протокол организации очереди сообщений. Как следует из названия, протокол обслуживает исключительно очереди – пересылает транзакционные сообщения между серверами. Работает поверх TCP.

AMQP основан на трех понятиях:

- Сообщение – единица передаваемых данных.
- Брокер – приемник всех сообщений. Брокер распределяет сообщения в одну или несколько очередей. При этом в брокере сообщения не хранятся.

Механизмы работы брокера могут быть разными (зависит от версии протокола):

- fanout – сообщение передается во все прицепленные к брокеру очереди;
- direct – сообщение передается в очередь с именем, совпадающим с ID маршрута;

- topic – сообщение передается в очередь по теме подписки.

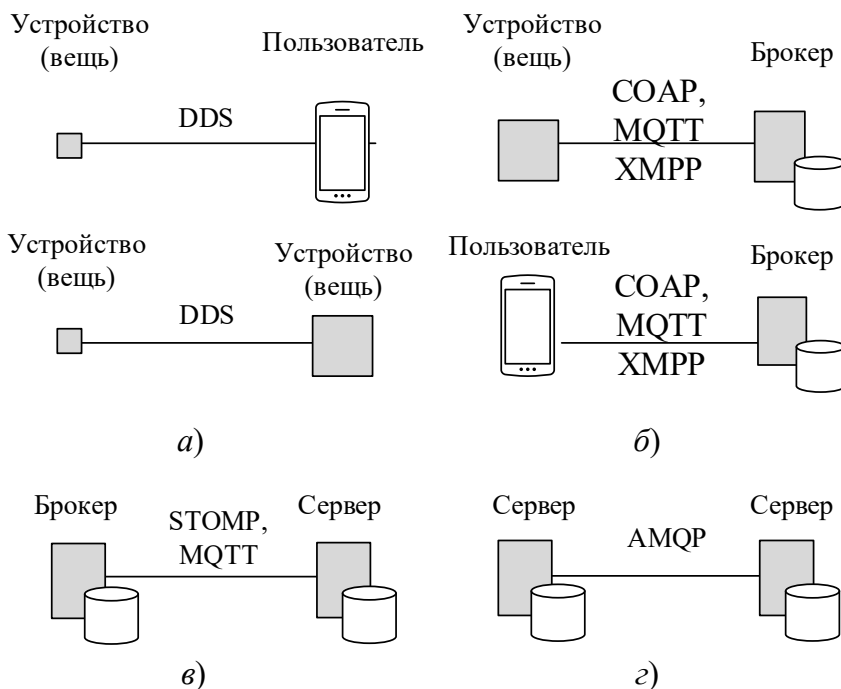
В очереди сообщения хранятся до тех пор, пока они не будут забраны пользователем. Клиент всегда забирает сообщения из одной или нескольких очередей.

Для эффективного использования протокола в интернете вещей необходимо правильно выбрать версию AMQP-брокер.

AMQP применяется главным образом в обмене деловыми сообщениями. В интернете вещей AMQP наилучшим образом подходит для реализации аналитических функций на базе серверов.

Итак, краткий обзор протоколов интернета вещей позволяет классифицировать их по назначению (табл. 6.1, рис. 6.18).

Беспроводные сенсорные сети считаются одной из самых перспективных технологий XXI века. Недорогие и «умные» сенсоры, объединенные в беспроводную сеть, подключенную к интернету (интернет вещей), предоставляют широкий набор услуг контроля и управления телами, домами, предприятиями, автомобилями и т. д.



**Рис. 6.18. Протоколы взаимодействия вещей и пользователей в IoT**

Таблица 6.1

## Сравнение протоколов интернета вещей

Протокол	Транспорт	Назначение	Особенность
DDS (рис. 6.18, а)	UDP	Для сетей, нуждающихся в распределении нагрузки	Реализует прямую шинную связь между устройствами на базе реляционной модели данных
CoAP (рис. 6.18, б)	UDP	Для сетей с ограниченным ресурсом по энергопотреблению	Учитывает различные вопросы среды реализации в ограниченных сетях
MQTT (рис. 6.18, в, 6.18, г)	TCP	Для загруженных сетей с большим количеством устройств и брокером	Использование механизма очередей сообщений
XMPP (рис. 6.18, б)	TCP	Для адресации в небольшой персональной сети	Для идентификации используются ID, по формату похожие на адреса электронной почты
STOMP (рис. 6.18, в)	TCP	Для сети, в которой несколько разных протоколов, нуждающейся в простом протоколе передачи сообщений через брокера	Взаимодействие со многими языками, платформами и брокерами
AMQP (рис. 6.18, з)	TCP	Для реализации аналитических функций на базе серверов	Обслуживает очереди при передаче транзакционных сообщений между серверами

В настоящее время интернет вещей представляет собой множество разрозненных сетей, однако уже в ближайшем будущем эксперты прогнозируют переход к Всеобъемлющему Интернету (Internet of Everything, IoE). Internet of Everything объединит различные процессы, объекты, большие данные и людей для интеллектуального взаимодействия и принятия обоснованных решений по регулировке системы.

## Контрольные вопросы

1. В чем особенность стандарта IEEE 802.11 с базовым набором услуг?
2. За счет чего облегчается объединение беспроводных локальных сетей с проводными?
3. Почему в сетях 802.11 станция не может обнаружить коллизию во время передачи?
4. Какова основная задача базовой станции стандарта IEEE 802.16?
5. В чем преимущества WiMAX перед проводными сетями?
6. Каково назначение базовых станций в сенсорной сети?

7. Назовите элементы структурной схемы сенсорного устройства и назначение каждого из них.
8. Сопоставьте одноранговые и иерархические беспроводные сенсорные сети. В чем преимущества первых перед вторыми и наоборот?
9. С какой целью кластеризуются беспроводные сенсорные сети?
10. Охарактеризуйте качественно способы разрешения коллизий в кластере базовой сенсорной сети.
11. Что подразумевают под «вещью» в интернете вещей?
12. Каково назначение выделенных слоев в архитектуре интернета вещей?
13. Чем отличаются облачные технологии в интернете вещей?
14. Охарактеризуйте кратко способы взаимодействия в интернете вещей.

## 7. СЕТЕВЫЕ СЛУЖБЫ

### 7.1. Качество обслуживания (службы QoS)

Основное требование к компьютерной сети – это выполнение сетью того набора услуг, для которого она предназначена. К таким услугам могут относиться:

- доступ к файловым архивам;
- доступ к страницам веб-сайтов;
- обмен с использованием электронной почты;
- интерактивный обмен с помощью IP-телефонии;
- потоковое видео и т. д.

В общем случае качество обслуживания (Quality of Service – QoS) определяет вероятностные оценки выполнения тех или иных требований, предъявляемых к сети пользователями или приложениями.

Например, при передаче голосового трафика пакеты должны доставляться с задержкой не более  $N$  мс. Вариация задержки не должна быть более  $M$  мс и  $M \ll N$ . Это требование должно выполняться сетью с вероятностью 0,95.

Поддержка QoS требует взаимодействия всех сетевых элементов (концентраторов, маршрутизаторов, коммутаторов) на пути следования трафика, т. е. «из конца в конец». Гарантия обеспечения QoS зависит от самого «слабого» элемента между отправителем и получателем.

Традиционно сети поддерживают три типа QoS.

**Сервис Best effort** – сервис с максимальными усилиями. Это означает, что сеть не дает никаких гарантий на обслуживание (Ethernet, Token Ring, IP, X25).

**Сервис с предпочтением** (называют «мягким» сервисом QoS). Здесь некоторые виды трафика обслуживаются лучше остальных. Но характеристики обслуживания точно неизвестны – они зависят от характеристик трафика.

Например, при высокой интенсивности высокоприоритетного трафика может совсем прекратиться обслуживание трафика с низким приоритетом.

**Гарантированный сервис** (называют «жестким» или «истинным» сервисом QoS). Различным типам трафика даются статистические гарантии. Обычно этот тип QoS основан на предварительном резервировании сетевых ресурсов для каждого из потоков, получивших гарантии обслуживания.

Однако эти гарантии носят статистический характер и требуют контроля интенсивности входных потоков с тем, чтобы это значение не превышало заранее оговоренную величину.

Такой тип QoS применяется обычно для обслуживания приложений, для которых важны гарантии пропускной способности и/или задержек. Например, это может быть трафик видеоконференции или трафик, поступающий от измерительных систем реального времени.

Между поставщиком и потребителем сервиса заключается договор, который называется соглашением об уровне обслуживания – *Service Level Agreement, SLA*.

В соглашении определяются:

- параметры качества трафика (средняя пропускная способность, максимальные задержки и их вариации, максимальная интенсивность потерь информации, коэффициент готовности сервиса, максимальное время восстановления сервиса после отказа и т. п.);
- методы измерения качества обслуживания;
- система оплаты за обслуживание;
- санкции за нарушение обязательств поставщиком услуг;
- меры, предпринимаемые, если трафик пользователя превышает согласованное значение (отбрасывание пакетов или пометка о возможности их последующего отбрасывания).

Таким образом, уровни обслуживания ориентированы и зависят от требований, которые предъявляют к сети пользователи.

### 7.1.1. Требования разных типов приложений

Требования разных типов приложений классифицируются по трем параметрам:

- предсказуемость скорости передачи данных;
- чувствительность трафика к задержкам пакетов;
- чувствительность трафика к потерям и искажениям пакетов.

**Предсказуемость скорости передачи данных.** Приложения делятся на 2 класса:

- потоковый трафик (рис. 7.1);
- пульсирующий поток (рис. 7.2).

Приложения с потоковым трафиком порождают поток с постоянной битовой скоростью – Constant Bit Rate, CBR

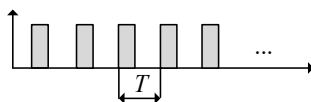


Рис. 7.1. Потоковый график

Средняя скорость  $CBR = B/T$  (бит/с), где  $B$  – размер пакета,  $T$  – период потока.

Приложения же с пульсирующим трафиком отличаются высокой степенью непредсказуемости. Они характеризуются переменной битовой скоростью – Variable Bit Rate, VBR.

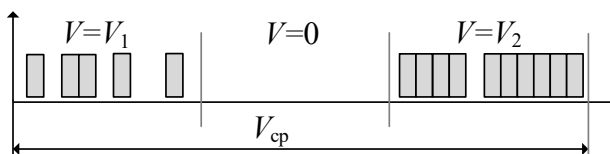


Рис. 7.2. Пульсирующий поток

Такой поток характеризуют коэффициентом пульсации в виде отношения средней скорости на определенном коротком интервале к средней скорости на всем интервале сеанса связи.

Например (см. рис. 7.2):  $K_{п1} = V_1/V_{ср}$ ,  $K_{п2} = V_2/V_{ср}$ .

Обычно значения коэффициента пульсации находятся в интервале от 2:1 до 100:1

**Чувствительность трафика к задержкам пакетов.** В порядке повышения чувствительности к задержкам используется следующая классификация:

- *Асинхронные приложения.* Практически нет ограничений I (пример — электронная почта).
- *Синхронные приложения.* Чувствительны к задержкам, но допускают их в некоторых пределах.
- *Интерактивные приложения.* Здесь задержки замечаются пользователями, но не сказываются негативно на функциональности приложения. Пример — текстовый редактор, работающий с удаленным файлом.
- *Изохронные приложения.* В них присутствует порог чувствительности к задержкам, после которого резко снижается функциональность. Пример: передача речи. При задержках более 100–150 мс резко снижается качество передаваемого голоса.
- *Сверхчувствительные к задержкам приложения.* Задержка может привести к потере функциональности. Пример: управление объектами в реальном времени (задержка может привести к аварии).

**Чувствительность трафика к потерям и искажениям пакетов.** В этой категории трафика различают 2 группы:

- *чувствительные к потерям приложения.* Сюда относятся файловый сервис, сервис баз данных, электронная почта и т. д. — все приложения, передающие алфавитно-цифровые данные;
- *устойчивые к потерям приложения.* Это приложения, порождающие трафик, несущий информацию об инерционных физических процессах. Сюда относится большая часть мультимедийных приложений (передача аудио- или видеотрафика).

Однако устойчивость такого трафика имеет предел. Например, процент потерянных пакетов не должен превышать 1%.

Не любой мультимедийный трафик устойчив к потерям. Например, компрессированные голос и видео очень чувствительны к потерям и поэтому относятся к первому типу приложений.

Возможны сочетания чувствительностей по отношению к некоторым конкретным видам трафика. Например, сочетанию равномерного потока, изохронного приложения, устойчивости к потерям соответствуют:

- IP-телефония;
- поддержка видеоконференций;
- аудиовещание через Интернет.

Таких устойчивых сочетаний характеристик относительно немного.



Изложенная классификация приложений лежит в основе типовых требований к параметрам и механизмам качества обслуживания в современных сетях.

Вероятностно-временные характеристики качества обслуживания подобных требований зависят чаще всего от: скорости передачи данных; задержки передачи пакетов; уровня потерь и искажений пакетов.

Механизмы QoS могут только управлять распределением имеющейся пропускной способности сети в соответствии с требованиями приложений. А распределение имеющейся пропускной способности сети сводится к управлению трафиком сети.

### **7.1.2. Управление трафиком. Службы QoS**

Итак, службы QoS должны обеспечивать вероятность того, что сеть соответствует заданному соглашению о трафике.

Приложения запускаются и работают на хостах и обмениваются данными между собой.

Приложения отправляют данные операционной системе для передачи по сети.

Как только данные переданы операционной системе, они становятся сетевым трафиком.

Сетевая служба QoS опирается на способность сети обработать трафик так, чтобы гарантированно выполнить запросы некоторых приложений.

Гарантированное выполнение запросов приложений по QoS требует наличия в сети специального механизма по обработке сетевого трафика.

Такой механизм должен быть способен идентифицировать трафик, имеющий право на особую обработку и право управлять этими механизмами.

Функциональные возможности QoS призваны удовлетворить требования сетевых приложений.

**Основные параметры QoS.** Требования различных приложений по обработке их сетевого трафика нашли выражение в следующих параметрах, связанных с QoS:

- Bandwidth (полоса пропускания) – скорость, с которой трафик, генерируемый приложением, должен быть передан по сети;
- Latency (задержка) – задержка, которую приложение может допустить в доставке пакета данных.
- Jitter – изменение времени задержки.
- Loss (потеря) – процент потерянных данных.

Сетевые ресурсы не безграничны. Механизм QoS контролирует распределение сетевых ресурсов, чтобы выполнить требования по передаче трафика приложения.

**Фундаментальные ресурсы QoS и механизмы обработки трафика.** Сети, которые связывают хосты, используют разнообразные устройства (сетевые адаптеры хостов, маршрутизаторы, коммутаторы, концентраторы).

Каждое устройство имеет сетевые интерфейсы. Каждый сетевой интерфейс может принять и передать трафик с конечной скоростью.

Если скорость поступления трафика на интерфейс выше, чем скорость, с которой интерфейс передает трафик дальше, то возникает перегрузка.

Сетевые устройства могут обработать состояние перегрузки, организовав очередь трафика в памяти устройства (в буфере), пока перегрузка не пройдет.

В других случаях сетевое оборудование может отказаться от трафика, чтобы облегчить перегрузку.

Возникает либо задержка, либо потеря трафика.

Пропускная способность сетевых интерфейсов и наличие буферной памяти составляют фундаментальные ресурсы, обеспечивающие QoS.

**Распределение ресурсов QoS по сетевым устройствам.** Трафик приложений, более терпимых к задержкам, становится в очередь. Трафик приложений, критичных к задержкам, передается далее.

Для выполнения такой задачи сетевое устройство должно идентифицировать трафик, а также иметь *очереди и механизмы их обслуживания*.

**Механизм обработки трафика** включает в себя услуги, обеспечивающие названные выше типы сервиса: с максимальными усилиями, с предпочтением и гарантированный.

Различным типам трафика даются статистические гарантии.

Гарантированный сервис основан на предварительном резервировании сетевых ресурсов для каждого из потоков, получивших гарантии обслуживания.

Операционная система Windows содержит компоненты QoS, обеспечивающие:

- резервирование ресурсов и службы RSVP ((ReSerVation Protocol);
- управление трафиком;

• классификатор пакетов, который относит пакет к определенному классу сервиса. При этом пакет будет поставлен в соответствующую очередь. Очереди управляются *планировщиком пакетов* QoS. Планировщик пакетов QoS определяет параметры QoS для специфического потока данных. Трафик помечается определенным значением приоритета. Планировщик пакетов QoS определяет график постановки в очередь каждого пакета и обрабатывает конкурирующие запросы между поставленными в очередь пакетами, которые нуждаются в одновременном доступе к сети.

**Модель службы QoS.** Эта служба имеет распределенный характер (ее элементы должны быть на всех сетевых устройствах, продвигающих пакеты)

Должны быть и элементы централизованного управления, позволяющие администратору конфигурировать механизмы QoS в устройствах сети.

Архитектура службы QoS включает элементы трех основных типов:

- *средства QoS узла*, выполняющие обработку трафика в соответствии с требованиями на качество обслуживания;
- *протоколы QoS-сигнализации* для координации работы сетевых элементов по поддержке качества обслуживания «из конца в конец» (end-to-end, e2e);

- *централизованные функции политики* управления и учета QoS, позволяющие администраторам сети целенаправленно воздействовать на сетевые элементы для разделения ресурсов сети между различными видами трафика с требуемым уровнем QoS.

Средства QoS узла могут включать механизмы двух типов:

- механизмы обслуживания очередей;
- механизмы кондиционирования трафика.

*Механизм обслуживания очередей* – элемент любого устройства, работающего по принципу коммутации пакетов. Эти механизмы работают в любом сетевом устройстве (за исключением повторителей).

**Алгоритмы управления очередями.** Чаще всего применяют следующие алгоритмы управления очередями:

- FIFO (достаточен только для Best effort (с максимальными усилиями);
- приоритетное обслуживание, которое называют также «подавляющим»;
- взвешенное обслуживание.

### 1) Традиционный алгоритм FIFO

Достоинство – простота реализации и отсутствие необходимости конфигурирования. Недостаток – невозможность дифференцированной обработки пакетов различных потоков.

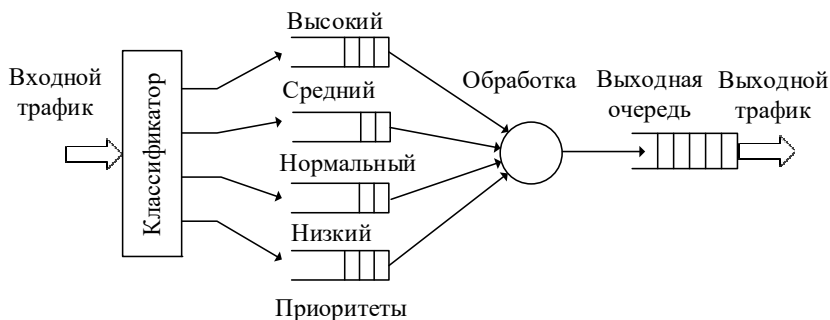
### 2) Приоритетное обслуживание (Priority Queuing)

Сначала необходимо решить отдельную задачу – разбить общий входной поток устройства на классы (приоритеты). Применяют адрес пункта назначения, приоритет, идентификатор приложения и т. д. Признак помещают в поле приоритета. Если же такое поле в пакете не предусмотрено, приходится разрабатывать дополнение к протоколу.

Например, для сети Ethernet был разработан протокол IEEE 802.1 Q/p, в котором вводится специальное трехбитовое поле для приоритета.

Затем пакет помещается в очередь, соответствующую заданному приоритетному классу.

Пример с четырьмя приоритетными очередями: высокий, средний, нормальный и низкий приоритет – приведен на рис. 7.3.



**Рис. 7.3. Приоритетное обслуживание**

Здесь очереди имеют абсолютный приоритет – пока не обработаны пакеты, из очереди более высокого приоритета не производится переход к более

низкоприоритетной очереди. Недостаток: если высока интенсивность высокоприоритетного трафика, обслуживание низкоприоритетного трафика может совсем не производиться.

### 3) Взвешенные настраиваемые очереди (Weighted Queuing)

Вес класса – процент предоставляемой классу пропускной способности (от полной выходной пропускной способности).

Алгоритм для назначения весов называется *настраиваемой* очередью (рис. 7.4). Гарантируются некоторые требования к задержкам.

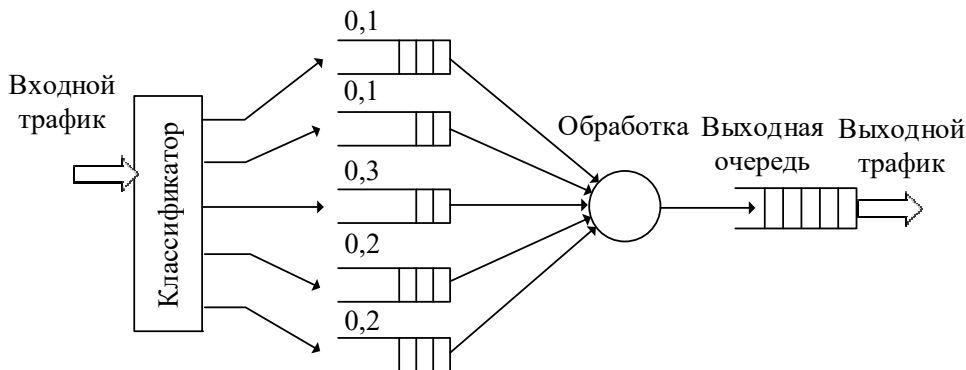


Рис. 7.4. Взвешенные настраиваемые очереди

Очереди обслуживаются циклически, и в каждом цикле обслуживания из каждой очереди выбирается такое число байтов, которое соответствует весу этой очереди. Например: цикл = 1 сек., скорость выходного интерфейса = 100 Мбит/с. В каждом цикле из очередей выбираются следующие объемы данных: 1 – 10 Мбит; 2 – 10 Мбит; 3 – 30 Мбит; 4 – 20 Мбит; 5 – 30 Мбит.

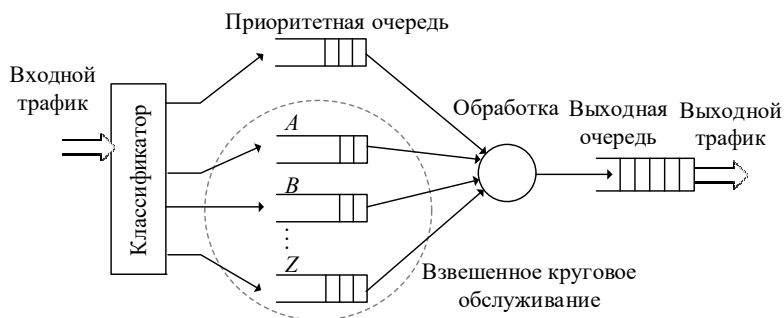
Здесь, как и в приоритетном обслуживании, администратор может назначать различные длины очередям. Тем самым появляется возможность отбрасывания пакетов, то есть сглаживания нагрузки.

### 3а) Взвешенное справедливое обслуживание (Weighted Fair Queuing – WFQ)

Это сочетание приоритетного обслуживания со взвешенным.

Существует большое число реализаций WFQ в оборудовании разных производителей. Наиболее распространена следующая схема.

Имеется одна приоритетная очередь, все заявки из которой обслуживаются в первую очередь. Эта очередь предназначена для системных сообщений, сообщений управления сетью и наиболее критичных и требовательных приложений. Предполагается, что обслуживаемый этой очередью трафик имеет невысокую интенсивность (тем самым остается еще и пропускная способность для других очередей). Остальные очереди просматриваются маршрутизатором по алгоритму взвешенного обслуживания. Веса задаются администратором (рис. 7.5).



**Рис. 7.5. Взвешенное справедливое обслуживание**

**Механизм кондиционирования трафика** не является обязательным. Его задача – это уменьшение скорости входного потока настолько, чтобы она всегда оставалась меньшей, чем скорость продвижения этого потока в узле.

Этот механизм включает в себя выполнение следующих функций.

- *Классификация трафика.* Задача – выделить в общем потоке пакеты одного потока. Для классификации используются все возможные параметры потока: узел назначения, идентификатор приложения, значение приоритета, метка потока и т. д.

- *Профилирование потока* на основе правил политики (Policing). Для каждого входного потока имеется его набор параметров QoS – профиль трафика. Профилирование трафика – это проверка соответствия потока параметрам его профиля. Например, если превышена согласованная скорость, производится отбрасывание или маркировка (для возможности удаления) пакетов. Отбрасывание позволяет снизить интенсивность потока. При маркировке пакеты сохраняются, но снижается качество их обслуживания. Для проверки используется один из алгоритмов, например, алгоритм «дырявого ведра» (Leaky bucket).

- *Формирование трафика* (Shaping). Здесь, в основном, стараются сгладить пульсацию трафика, чтобы на выходе поток был более равномерным, чем на входе. Это уменьшает очереди в последующих устройствах, а также позволяет сделать поток более равномерным (что, например, полезно в голосовых приложениях).

#### **Механизмы профилирования и формирования трафика.**

Существует несколько популярных алгоритмов, которые рекомендуется применять для профилирования и формирования трафика, с целью обеспечения требуемого QoS.

**Алгоритм «дырявого ведра» (Leaky bucket)** разработан для профилирования пульсирующего трафика. Он позволяет проверить соблюдение трафиком оговоренных (в соглашении об уровне услуг) значений средней скорости и пульсаций.

У алгоритма имеются следующие настраиваемые значения:

- период усреднения скорости  $T$ ;
- средняя скорость, которую трафик не должен превышать на периоде  $T$  (скорость, согласованная с провайдером);

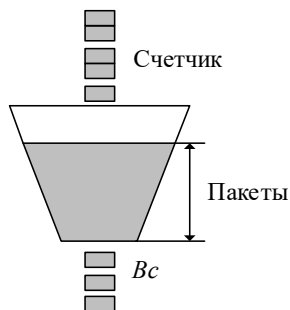
- объем пульсации  $B_c$ , соответствующий средней скорости и периоду  $T$ ;
- допустимое превышение объема пульсации  $B_e$ . Предполагается, что трафик контролируется каждые  $T$  секунд.

Трафик должен иметь среднюю скорость не выше оговоренной скорости средней скорости.

Превышение объемом пульсации оговоренного значения  $B_c$  на величину  $B_e$  считается мягким нарушением. Пакеты-нарушители помечаются специальным признаком, но не удаляются.

При превышении объемом пульсации величины  $B_c + B_e$  пакеты отбрасываются. Но фактически предпринимаемые при этом действия являются настраиваемым параметром.

Алгоритм использует счетчик поступивших от пользователя байтов. Каждые  $T$  секунд счетчик уменьшается на величину  $B_c$  (или сбрасывается в 0, если его значение меньше  $B_c$ ). Это иллюстрируется обычно ведром, из которого дискретно каждые  $T$  секунд вытекает объем накопленного трафика (рис. 7.6).



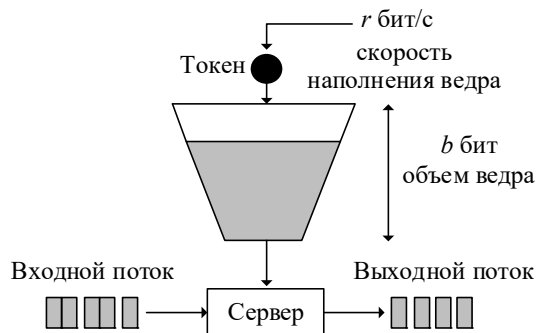
**Рис. 7.6. Алгоритм «Leaky bucket»**

Все кадры, которые находятся в объеме, не превышающем  $B_c$ , пропускаются в сеть со значением признака  $DE=0$  (нормальная доставка). Кадры, находящиеся в промежутке  $(B_c) \div (B_c + B_e)$ , тоже передаются в сеть, но с признаком  $DE=1$  (возможность удаления на следующих маршрутизаторах сети). Те кадры, которые находятся за пределами объема  $(B_c + B_e)$ , отбрасываются маршрутизатором.

Такой вариант алгоритма используется в сети Framy Relay.

**Алгоритм «Ведро меток» (Token bucket)** используется как для профилирования, так и для формирования, т.е. сглаживания трафика. Цель алгоритма – уменьшение неравномерности продвижения пакетов, когда из-за значительной пульсации они сбиваются в плотные группы.

Под меткой (Token) понимается абстрактный объект, носитель «порции» информации. Генератор меток периодически направляет очередную метку в «ведро» с ограниченным объемом  $b$  байт. Все метки имеют одинаковый объем  $m$  байт, а генерация меток происходит с такой скоростью, что ведро заполняется со скоростью  $r$  байт в секунду. Скорость  $r$  является желательной средней скоростью для формируемого трафика (рис. 7.7).



**Рис. 7.7. Алгоритм Token bucket**

Пакеты поступают в систему и попадают в очередь объемом  $K$  байт. Из очереди пакет продвигается сервером только в том случае, когда к этому моменту «ведро» наполнено до уровня  $M$  байт, где  $M$  – объем пакета. Пакет в этом случае продвигается вперед, а из «ведра» удаляется объем в  $M$  байт. Таким образом, достигается улучшение трафика. Даже если приходит большое число пакетов, из очереди они выходят равномерно и в темпе, задаваемом генератором меток. То есть поток меток – это идеальный трафик, к форме которого стараются привести входной трафик.

Расчет длительности выходной пачки:

$S$  – длительность пачки/с;

$C$  – емкость маркерного ведра/байт;

$\rho$  – скорость появления маркеров, байт/с;

$M$  – максимальная выходная скорость, байт/с.

Максимальное количество переданных байтов в пачке будет равно  $(C + \rho S)$ .

$M S$  – количество байтов, переданных в пачке с максимальной скоростью.

$C + \rho S = M S$ , отсюда

$S = C / (M - \rho)$

**Протоколы сигнализации QoS** используются для обмена служебной информацией, необходимой для обеспечения QoS (резервирование вдоль маршрута требуемой пропускной способности, использование маркировки пакетов признаком, указывающим на требуемое качество обслуживания).

**Политики QoS.** Службы QoS, в которых используют централизованные службы поддержки политики, называются службами, основанными на политике (*Policy based QoS*).

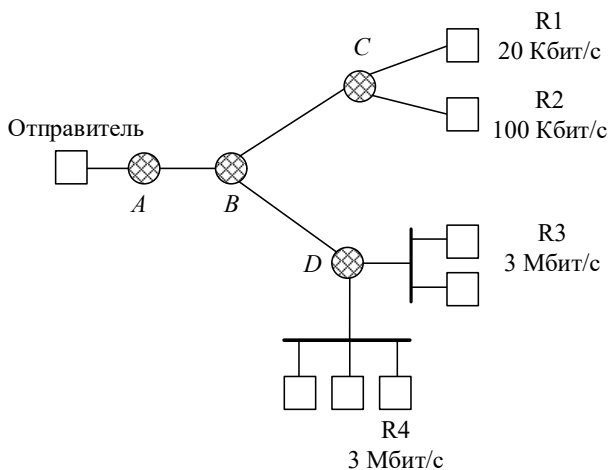
Правила политики применяются не только для управления QoS, но и для координации выполнения сетевыми устройствами других функций, например, функций защиты трафика. Централизованная служба политики сети обычно базируется на общей справочной службе сети (*Directory Service*), хранящей все учетные данные о пользователях (имя, пароль, ...). В последнее время ее функции расширены – добавлено хранение самых разных данных о сети, в том числе и данных о политике QoS, политике безопасности и т. д.

**Протокол RSVP.** Чтобы сеть могла предоставлять гарантии качества обслуживания, нужен специальный механизм, позволяющий работающим на хостах приложениям резервировать ресурсы в Интернете. Таким механизмом является протокол RSVP (ReSerVation Protocol) – протокол резервирования. Его задача – резервирование пропускной способности линий и буферов маршрутизаторов. Используя этот протокол, хост от имени приложения запрашивает у сети определенный объем пропускной способности. Маршрутизаторы с помощью протокола RSVP пересылают запросы на резервирование ресурсов. Два свойства протокола RSVP:

- он может резервировать пропускную способность в деревьях групповой рассылки;
- резервирование инициируется и управляется получателем потока данных.

Однако протокол RSVP не определяет, каким образом сеть предоставляет зарезервированную пропускную способность. Этим занимаются маршрутизаторы (используя механизмы планирования – приоритетное обслуживание, взвешенная справедливая очередь и т. д.). RSVP не определяет также и те линии, на которых должна быть зарезервирована пропускная способность. Это тоже функция маршрутизаторов.

Рассмотрим пример групповой рассылки (рис. 7.8).



**Рис. 7.8. Пример резервирования при групповой рассылке**

Имеется источник, передающий в Интернет видео о спортивном соревновании. Видеоданные передаются с разными уровнями качества – 20, 100, 3000 Кбит/с. Каждый получатель посылает вверх по дереву групповой рассылки запрос на резервирование ресурсов. В нем указывается требуемая скорость.

Запрос обрабатывается планировщиком узла, а затем пересылается вверх по дереву. Пользователи, подключенные к узлу D, резервируют 3 Мбит/с. Маршрутизатор D по линии D-B посылает запрос на резервирование маршрутизатору B (то есть здесь объединяются запросы от R3 и R4 – суммарная пропускная способность – 3 Мбит/с).



Аналогично, на узле *C* резервируются пропускные способности 20 и 100 Кбит/с от R1 и R2. Узел *C* обрабатывает запрос и пересылает его в узел *D* (запрос на 100 Мбит/с). Здесь поток 20 Кбит/с включают в поток 100 Кбит/с.

Узел *B* обрабатывает запросы и по линии *B-A* передает на узел *A* запрос на резервирование 3 Мбит/с (100 Кбит/с включается внутрь этого потока).

Другой пример – проведение видеоконференции (рис. 7.9).

Положим, что в видеоконференции участвуют 4 собеседника, каждый из которых открывает на экране 3 окна.

Каждый из пользователей хочет получать от своих собеседников высококачественный поток 3 Мбит/с. Таким образом, надо резервировать пропускную способность: 3 Мбит/с – от пользователя и 9 Мбит/с – к пользователю.

Объединить потоки (как в предыдущем примере) здесь нельзя.

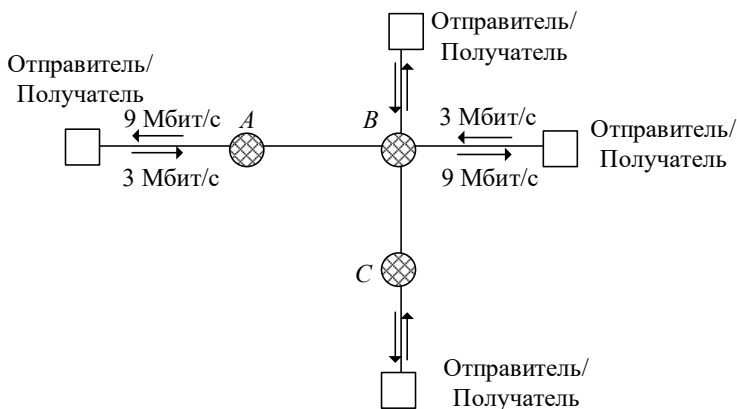


Рис. 7.9. Проведение видеоконференции

**Реализация резервирования.** Резервирование на маршрутизаторах и узлах реализуется в форме так называемого неустойчивого состояния.

С каждым запросом на резервирование на маршрутизаторе связывается таймер. Когда указанный интервал времени истекает, резервирование отменяется и ресурсы освобождаются.

Если получатель хочет продлить резервирование, он должен периодически повторять запросы.

**Ограничение.** Маршрутизатор, получив запрос, проверяет его на допустимость – имеются ли требуемые пропускные способности. Если таковых нет – запрос отвергается.

**Стандарт на протокол.** Протокол RSVP описан в документе RFC 2205. Он работает на уровне выше протоколов IP и UDP (т.е. уже на прикладном уровне).

**Возможность подтверждения.** Получатель может включать в запрос на резервирование **RESV** еще запрос на подтверждение этого резервирования. При успешном резервировании он получает подтверждение **RESV Conf.**

## 7.2. Службы трансляции имен интернета

### 7.2.1. Функции DNS

Чтобы установить связь между двумя идентификаторами хоста – именем и IP-адресом, используется система доменных имен (Domain Name System, DNS).

DNS – это база данных, распределенная между иерархически структурированными серверами имен, а также протокол прикладного уровня, организующий взаимодействие между хостами и серверами имен для выполнения операции преобразования.

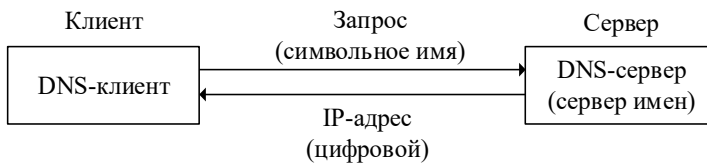
Протокол DNS работает поверх протокола UDP.

Упрощенно процедура выглядит следующим образом (рис. 7.10):

1) пользователь запрашивает (вводит) символьное имя хоста, а далее по цепочке

ИМЯ → ПРИКЛАДНАЯ ПРОГРАММА → РАСПОЗНАВАТЕЛЬ (имя – параметр) → UDP → локальный DNS-сервер → IP-адрес → ПРИКЛАДНАЯ ПРОГРАММА

2) прикладная программа открывает TCP-соединение с адресатом, содержащимся на хосте с полученным IP-адресом.



**Рис. 7.10. Обычная процедура «клиент-сервер»**

Это обычная процедура «клиент-сервер».

В последнее время DNS все чаще используется для распределения загрузки между дублирующими серверами.

Популярные сайты (например, CNN) имеют несколько копий (реплик, зеркал), размещенных на различных серверах с разными IP-адресами. В этом случае с одним именем связываются несколько IP-адресов, хранимых в базе данных DNS (рис. 7.11). Когда происходит запрос по имени, в ответ включаются все IP-адреса, однако сервер может изменять порядок их перечисления.

Имя A	IP1
	IP2
	IP3
Имя B	IP4
	IP5
...	...

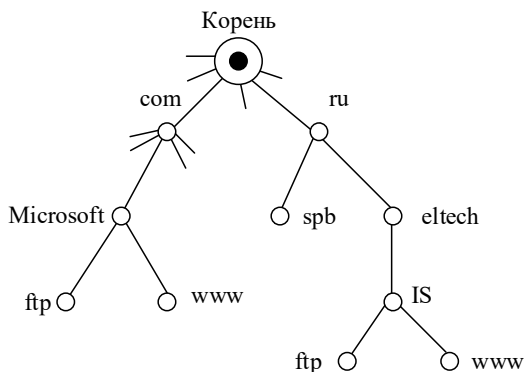
**Рис. 7.11. Одно имя – несколько IP-адресов**

HTTP-клиент выбирает первый адрес из полученного списка, что позволяет распределять загрузку между дублирующими серверами, то есть этот механизм позволяет распределять web-ресурсы.

Описание DNS содержится в RFC 1034 и 1035.

### 7.2.2. Иерархия службы имен

Используемая словесная форма записи имеет иерархическую доменную структуру, которая может иметь произвольное число уровней (рис. 7.12).



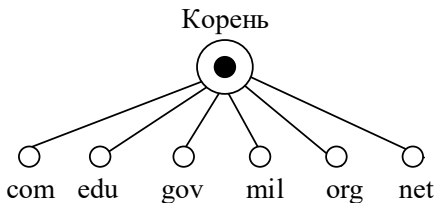
**Рис. 7.12. Иерархическая доменная структура**

Дерево начинается с точки (•), обозначающей корень. Затем идут разделяемые точкой части символического имени.

Количество уровней не лимитируется, однако редко превышает 5.

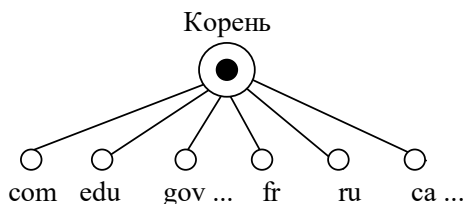
Совокупность имен, у которых старшие части совпадают, образует домен. Компьютеры, входящие в домен, могут иметь абсолютно отличающиеся цифровые IP-адреса, например: 132.4.12.110; 14.134.15.11.

Корневой домен (1-го уровня) управляется в Интернет центром InterNIC (центр сетевой информации). Его работа определена стандартом ISO 3266. В соответствии с ним введены двух- и трехбуквенные аббревиатуры для стран и различных организаций. Так как сеть возникла в США, изначально было введено 6 доменов высшего уровня (рис. 7.13):



**Рис. 7.13. Структура сети Интернет США: com – коммерческие организации, mil – военные учреждения (США), gov – государственные организации, org – прочие организации, net – сетевые ресурсы, edu – образовательные организации**

Когда Интернет стал международной сетью, были добавлены домены для стран-участниц: fr (Франция), ru (Россия), ca (Канада) и другие (рис. 7.14).



**Рис. 7.14. Структура международной сети Интернет**

Каждый такой домен администрируется отдельной организацией, которая разбивает его на поддомены. В России для домена «.ru» этим занимается РосНИИРОС.

### **7.2.3. Общие принципы функционирования DNS**

Система DNS спроектирована в виде иерархической структуры серверов, разбросанных по всему миру. Она строится как распределенная база данных. Ни один сервер имен не содержит информацию обо всех IP-хостах.

Укрупненно DNS-серверы делятся на:

- локальные;
- корневые;
- полномочные.

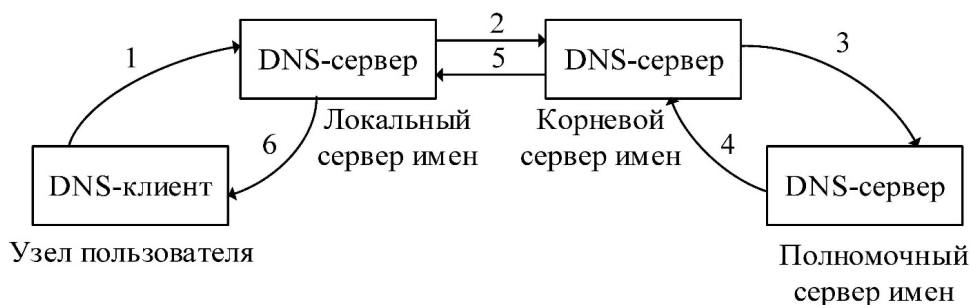
*Локальные DNS-серверы* имеются у каждого интернет-провайдера. Когда DNS-клиент посылает запрос, тот сначала поступает на локальный сервер имен. Адрес такого локального сервера имен часто конфигурируется пользователем вручную. Если запрашиваемый хост принадлежит тому же интернет-провайдеру, сразу же будет отослан ответ с IP-адресом.

*Корневые серверы имен* – следующая ступень в иерархии серверов DNS. Их число в мире составляет немногим более 10, и большая их часть находится в США, а также в Лондоне, Стокгольме и Токио.

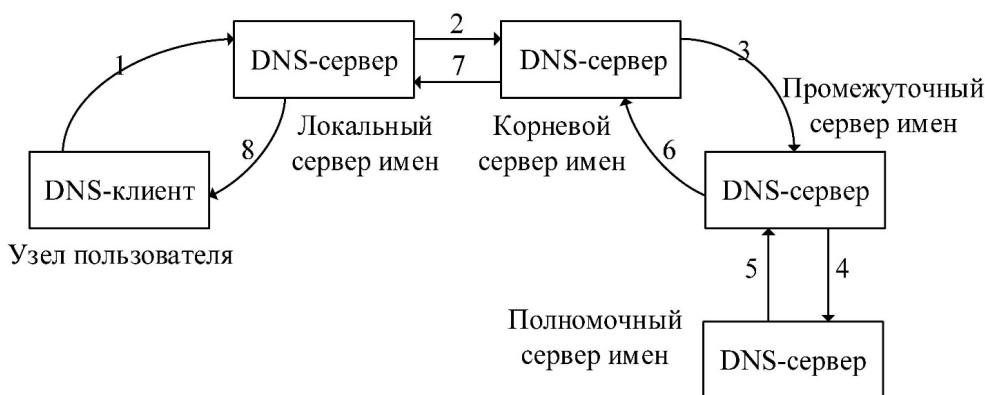
Если локальный сервер не может удовлетворить запрос, он берет на себя роль клиента и передает запрос одному из корневых серверов сети. Если в базе сервера есть такой адрес – он посылается в виде ответа локальному серверу, который в свою очередь, передает его далее пользовательскому хосту. Но база данных корневого сервера ограничена, и он не всегда может выполнить запрос. Если адрес не найден, локальному серверу имен отсылается IP-адрес полномочного сервера имен, который располагает искомым IP-адресом.

*Полномочный сервер имен* – это тот сервер, на котором зарегистрирован данный хост. Полномочный сервер отсылает ответ корневому серверу, тот ретранслирует его локальному серверу, а тот – хосту пользователя (рис. 7.15).

Возможен и такой вариант, когда корневой сервер имен не знает полномочный сервер, а знает только адрес промежуточного сервера имен. Такая последовательность рекурсивных запросов представлена на рис. 7.16.



**Рис. 7.15. Последовательность рекурсивных запросов (требуется выполнить 6 запросов-ответов)**



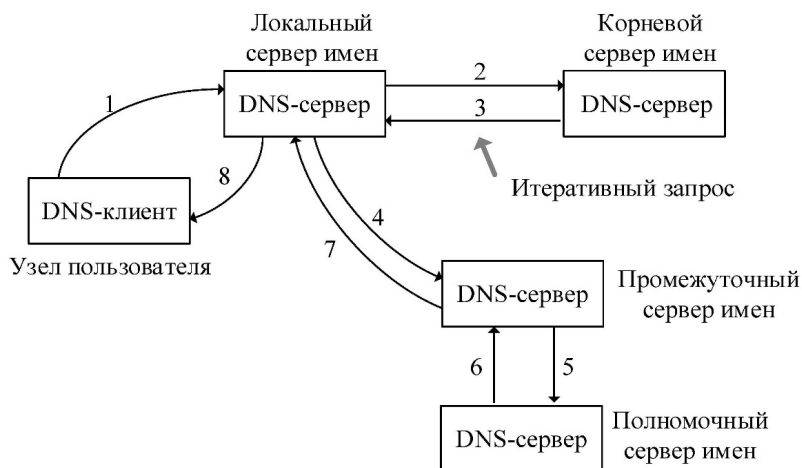
**Рис. 7.16. Последовательность рекурсивных запросов к полномочному серверу через промежуточный**

На практике же бывают случаи, когда между корневым и полномочным серверами находятся 2 и более промежуточных сервера. Это еще более увеличивает число запросов-ответов (и, соответственно, время поиска).

В рассмотренных случаях все запросы были рекурсивными, то есть если сервер *A* обращается к серверу *B*, то тот предпринимает необходимые действия для получения IP-адреса и затем передает адрес серверу *A*.

Протокол DNS предусматривает также итеративные запросы. Они отличаются от рекурсивных тем, что в случае отсутствия искомого IP-адреса, сервер имен *B* возвращает *A* IP-адрес следующего сервера в цепочке, к которому *A* должен обратиться уже самостоятельно (рис. 7.17). То есть, последовательность может содержать как рекурсивные, так и итеративные запросы.

Для сокращения числа запросов и времени получения IP-адресов хостами в DNS используется механизм хэширования. Обычно записи остаются в хэш-памяти ограниченное время (чаще всего это 48 часов). Хэширование поддерживается всеми серверами имен.

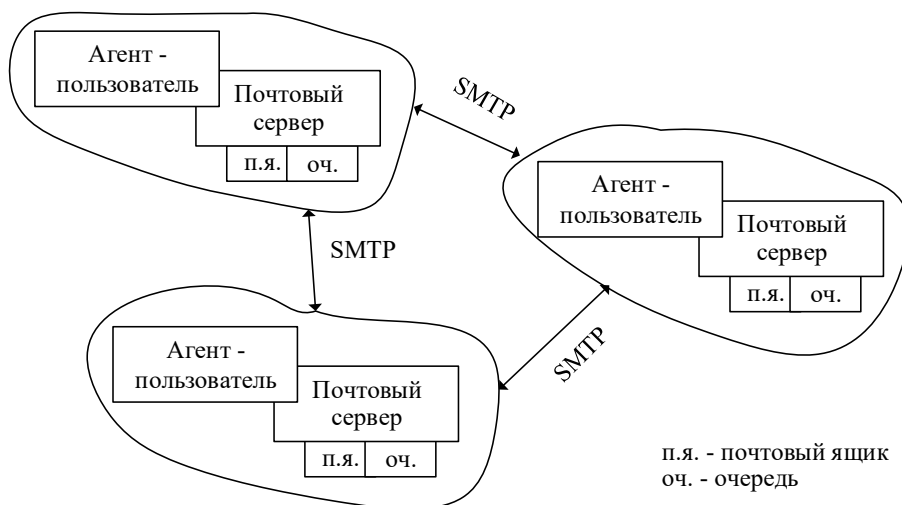


**Рис. 7.17. Пример цепи с рекурсивными и итеративными запросами**

## 7.3. Электронная почта

### 7.3.1. Основные элементы службы электронной почты

Общая структура системы электронной почты включает три ключевых компонента: агенты пользователя; почтовые серверы; протокол SMTP (рис. 7.18).



**Рис. 7.18. Общая структура системы электронной почты**

Агенты пользователя – программы, позволяющие читать, пересылать, создавать, сохранять электронные письма. К числу наиболее популярных агентов относятся Eudora, Microsoft Outlook, Netscape Messenger.

Почтовые сервера составляют ядро инфраструктуры электронной почты Интернета.

Каждый пользователь электронной почты обладает собственным почтовым ящиком, расположенном на почтовом сервере.

Протокол SMTP является главным протоколом прикладного уровня для доставки электронной почты. Для надежной доставки протокол SMTP использует механизм транспортного протокола TCP.

Протокол работает по принципу «клиент–сервер». Сторона клиента выполняется на почтовом сервере отправителя, а сторона сервера – на почтовом сервере получателя. Каждый почтовый сервер в процессе работы может менять свою роль, и принимая, и отправляя сообщения.

**Протокол SMTP** (Simple Mail Transfer Protocol) составляет основу службы электронной почты, описан в документе RFC 2821.

Упрощенно процедура пересылки письма выглядит следующим образом (рис. 7.19).

1. Пользователь запускает агент электронной почты, вводит текст, адрес и дает агенту команду на отсылку сообщения.

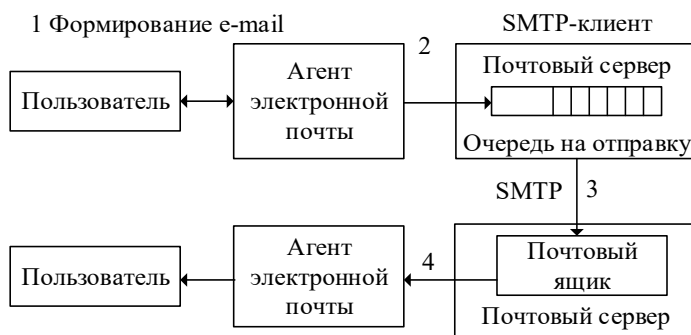
2. Агент пользователя отправляет сообщение почтовому серверу, где оно ставится в очередь исходящих сообщений.

3. SMTP-клиент на почтовом сервере обнаруживает сообщение в очереди и устанавливает TCP-соединение с серверной стороной SMTP, выполняющееся на почтовом сервере получателя.

4. После установления TCP-соединения SMTP-клиент пересылает SMTP-серверу письмо.

5. Сервер принимает сообщение и помещает его в почтовый ящик получателя.

6. Получатель в удобное время запускает свой агент электронной почты, получает доступ к почтовому ящику и читает сообщение.



**Рис. 7.19. Процедура пересылки письма**

Протокол SMTP обычно не предусматривает передачу сообщений через промежуточные почтовые сервера, даже если клиент и сервер разделены тысячами километров. Если соединение с сервером установить не удалось, то через определенное время предпринимается следующая попытка отсылки сообщения.

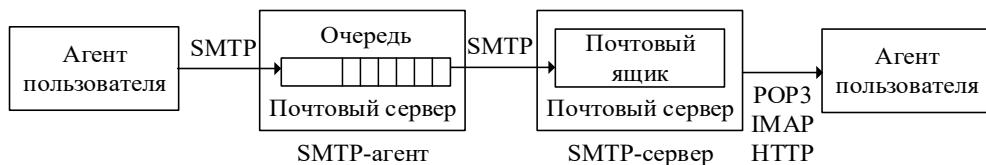
**Принимаемые сообщения.** После получения сообщения почтовый сервер добавляет в начало сообщения строку: *Received*, содержащую адрес отправителя, адрес получателя, и время получения сообщения сервером.

Если сообщение проходит через несколько почтовых серверов, каждый из них добавляет свою строку заголовка. Это позволяет проследить путь письма по сети.

**Протоколы доступа к электронной почте.** Рассмотренный метод обмена предполагает, что агент пользователя и SMTP-клиент и сервер располагаются в непосредственной близости. Но на самом деле обычно SMTP-клиент и сервер размещаются у интернет-провайдера или на общедоступном сервере (типа mail.ru), к которому пользователи подключаются только периодически со своего компьютера.

Поступившее сообщение находится в почтовом ящике пользователя, и этот ящик размещается на сервере. Протокол SMTP является протоколом отправки, а здесь требуется протокол получения. То есть необходим специальный протокол получения электронной почты, находящейся в почтовом ящике сервера (рис. 7.20).

Существует несколько таких протоколов и наиболее распространенными из них являются: POP3 – post Office Protocol, version 3, IMAP – Internet Mail Access Protocol.



**Рис. 7.20. Протоколы электронной почты**

В этой схеме протоколы POP3, IMAP или HTTP<sup>8</sup> используются для получения писем на компьютере пользователя.

**Протокол POP3** описан в документе RFC 1939 и устанавливает (с помощью агента пользователя) TCP-соединение с портом 110 почтового сервера.

Выполняются 3 основные фазы:

- авторизация;
- транзакция;
- обновление.

Во время авторизации агент пользователя передает серверу имя пользователя и пароль, чтобы получить право доступа к сообщениям электронной почты.

На фазе транзакции пользователь получает сообщения и сможет получить статистику почтовой связи, а также помечает сообщение для удаления.

**Протокол IMAP** описан в документе RFC 2060. Для пользователя, который входит на почтовый сервер с разных компьютеров (дома, на работе, в пу-

<sup>8</sup> HyperText Transfer Protocol – протокол передачи гипертекста



ти), удобно было бы организовать на почтовом сервере иерархию папок. Именно с этой целью и был разработан протокол IMAP. В этом протоколе сложнее как клиентская, так и серверная части протокола.

IMAP поддерживает операции создания, удаления, переименования почтовых ящиков; проверки поступления новых писем; оперативное удаление писем; установку и сброс флагов операций; поиск среди писем; выборочное чтение писем.

Протокол IMAP базируется на транспортном протоколе TCP и использует порт 143. Протокол IMAP представляет собой альтернативу POP-3. Так же как и последний, он работает только с сообщениями и не требует каких-либо пакетов со специальными заголовками. Протокол позволяет получать доступ к письму не только по его номеру, а и по содержанию.

IMAP-сервер связывает каждое сообщение с некоторой пользовательской папкой. Изначально каждое сообщение попадает в папку INBOX, где пользователь может прочитать его, а затем переместить в другую папку или удалить. Для всех этих действий в протоколе IMAP имеются специальные команды.

В отличие от POP3, IMAP хранит почтовые сообщения у себя «вечно» (пока клиент сам не пожелает их стереть).

При доступе к электронной почте через web-интерфейс роль агента пользователя играет web-браузер, который взаимодействует с удаленным почтовым ящиком по протоколу HTTP. Однако взаимодействие между самими почтовыми серверами выполняется по протоколу SMTP.

### **7.3.2. Угрозы безопасности электронной почты**

Использование электронной почты сопряжено с рядом проблем по безопасности.

1) Адреса электронной почты в Интернете легко подделать. Практически нельзя сказать наверняка, кто написал и послал электронное письмо, только на основе его адреса.

2) Электронные письма могут быть легко модифицированы. Стандартное SMTP-письмо не содержит средств проверки целостности.

3) Существует ряд мест, где содержимое письма может быть прочитано теми, кому оно не предназначено. Электронное письмо скорее похоже на открытку – его могут прочитать на каждой промежуточной станции.

4) Обычно нет гарантий доставки электронного письма. Некоторые почтовые системы предоставляют возможность получить сообщение о доставке. Однако часто такие уведомления означают лишь то, что почтовый сервер получателя (а не обязательно сам пользователь) получил сообщение.

Перечисленные проблемы безопасности службы электронной почты создают соответствующие угрозы, которые можно классифицировать следующим образом.

**Случайные ошибки.** Письмо может быть случайно послано по неправильному адресу. Архивы писем могут возрасти до такой степени, что система будет аварийно завершаться. Неправильно настроенная программа чтения групп новостей может привести к отправке сообщения не в те группы. Ошибки

в списках рассылки могут привести к долгому блужданию писем между почтовыми серверами, причем число писем может увеличиться до такой степени, что почтовые сервера аварийно завершатся.

Почтовые сообщения могут храниться годами, поэтому случайная ошибка может нанести вред через много времени.

Последствия ошибок существенно возрастают, если почтовая система организации присоединена к Интернету.

**Фальшивые адреса отправителя.** Протоколы электронной почты SMTP, POP, IMAP обычно не обеспечивают надежной аутентификации. Это позволяет создавать письма с фальшивыми адресами. Некоторые расширения этих протоколов используют уже имеющиеся средства аутентификации, а другие позволяют клиенту и серверу согласовать тип аутентификации, который будет использоваться в данном соединении. Криптографию эти протоколы не используют.

В электронной почте Интернета отправитель может указать фальшивый обратный адрес, или заголовок может быть модифицирован в ходе передачи письма, или отправитель может сам соединиться с SMTP-портом на машине, от имени которой он хочет отправить письмо, и ввести текст письма.

**Перехват письма.** Заголовки и содержимое электронных писем передаются в открытом виде. В результате содержимое сообщения может быть прочитано или изменено в процессе передачи его по Internet. Заголовок может быть модифицирован, чтобы скрыть или изменить отправителя или чтобы перенаправить сообщение.

**Почтовые бомбы.** Почтовая бомба – это атака с помощью электронной почты. Атакуемая система переполняется письмами до тех пор, пока она не выйдет из строя. Как это может случиться, зависит от типа почтового сервера и того, как он сконфигурирован. Типовые варианты выхода почтового сервера из строя:

- почтовые сообщения принимаются до тех пор, пока диск, где они размещаются, не переполнится. Следующие письма не принимаются. Если этот диск – также основной системный диск, то вся система может аварийно завершиться;
- входная очередь переполняется сообщениями до тех пор, пока не будет достигнут предельный размер очереди. Последующие сообщения не попадут в очередь;
- у некоторых почтовых систем можно установить максимальное число почтовых сообщений или максимальный общий размер сообщений, которые пользователь может принять за один раз. Последующие сообщения будут отвергнуты или уничтожены;
- может быть превышена квота диска для данного пользователя. Это мешает принять последующие письма и может помешать ему выполнять другие действия. Большой размер почтового ящика может сделать трудным для системного администратора получение системных предупреждений и сообщений об ошибках;

- **вредоносные программы.** Вредоносное программное обеспечение, пересылаемое вместе с электронным сообщением, может нанести существенный ущерб серверам, рабочим станциям и находящейся в них информации – исказить или уничтожить данные, блокировать работу приложений и операционной системы в целом.

**Фишинговые ссылки.** Фишинг – вид интернет-мошенничества, целью которого является получение доступа к конфиденциальным данным пользователей – логинам и паролям. Это достигается путём проведения массовых рассылок электронных писем от имени популярных брендов, а также личных сообщений внутри различных сервисов (от имени банков, внутри социальных сетей). В письме часто содержится прямая ссылка на сайт, внешне неотличимый от настоящего. Переход по фишинговым ссылкам на хакерские сайты грозит тем, что на компьютеры пользователей будут незаметно установлены программы, позволяющие получить злоумышленникам доступ к ценной персональной информации, логинам и паролям от корпоративных ресурсов.

**Спам.** Через электронную почту распространяется самый большой поток спама. В настоящее время доля вирусов и спама в общем трафике электронной почты составляет по разным оценкам от 70 до 95 процентов.

Спам (англ. *spam*) – массовая рассылка коммерческой, политической и иной рекламы (информации) или иного вида сообщений лицам, не выразившим желания их получать.

Такие сообщения не только серьезно загружают память, но и ежедневно отвлекают сотрудников от выполнения служебных обязанностей.

Массовая рассылка спама имеет низкую себестоимость для отправителя в расчёте на сообщение. Однако огромное количество бесполезных сообщений наносит очевидный вред получателям. В первую очередь речь идёт о времени, потраченном впустую на отсеивание ненужной почты и выискивании среди неё отдельных нужных писем. Очень часто интернет-трафик стоит дорого, и пользователю приходится платить за очевидно ненужные письма. Кроме того, провайдерам приходится тратить ресурсы на избыточное оборудование и системы защиты от спама (избыточное оборудование, избыточная ёмкость каналов, специальное программное обеспечение для распознавания спама). Спам также наносит вред репутации компании, если спам используется в недобросовестной конкуренции и «чёрном» пиаре.

Сбор e-mail адресов для рассылки спама осуществляется с помощью специального робота, используя веб-страницы, конференции, списки рассылки, электронные доски объявлений, гостевые книги, чаты и другое. Такая программа-робот способна собрать за час тысячи адресов и создать из них базу данных для дальнейшей рассылки по ним спама.

**Безопасность** электронной почты обеспечивается рядом мероприятий: не допускать случайные ошибки, использовать технологию электронных подписей, шифрования сообщений, межсетевые экраны, антивирусные программы и сканирующие сообщения, программные фильтры и антиспам-системы.

Защита писем, почтовых серверов и программ должна соответствовать важности информации, передаваемой по корпоративным сетям. Должна быть разработана политика, в которой указывался бы нужный уровень защиты.

### **Контрольные вопросы**

1. Какие основные характеристики сети обеспечивают качество обслуживания (QoS)?
2. В чем проявляются требования разных типов приложений к качеству обслуживания?
3. Какие типы элементов обеспечивают службу QoS?
4. Какие алгоритмы используются для управления очередями?
5. В чем заключается задача механизма кондиционирования трафика?
6. Для чего используется алгоритм «дырявого окна»?
7. В каких случаях используется алгоритм «ведро меток»?
8. Каково назначение протокола резервирования RSVP?
9. В чем назначение службы трансляции имен Интернета?
10. Нарисуйте цепочку переходов от обычного имени пользователя к IP-адресу.
11. Для чего введена иерархия службы имен?
12. Чем отличаются полномочные серверы от конечных в службе имен?
13. Назовите основные элементы электронной почты.
14. Чем различаются протоколы POP3 и IMAP?
15. Что такое почтовые бомбы?

## 8. ПРАКТИКУМ

### 8.1. Исследование информационного канала

**Цель работы:** исследование процедур протоколов управления информационным каналом и выбор оптимальных системных параметров протокола для заданных условий обмена.

#### Общие сведения

Протоколы управления информационным каналом являются важнейшим элементом информационной сети с точки зрения обеспечения требуемых характеристик передачи данных – времени доставки, скорости обмена и надежности передачи. В рамках семиуровневой модели взаимосвязи открытых систем (ВОС) данные протоколы обеспечивают сервис уровня информационного канала (уровень 2), на котором базируются все вышележащие протокольные уровни (сетевой, транспортный и т. д.).

Основной задачей протокола 2-го уровня является надежная и своевременная доставка пользовательских данных по двухточечному соединению. Семантический анализ этих пользовательских данных не входит в задачи протокола, однако он должен обеспечить их кодовую прозрачность, то есть возможность любых кодовых (битовых) сочетаний. Единицей пользовательских данных в протоколе выступает некоторая структура с ограничением по максимуму длины, которая называется в разных протоколах пакетом, сегментом, фрагментом и т.д. Будем использовать название **пакет**, как наиболее употребительное.

В зависимости от способов обеспечения кодовой прозрачности передаваемого пакета протоколы 2-го уровня делятся на бит- и байт-ориентированные. В современных сетях наибольшее распространение получили бит-ориентированные протоколы (SDLC, HDLC, X.25/2). Структура информационного кадра приведена в разделе 2.2.1 на рис. 2.13.

Для целей управления передачей информации в протоколах класса HDLC используются служебные кадры (см. разд. 2.2.4).

Функционирование информационного канала разбивается на ряд последовательных фаз:

- установления соединения;
- передачи данных;
- разъединения соединения.

При обнаружении неисправимых ошибок на фазе передачи данных возможен переход в фазу повторного установления соединения (рестарта).

Протоколы информационного канала используют различные методы (процедуры) для борьбы с ошибками, вызванными ненадежностью среды передачи (канала связи-КС). Отметим важнейшие из них.

1. Основным элементом для борьбы с ошибками на уровне структуры кадра выступает **контрольная последовательность кадра FCS**, представляющая собой остаток от деления внутренней области кадра на образующий полином циклического кода. Наибольшее распространение получил в современных сетях образующий полином 16-й степени.

Циклический код используется **в режиме обнаружения ошибок**. Если на приемной стороне процедура декодирования обнаруживает несовпадение синдрома ошибки, то производится стирание принятого кадра.

2. В информационном канале производится **последовательная нумерация передаваемых информационных кадров**. Каждому кадру, содержащему пакет данных, присваивается последовательный номер передачи  $N(S)$ . Для сокращения размера поля номера  $N(S)$  в заголовке кадра применяется нумерация по модулю 8 (иногда 128). Это позволяет сократить размер поля номера до 3-х (в случае модуля 128 – до семи) бит.

Процедура последовательной нумерации позволяет станции-приемнику следить за отсутствием пропуска кадров, использовать различные процедуры решающей обратной связи (РОС).

3. В случае обнаружения ошибок на уровне последовательной нумерации информационных кадров станция-приемник может сообщить об этой ситуации с помощью **специальных управляющих кадров (REJ, SREJ)**.

4. **Подтверждение доставки данных** обеспечивается процедурой РОС «с положительным квитированием». Станция-приемник сообщает отправителю о поступлении без обнаруженных ошибок информационного кадра с номером  $N(S)$ . Для этой цели используется управляющий кадр (типа  $RR$ ) либо специальное поле в передаваемом во встречном направлении информационном кадре.

5. Для борьбы с «зависанием» информационного канала, когда станция-отправитель в течение длительного времени не получает ни положительных, ни отрицательных квитанций из-за ошибок в обратном канале, используемом для РОС, применяется **восстановление по таймеру**.

Станция-отправитель при посылке информационного кадра в прямой канал запускает таймер  $T_{ож}$  ожидания подтверждения. Если за интервал  $T_{ож}$  не поступит сигнала РОС по обратному каналу, то срабатывание таймера инициирует процедуру восстановления (обычно это повторная передача кадра в прямом канале с запуском таймера).

6. Повышение эффективности использования информационного канала обеспечивается **процедурой передачи некоторого фиксированного количества информационных кадров без ожидания квитанций на их доставку**. Для этой цели станция-отправитель использует системный параметр «окно передачи»  $W$ , задающий разрешение на отправку без получения подтверждения  $W$  кадров информации. Величина параметра  $W$  должна быть меньше модуля циклической нумерации передаваемых информационных кадров.

7. Протоколы уровня информационного канала могут использовать при обмене **три основных метода РОС**.

а) При передаче с ожиданием подтверждения на каждый посланный информационный кадр ( $W = 1$ ) – станция-получатель отправляет положительную квитанцию по обратному каналу в случае приема кадра без обнаруженных ошибок и с ожидаемым порядковым номером  $N(S)$ . Прием кадра с номером  $N(S)$ , не соответствующим ожидаемому, приводит к посылке по обратному каналу отрицательной квитанции с указанием номера ожидаемого кадра.

б) При режиме непрерывной передачи с окном передачи  $W > 1$  – станция-получатель формирует и отправляет по обратному каналу положительные квитанции (кадры типа  $RR$ ) с указанием номера последнего принятого без ошибок и в порядке следования номеров кадра. Прием кадра  $RR$  на станции-отправителе инициирует стирание подтвержденных информационных кадров и смещение в сторону увеличения порядковых номеров окна передачи  $W$ .

Появление ошибки в последовательности принимаемых на станции-получателе информационных кадров вызывает посылку по обратному каналу кадра  $REJ$  с номером последнего принятого без ошибок и в порядке следования номеров кадра. Такой кадр  $REJ$  инициирует процедуру «группового переспроса», т. к. при его получении станция-отправитель стирает подтвержденные информационные кадры и немедленно начинает повторную выдачу в канал связи всех остальных информационных кадров в пределах окна передачи.

в) Разновидностью режима непрерывной передачи, позволяющей в ряде случаев несколько повысить эффективность использования канала, является процедура «избирательного переспроса». В этом случае станция-получатель при ошибке в последовательном номере принятого кадра отправляет по обратному каналу управляющий кадр  $SREJ$ , содержащий номер ожидаемого последовательного кадра. Получаемые на станции кадры не стираются, а хранятся в буфере в ожидании получения недостающего кадра. После его приема вся последовательность передается пользователю (протоколам более высоких уровней). Данный режим используется относительно редко из-за сложности реализации.

Функционирование протокола информационного канала осуществляется с использованием среды передачи (телефонный, радио- или спутниковый канал, коаксиальный кабель, витая пара, волоконно-оптический канал и т. д.), характеризующейся:

1) задержкой распространения сигнала  $T_z$ , зависящей от длины канала  $L_k$  и скорости передачи сигналов (модуляции) в канале  $V_m$ ;

2) ошибками передачи информации, которые зависят как от средней вероятности ошибок в канале на бит передаваемой информации  $P_{ош}$ , так и от характера (модели) распределения ошибок во времени.

Выбор конкретной версии протокола информационного канала включает в себя:

1. Формирование базовой структуры протокола, включающей некоторое подмножество рассмотренных выше процедур для борьбы с ошибками с соответствующим набором управляющих кадров.

2. Выбор системных параметров протокольных процедур, существенно влияющих на эффективность использования канала. К ним относятся:

- максимальная длина информационной части кадра, т.е. длина пакета данных  $I_{max}$ ;
- ширина окна передачи  $W$ ;
- длительность тайм-аута ожидания подтверждения  $T_{ож}$ .

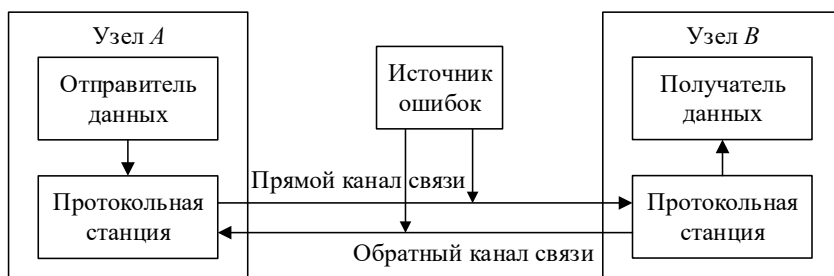
Для выбора оптимальной (для конкретной системы и среды передачи) версии используются различные критерии. Это может быть в зависимости от назначения системы, например:

- минимальное время доставки сообщения абоненту-приемнику ( $T_{\text{дост}} \rightarrow \min$ );
- максимальная вероятность доставки сообщения (пакета) за время, не превышающее максимально заданное –  $P(T_{\text{дост}}, T_{\text{max}}) \rightarrow \max$ ;
- максимум относительной эффективности использования канала связи ( $E \rightarrow \max$ ).

Достаточно часто в качестве критерия используется показатель  $E$ , т. к. это позволяет при проектировании информационной сети обеспечить наилучшее использование пропускных способностей имеющихся каналов связи.

### Описание работы

В данной работе используется имитационная модель информационного канала между двумя станциями (узлами) информационно-вычислительной сети (рис. 8.1).



**Рис. 8.1. Структура имитационной модели информационного канала**

В блоке отправителя данных моделируется простейший протокол верхнего уровня, включающий:

- формирование непрерывной последовательности передаваемых сообщений заданной длины;
- сегментирование сообщения на пакеты длиной  $I_{\text{max}}$ , причем длина последнего пакета может быть меньше  $I_{\text{max}}$ ;
- фиксацию моментов времени выдачи службе информационного канала каждого сообщения и пакета.

В блоке получателя данных соответствующий протокол для сообщений производит сборку сообщения из пакетов и фиксирует время доставки как каждого пакета, так и сообщения в целом.

На этом же уровне модели рассчитывается относительная эффективная скорость передачи данных между узлами на заданном временном интервале (цикле)  $E_i$ :

$$E_i = \frac{N_d}{V_m} T_{\text{ц}},$$

где  $N_d$  – объем пакетов данных (в битах), доставляемых от  $A$  к  $B$  за время цикла;  
 $V_m$  – скорость передачи (модуляции) в канале, бит/с;  
 $T_{\text{ц}}$  – заданный временной интервал (цикл) для оценки величины  $E_i$ .



Протокольная станция узла  $A$  моделирует основные процедуры по формированию информационных кадров, их выдаче в прямой канал связи, ожиданию подтверждений, анализу управляющих кадров, поступающих из обратного канала, выдаче сигналов отправителю данных о возможности начала передачи следующего сообщения.

Протокольная станция узла  $B$  моделирует процедуры анализа поступающих информационных кадров, передачи их получателю данных, выдачи управляющих кадров в обратный канал связи.

Блоки моделей прямого и обратного каналов связи и источника ошибок моделируют задержку распространения сигналов и процесс поражения случайными ошибками кадров в прямом и обратном каналах.

Используемая модель позволяет изменять следующие параметры информационного канала, среды передачи и пользователя:

- $N_c$  – длина передаваемых сообщений (бит);
- $T_{\max}$  – максимальная длина пакета данных (бит);
- $T_z$  – задержка распространения сигнала между станциями  $A$  и  $B$ ;
- $P_{\text{ош}}$  – вероятность ошибки в канале связи (на бит информации);
- $T_{\text{ож}}$  – тайм-аут ожидания подтверждения;
- $W$  – окно передачи.

Выходными параметрами модели являются:

- математическое ожидание  $\bar{T}_{\text{п}}$ , дисперсия  $D_{\text{п}}$  и гистограмма распределения времени доставки пакетов данных  $T_{\text{п}i}$ ;
- математическое ожидание  $\bar{T}_{\text{с}}$ , дисперсия  $D_{\text{с}}$  и гистограмма распределения времени доставки сообщений  $T_{\text{с}i}$  между узлами  $A$  и  $B$ ;
- значение относительной эффективной скорости передачи данных  $E$ , ее дисперсия  $D_E$ , и гистограмма значений  $E_i$  по циклам измерения;
- число доставленных по информационному каналу сообщений;
- число доставленных по информационному каналу пакетов.

### **Порядок выполнения работы**

В таблице исходных данных к данной работе приведены основные параметры исследуемой системы передачи данных:

- $V_{\text{м}}$  – скорость передачи по каналу связи;
- $L_{\text{к}}$  – длина канала связи;
- $N_c$  – длина передаваемых сообщений;
- $P_{\text{ош}}$  – вероятность ошибки в канале связи.

### **Расчет параметров модели.**

Необходимо предварительно выполнить следующие расчеты.

1. В связи с тем, что в модели используется условная единица модельного времени, необходимо провести калибровку модели, рассчитав соответствие между модельным и реальным временем.

Например, исследуется система передачи с  $V_{\text{м}} = 2400$  бит/с. Интервал выдачи одного бита информации в канал связи будет соответствовать 1 единице модельного времени, следовательно, 1 единица модельного времени = 416,6 мкс.

2. Произведите расчет задержки распространения сигнала  $T_3$  в канале связи, исходя из заданной длины канала связи и скорости распространения сигнала ( $0,77c$ , где  $c$  – скорость света).

Например, задано  $L_k = 2000$  км, тогда

$$T_3 = L_k / 0,77c = 8,658 \text{ мс} = 20,81 \text{ единиц модельного времени}$$

С учетом округления до целых получим  $T_3 = 21$  единица модельного времени.

3. Выполните расчет времени  $T_{ож}$  – тайм-аута ожидания подтверждения, исходя из максимальных задержек во всех элементах системы передачи, то есть

$$T_{ож} > T1_{ВК} + T_3 + T2_{АН} + T2_{ВК} + T_3 + T1_{АН},$$

где  $T1_{ВК}$  – время выдачи информационного кадра в канале связи станцией  $A$ ;

$T_3$  – время задержки распространения сигнала в канале связи;

$T2_{АН}$  – время анализа кадра на узле  $B$  (принято равным в модели 1 единице модельного времени);

$T2_{ВК}$  – время выдачи управляющего кадра в КС на узле  $B$ ;

$T1_{АН}$  – время анализа кадра на узле  $A$  (принято равным в модели 1 единице модельного времени).

Длина управляющего кадра, содержащего только заголовок и концевик, составляет 48 бит (см. рис. 2.13), длина информационного кадра увеличивается на принятую величину  $I_{\max}$ .

Например, при  $I_{\max} = 128$  бит получим:

$$T_{ож} > 176 + 21 + 1 + 48 + 21 + 1 = 268 \text{ единиц модельного времени; при } I_{\max} = 256 \text{ бит получим } T_{ож} > 396 \text{ и т.д.}$$

4. Выполните расчет времени моделирования работы информационного канала.

В таблице 8.1 приведено время моделирования работы исследуемого варианта информационного канала в секундах. Для проведения эксперимента необходимо пересчитать это время в единицы условного модельного времени. В п.1 приведена методика калибровки модели исходя из заданной скорости системы передачи. Теперь нужно рассчитать общее время моделирования.

Например,  $V_m = 2400$  бит/с. Было получено: 1 единица модельного времени = 416,6 мкс. Положим, что цикл моделирования (из таблицы 1) составляет 150 с. Получим –  $150 \text{ с} / 416,6 \text{ мкс} = 360000$  единиц модельного времени.

**Этап 1. Сравнение характеристик систем передачи с ожиданием подтверждения и с групповым переспросом.**

Проведите имитационные эксперименты для системы с ожиданием подтверждения на каждый посланный кадр и для системы с групповым переспросом.

Первый тип системы соответствует установке в имитационной модели следующего значения строки, задающей емкость многоканального устройства: 100 BUF1 STORAGE 1;

Для второго типа системы, использующего окно передачи, равное 3, необходимо установить соответствующую емкость многоканального устройства, то есть значение данной строки: 100 BUF1 STORAGE 3.

Остальные параметры исследуемой системы передачи задаются в модели следующим образом:

- длина передаваемых сообщений указывается в операторе INITIAL X1;
- задаваемая в эксперименте длина пакета данных устанавливается с помощью оператора INITIAL X2;
- средняя вероятность ошибки на бит информации задается в операторе INITIAL X4,

при этом указывается величина вероятности, умноженная на  $1E+6$  (то есть для  $I_{\max} = 1E-4$  надо задавать величину 100);

- задержка распространения сигнала в канале связи  $T_3$  указывается в единицах модельного времени (методику расчета см. выше) в операторе INITIAL X5;
- тайм-аут ожидания подтверждения  $T_{\text{ож}}$  (рассчитанный в условных единицах времени по вышеприведенной методике) устанавливается в операторе INITIAL X6;

- время моделирования работы информационного канала (в условных единицах времени) указывается в операторе INITIAL X7.

Для первого и второго типов системы необходимо исследовать зависимость характеристик информационного канала от длины пакета данных  $I_{\max}$ .

На основе проведенной серии экспериментов постройте следующие зависимости:

$$\bar{T}_{\Pi} = f(I_{\max}), \bar{T}_c = f(I_{\max}), E = f(I_{\max}).$$

На основе анализа полученных значений выберите оптимальный по критерию  $E$  (максимума относительной эффективности использования канала связи) системный параметр  $I_{\max}$  для исследуемых типов систем.

### **Этап 2. Оценка влияния величины тайм-аута ожидания подтверждения на характеристики информационного канала.**

Проведите исследование системы с групповым переспросом при величине окна передачи равном 3 и выбранном на предыдущем этапе оптимальном значении  $I_{\max}$ .

Эксперимент предполагает изменение величины  $T_{\text{ож}}$  в пределах порядка  $+100\%$ ,  $-20\%$  от предварительно рассчитанного значения.

Постройте зависимости  $\bar{T}_{\Pi} = f(T_{\text{ож}})$ ,  $\bar{T}_c = f(T_{\text{ож}})$ ,  $E = f(T_{\text{ож}})$ . По результатам эксперимента выберите оптимальное значение  $T_{\text{ож}}$  по критерию  $E$ .

### **Этап 3. Исследование влияния окна передачи на характеристики информационного канала при работе в режиме с групповым переспросом.**

Проведите эксперименты для системы с ранее выбранными оптимальными значениями  $I_{\max}$  и  $T_{\text{ож}}$ . Переменным параметром в этой серии экспериментов является значение окна передачи  $W$ , которое нужно увеличивать, начиная со значения равного 2 с шагом 1, с отслеживанием изменения критерия  $E$ . Эксперимент можно завершить после прохождения максимального значения относительной эффективной скорости  $E$ .

Постройте зависимости  $\bar{T}_{\Pi} = f(W)$ ,  $\bar{T}_c = f(W)$ ,  $E = f(W)$ .

По результатам эксперимента выберите оптимальное значение  $W$  по критерию  $E$ .

### Содержание отчета

1. Результаты расчетов по калибровке модели, определению величин задержки распространения сигнала, тайм-аута ожидания подтверждения и интервала моделирования системы в единицах модельного времени.

2. Полученные на этапах 1–3 зависимости характеристик информационного канала от системных параметров.

3. Выводы по результатам моделирования с обоснованием выбранных оптимальных значений системных параметров.

**Таблица 8.1**

### Исходные данные для проведения экспериментов

№ варианта	$V_m$ , бит/с	$L_k$ , км	$N_c$ , бит	$P_{\text{ош}}$	$T_{\text{мод}}$ , мин
1	2400	2000	1000	1E-4	2,5
2	4800	5000	2500	1E-3	2,0
3	1200	4000	900	5E-3	3,0
4	2400	7000	650	5E-4	2,0
5	9600	9000	1200	3E-3	2,5
6	1200	4000	2000	2E-4	2,0

### Фрагмент моделирующей программы

- \* GPSS/PC Program File L1x.GPS. (V 2, # 37349)
- \*     MODEL OF INFORMATION LINK
- \*     Data transmission system with feedback
- \*     Messages transfer with segmentation on standard packet
- \*     Frame failures take place because of errors
- \*         in forward and backward data link
- \*     X1 - message length;
- \*     X2 - standard segment (information packet) length
- \*     X3 - control packet length
- \*     X4 - probability of error per bit ( x 1E-6 )
- \*     X5 - propagation delay in channel
- \*     X6 - time-out of frame answerback
- \*     X7 - time of modeling
- \*     X8 - number of transmission frame
- \*     X9 - counter of transmission messages
- \*     X10 - counter of receive segment on destination point
- \*     X11 - counter of received data
- \*     X12 - number of next frame on destination point
- \*     X13 - counter of prepered segment
- \*     X14 - counter of received date in cycle
- \*     P1 - storage of start time for transmission message
- \*     P2 - number of segment in message

*	P3	- total number of segment in message
*	P4	- packet length
*	P5	- storage of start time for transmission frame
*	P6	- error flag (0 - error-free, 1 - error frame)
*	P7	- tipe of control frame (1 - RR, 0 - REJ)
*	P8	- frame number
*	P9	- frame transmission flag (0 - no, 1 - yes)
20	INITIAL	X1,2000; message length
30	INITIAL	X2,512; standard packet length
40	INITIAL	X3,48; length of control packet
50	INITIAL	X4, 1000; probability of error
60	INITIAL	X5,800; propagation delay
70	INITIAL	X6,1500; time-out of answerback
75	INITIAL	X7,100000; time of modeling
80	INITIAL	X8,0
85	INITIAL	X9,0
90	INITIAL	X10,1
93	INITIAL	X11,0
94	INITIAL	X12,1
95	INITIAL	X13,0
96	INITIAL	X14,0
100	BUF1	STORAGE 3; buffer for window
110	BUF2	STORAGE 1
120	LKF	VARIABLEX1@X2; length of non-standard packet
130	LKI	VARIABLEX1/X2; number of standard packet
140	LKS	VARIABLEX1/X2+1; number of packet
150	LKY	VARIABLEP3-1; number of packet - 1
160	LKX	VARIABLEX1/X2-1; number of copy
170	TKC1	VARIABLEP4+48; information frame length
180	TKC2	VARIABLEX3+48; control frame length
190	P_ER1	VARIABLEV\$TKC1#X4/1000; probability of
200	P_ER2	VARIABLEV\$TKC2#X4/1000; error per frame
211	HOM_R	VARIABLEX12-1; acknowledge frame number
212	CYCLE	VARIABLE(X2+48)+X5+(X3+48)+X5; stand.cycle
213	R_CICLE	VARIABLEX14#100/V\$CYCLE; relative rate
220	TIME	TABLE MP5,0,300,20; frame delivery time
230	TIM_M	TABLE MP1,0,4000,8; message delivery time
240	RATE	TABLE V\$R_CYCLE,0,10,10; relative rate

## 8.2. Исследование шинной ЛВС с методом доступа МДКН/ОК

**Цель работы:** исследование характеристик шинной локальной вычислительной сети (ЛВС), использующей множественный доступ с контролем несущей и обнаружением конфликтов (МДКН/ОК). Для исследования используется имитационная модель ЛВС.

## **Общие сведения**

Шинные ЛВС наряду с кольцевыми являются одними из самых распространенных типов локальных сетей. В шинных ЛВС применяются различные методы доступа: простая АЛОНА, тактированная АЛОНА, непрерывный множественный доступ с контролем несущей (МДКН), ненастойчивый МДКН, тактированный МДКН, МДКН с обнаружением конфликтов (МДКН/ОК).

Метод МДКН/ОК является наиболее совершенным и применяется в широко распространенных ЛВС типа Ethernet.

При методе МДКН/ОК каждый из абонентов прослушивает канал (шину) до того, как приступит к передаче. После освобождения канала абоненты могут приступить к передаче своего пакета. Из-за разницы в задержке распространения (наличия «окна конфликтов») два и более абонентов могут начать передачу в одно время, что приводит к наложению пакетов в шине и их искажению. При МДКН/ОК вводится прослушивание шины как до начала передачи (контроль несущей), так и во время передачи (обнаружение наложения). Каждый из отправителей, обнаружив наложение, сразу же прекращает передачу. При этом потерянное на конфликт время будет сравнительно небольшим по сравнению с общим временем передачи пакета.

После прекращения передачи конфликтующие абоненты повторяют попытку передачи своих пакетов через случайным образом сформированные тайм-ауты (для минимизации возможности нового конфликта). Если при повторной попытке снова возникает конфликт, то снова выбирается случайный период ожидания, но большего размера и т. д. После определенного числа попыток передать пакет устройство прекращает передачу и сообщает пользователю о невозможности передачи.

Наиболее распространенным является экспоненциальный алгоритм отсрочки передачи. Время задержки определяется в условных единицах, равных 51,2 мкс (эта величина больше типичной круговой задержки в шине ЛВС). Диапазон генерируемых случайных чисел в течение первых 10 попыток изменяется экспоненциально от (0,1) до (0,1023). С 11-й по 16-ю попытку диапазон остается неизменным (0,1023). Если 16-я попытка заканчивается неудачно, то каналный уровень отказывается от передачи пакета и оповещает об этом верхний протокольный уровень. Обычно это связано с нарушением кабеля.

## **Описание работы**

В данной работе используется имитационная модель шинной ЛВС с методом МДКН/ОК с изменяющимся числом (N) абонентских станций.

Модель имитирует поступление пакетов в соответствии с заданными законом и средней интенсивностью от абонентов для передачи их по ЛВС.

Модель позволяет собирать статистические данные о буферных накопителях как отдельных абонентских станций (АС), так и о суммарном объеме накопившейся не обслуженной нагрузки.

На модели могут быть получены характеристики загрузки шины ЛВС. Определяются также временные характеристики процесса доставки пакетов по ЛВС.

Входными переменными модели являются:

- число абонентских станций ЛВС;
- среднее время между моментами поступления пакетов от абонента (в модели принят экспоненциальный закон для потока входящей нагрузки);
- среднее время передачи пакета по шине (предполагается, что длительность передачи распределена экспоненциально);
- длительность интервала конфликтов в шине;
- время моделирования.

### Порядок выполнения работы

В таблице исходных данных приведены основные параметры исследуемой кольцевой ЛВС. Исходные данные вводятся в модель в интерактивном режиме в начале прогона.

Производится серия имитационных экспериментов с целью получения основных характеристик ЛВС при различном числе подключенных АС. Количество абонентов меняется в пределах 3–50.

### Содержание отчета

Отчет по лабораторной работе должен содержать:

1. Основные параметры ЛВС при каждом из экспериментов:

- загрузка шины ЛВС (см. параметр AVE.C. в статистике STORAGE BUS);
- среднее время доставки пакета (см. параметр MEAN в статистике TABLE TRAC);
- средняя длина очереди у абонента (см. параметр X9/1000);
- среднее число попыток на одну успешную передачу пакета (см. параметр X8/1000);
- число отказов в передаче пакетов (см. параметр X3).

2. Зависимости времени доставки пакета ( $T_{\text{дост}}$ ), коэффициента загрузки шины ( $\rho$ ), среднего объема данных в буферном накопителе абонентской станции ( $\bar{V}_{\text{AC}}$ ) и числа попыток передачи ( $K_{\text{поп}}$ ) от текущего числа АС в ЛВС ( $N_{\text{AC}}$ ), то есть:  $T_{\text{дост}} = f(N_{\text{AC}})$ ;  $\rho = f(N_{\text{AC}})$ ;  $\bar{V}_{\text{AC}} = f(N_{\text{AC}})$ ;  $K_{\text{поп}} = f(N_{\text{AC}})$ .

3. Выводы по результатам моделирования.

Таблица 8.2

### Исходные данные для проведения экспериментов

№ варианта	Средняя интенсивность	Интервал конфликтов	Среднее время передачи пакета	$T_{\text{мод}}$ мин
1	2000	2	14	0,5
2	3000	3	20	0,7
3	2500	2	10	0,6
4	1500	2	25	0,5
5	5000	3	18	0,8
6	4000	2	21	0,7

### 8.3. Исследование кольцевой локальной вычислительной сети

**Цель работы:** исследование характеристик кольцевой локальной вычислительной сети с маркерным методом передачи информации. Для исследования используется имитационная модель ЛВС.

#### Общие сведения

Кольцевые ЛВС являются одним из самых распространенных типов локальных сетей. В кольцевых ЛВС применяются три различных метода доступа: метод вставки регистра, метод тактируемого доступа и передача маркера.

При методе передачи маркера используется специальная последовательность символов, передаваемых по кольцу, – маркер (см. разд. 3.5, рис. 3.9).

В случае необходимости передачи данных АС ожидает прихода по кольцу к ней маркера. Получив маркер, АС удаляет его из кольца и посылает данные в кольцо. Затем АС ждет поступления обратно своего пакета, удаляет его из кольца и отправляет маркер следующему устройству в кольце. Получившая маркер станция может при необходимости отправить свой пакет по сети и т.д.

#### Описание работы

В данной работе используется имитационная модель локальной кольцевой маркерной сети с переменным числом ( $N$ ) абонентских станций.

Модель имитирует поступление сообщений в соответствии с заданными законом и средней интенсивностью от абонентов для передачи их по ЛВС. Поступившие от абонентов сообщения могут иметь различную длину и, следовательно, формируются как некоторое количество подготовленных для передачи по сети пакетов. Станция, имеющая подготовленные пакеты, ждет возможности занятия канала ЛВС – поступления маркера. Получив маркер, станция отправляет очередной пакет, дожидается его прихода обратно по кольцу, проверяет правильность передачи, и отсылает маркер следующей станции кольца.

Модель позволяет собирать статистические данные о буферных накопителях как отдельных АС, так и о суммарном объеме накопившейся не обслуженной нагрузки. На модели могут быть получены характеристики загрузки канала ЛВС. Определяются также временные характеристики процесса доставки пакетов по ЛВС.

Входными переменными модели являются:

- число абонентских станций ЛВС;
- среднее время между моментами поступления сообщений от абонента;
- закон распределения моментов поступления сообщений от абонентов;
- закон распределения длин поступающих сообщений (в пакетах);
- длительность цикла передачи пакета по ЛВС;
- время моделирования.

#### Порядок выполнения работы

В таблице исходных данных приведены основные параметры исследуемой кольцевой ЛВС.

Исходный закон распределения количества пакетов для категорий сообщений меняется первоначально в тексте моделирующей программы (функция 110 РАК).



Остальные исходные данные вводятся в модель в интерактивном режиме в начале прогона.

Производится серия имитационных экспериментов с целью получения основных характеристик ЛВС при различном числе подключенных АС. Количество абонентов меняется в пределах 3–50.

### Содержание отчета

Отчет по работе должен содержать следующие сведения.

1. Основные параметры ЛВС при каждом из экспериментов: загрузка канала ЛВС, среднее и максимальное число ожидающих отправки пакетов у абонентов, среднее время доставки пакета адресату по сети и его распределение.

2. Зависимости времени доставки пакета ( $T_{\text{дост}}$ ), коэффициента загрузки канала ( $\rho$ ), среднего и максимального объема данных в буферном накопителе абонентской станции ( $\bar{V}_{\text{AC}}$  и  $V_{\text{AC}}^{\text{max}}$ ) от текущего числа АС в ЛВС ( $N_{\text{AC}}$ ), то есть:

$$T_{\text{дост}} = f(N_{\text{AC}}); \rho = f(N_{\text{AC}}); \bar{V}_{\text{AC}} = f(N_{\text{AC}}); V_{\text{AC}}^{\text{max}} = f(N_{\text{AC}}).$$

3. Выводы по результатам моделирования.

Исходные данные по вариантам приведены в таблице 8.3.

**Таблица 8.3**

### Исходные данные для проведения экспериментов

№ варианта	Средняя интенсивность	Категории сообщений	Время передачи, мс	$T_{\text{мод}}$ , мин
1	2000	1 – 20% 2 – 60% 3 – 20%	14	1
2	3000	1 – 10% 2 – 50% 3 – 40%	20	0,8
3	2500	1 – 30% 2 – 60% 3 – 10%	10	1,2
4	1500	1 – 20% 2 – 30% 3 – 50%	25	1,3
5	5000	1 – 40% 2 – 50% 3 – 10%	18	1,1
6	4000	1 – 20% 2 – 40% 3 – 40%	21	0,9

## 8.4. Исследование транспортного соединения в глобальной сети

**Цель работы:** исследование характеристик транспортного соединения в широкомасштабной сети обмена информацией. Оценивается влияние на характеристики транспортного канала параметров магистральной и абонентских под-

сетей передачи данных. Для исследования используется имитационная модель транспортного соединения.

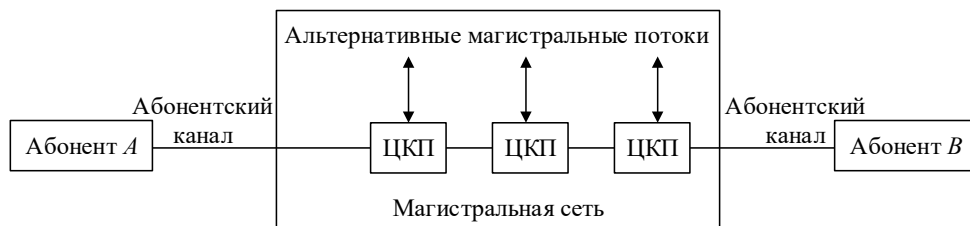
### Общие сведения

Основой современных информационных сетей является транспортная сеть, которая обеспечивает базу для разнообразных услуг – прикладных подсистем данной сети. С точки зрения концепции взаимосвязи открытых систем транспортная сеть реализует четыре нижних уровня семиуровневой иерархии протоколов – физический, управления информационным каналом, сетевой и транспортный. Само же транспортное соединение служит поставщиком сервиса для протоколов сеансового уровня, уровня представления и прикладного.

Работа транспортного протокола строится с учетом наиболее полного использования услуг двух нижележащих протокольных уровней – канального и сетевого. Протокол управления информационным каналом (наиболее широко используются протоколы HDLC – см. лабораторную работу 1) обеспечивает доставку данных по двухточечным участкам сети. С помощью различных процедур восстановления данных этот протокол гарантирует надежность передачи пакетов данных, помещенных в кадры. Протоколы сетевого уровня (типа X.25/3) отвечают за установление и поддержание сквозного виртуального канала между взаимодействующими абонентами с передачей по этому каналу пакетов данных. Транспортный протокол обеспечивает сервис передачи фрагментов сообщений, может содержать дополнительные процедуры для повышения надежности доставки, включает альтернативные режимы связи – например, срочную доставку данных.

### Описание работы

В данной лабораторной работе используется имитационная модель транспортного соединения двух абонентов через абонентские подсети и магистральную сеть связи. Моделируется работа фрагмента информационной сети с коммутацией пакетов (рис. 8.2).



**Рис. 8.2. Фрагмент моделируемой информационной сети**

Модель имитирует поступление от абонента *А* фрагментов сообщений, распределенных во времени в соответствии с экспоненциальным законом и заданной средней интенсивностью для передачи их по транспортному соединению абоненту *В*.

Модель позволяет изменять характеристики входного потока фрагментов, абонентских каналов связи (КС), магистральных каналов связи, имитировать степень загрузки магистральной сети путем изменения интенсивности входя-

щих магистральных потоков в центры коммутации пакетов (ЦКП), исследовать режимы нормальной и срочной доставки по транспортному каналу.

На модели могут быть получены характеристики загрузки отдельных каналов связи и ЦКП, временные параметры доставки фрагментов, сведения о входных и выходных очередях в центрах сети.

Входными переменными модели являются:

- средний интервал времени между моментами поступления фрагментов сообщений ( $\bar{T}_{\text{ФС}}$ );
- средний интервал времени между моментами поступления пакетов в центры коммутации пакетов магистральной сети ( $\bar{T}_{\text{ЦКП}}$ );
- характеристика качества использованных каналов связи в виде процента повторных передач пакета в канале при его передаче;
- скорость передачи в абонентских каналах связи ( $V_{\text{АК}}$ );
- скорость передачи в магистральных каналах связи ( $V_{\text{МК}}$ );
- длина сегмента сообщения, передаваемого от абонента;
- режим передачи данных от абонента – нормальный или срочный;
- время моделирования.

### **Порядок выполнения работы**

В таблице исходных данных приведены основные параметры исследуемого транспортного соединения. Исходные данные вводятся в модель в интерактивном режиме в начале прогона.

Производится следующие серии имитационных экспериментов с целью получения основных характеристик транспортного соединения.

1. Исследование зависимости характеристик доставки сообщений – времени доставки ( $\bar{T}_{\text{дост}}$ ) и максимальной длины очереди ( $Q_{\text{max}}$ ) от интенсивности входного потока информации у абонента. Получаемые зависимости:  $\bar{T}_{\text{дост}} = f(\bar{T}_{\text{ФС}})$ ,  $Q_{\text{max}} = f(\bar{T}_{\text{ФС}})$ .

$Q_{\text{max}}$  находится в файле статистики для очереди QUEUE BUF1A.

2. Сравнительные характеристики режимов нормальной и срочной доставки по транспортному каналу. Проводится эксперимент для исходных данных предыдущего прогона модели с близким к максимальному значением  $\bar{T}_{\text{дост}}$ .

3. Исследование зависимости характеристик доставки от скорости в абонентских КС.

Проводится серия экспериментов для одного из значений интенсивности входного потока сегментов с изменением скорости в абонентских каналах ( $V_{\text{АК}}$ ) по фиксированному ряду, 1200; 2400; 4800; 9200; 14400; 19200; 28800 бит/с.

Получаемые зависимости:

$$\bar{T}_{\text{дост}} = f(V_{\text{АК}}), Q_{\text{max}} = f(V_{\text{АК}})$$

4. Исследование зависимости характеристик доставки от скорости в магистральных КС.

Проводится серия экспериментов для одного из фиксированных значений скорости в абонентских КС с изменением скорости в магистральной сети ( $V_{\text{МК}}$ ).

Для экспериментов выбирается несколько значений из фиксированного ряда: 32; 64; 128; 256; 512 Кбит/с.

Получаемые зависимости:

$$\bar{T}_{\text{дост}} = f(V_{\text{МК}}), \Sigma Q_{\text{пих}} = f(V_{\text{МК}})$$

Величина  $\Sigma Q_{\text{пих}}$  вычисляется по файлу статистики по результатам эксперимента как сумма максимальных длин очередей:

(QUEUE) BUF1I, BUF1O, BUF2I, BUF2O, BUF3I, BUF3O.

5. Исследование зависимости характеристик доставки сообщений от загрузки магистральной сети.

Проводится серия экспериментов для одного из фиксированных значений скорости в абонентских КС с изменением «среднего интервала для нагрузки в ЦКП».

Получаемые зависимости:

$$\bar{T}_{\text{дост}} = f(\bar{T}_{\text{ЦКП}}), \Sigma Q_{\text{пих}} = f(\bar{T}_{\text{ЦКП}}).$$

### Содержание отчета

1. Основные зависимости, полученные в каждом из экспериментов.
2. Выводы по результатам моделирования.

### Исходные данные для проведения экспериментов

Исходные данные по вариантам приведены в таблице 8.4.

Таблица 8.4

Исходные данные по вариантам

№ варианта	Скорость в магистральном КС	Ср. интервал для нагрузки в ЦКП	Процент повторных передач в КС
1	64000	7	5
2	64000	8	10
3	128000	4	5
4	128000	5	10
5	64000	8	15

## 8.5. Сетевые утилиты

**Цель работы:** Определение настроек для подключения к локальной сети и к сети Internet с использованием утилиты **ipconfig**. Исследование вероятностно-временных характеристик фрагментов сети Internet с использованием утилиты **ping**. Исследование топологии фрагментов сети инетнет с использованием утилиты **tracert**.

### Общие сведения

Каждый компьютер в сети TCP/IP имеет адреса трех уровней.

1. Локальный адрес узла, определяемый технологией, с помощью которой построена отдельная сеть, в которую входит данный узел. Для узлов, входящих в локальные сети, это MAC-адрес сетевого адаптера или порта маршрутизатора, например, 11-A0-17-3D-BC-01. Эти адреса назначаются производителями оборудования и являются уникальными адресами, так как управляются централи-

зовано. Для всех существующих технологий локальных сетей MAC-адрес имеет формат 6 байтов: старшие 3 байта – идентификатор фирмы производителя, а младшие 3 байта назначаются уникальным образом самим производителем.

2. IP-адрес, состоящий из 4 байт, например, 109.26.17.100. Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации специального подразделения Internet (Network Information Center, NIC), если сеть должна работать как составная часть Internet. Обычно провайдеры услуг Internet получают диапазоны адресов у подразделений NIC, а затем распределяют их между своими абонентами. Номер узла в протоколе IP назначается независимо от локального адреса узла. Деление IP-адреса на поле номера сети и номера узла – гибкое, и граница между этими полями может устанавливаться весьма произвольно. Узел может входить в несколько IP-сетей. В этом случае узел должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

3. Символьный идентификатор-имя, например, SERV1.IBM.COM. Этот адрес назначается администратором и состоит из нескольких частей, например, имени машины, имени организации, имени домена. Такой адрес, называемый также DNS-именем, используется на прикладном уровне, например, в протоколах FTP или telnet.

Адреса всех трех уровней рассмотрим на примере общедоступных системных утилит сетевой диагностики.

**Утилита ipconfig** (IP configuration) предназначена для настройки протокола IP для операционной системы Windows. В данной лабораторной работе эта утилита используется только для получения информации о соединении по локальной сети. Для получения этой информации выполните «Пуск» → «Выполнить» → cmd и в командной строке введите:

```
ipconfig /all
```

В разделе «Адаптер Ethernet Подключение по локальной сети» для данной лабораторной будут необходимы поля «DHCP», «IP-адрес» и «DNS-серверы».

**Утилита ping** (Packet Internet Groper) является одним из главных средств, используемых для отладки сетей, и служит для принудительного вызова ответа конкретной машины. Она позволяет проверять работу программ TCP/IP на удаленных машинах, адреса устройств в локальной сети, адрес и маршрут для удаленного сетевого устройства. В выполнении команды ping участвуют система маршрутизации, схемы разрешения адресов и сетевые шлюзы. Это утилита низкого уровня, которая не требует наличия серверных процессов на проверяемой машине, поэтому успешный результат при прохождении запроса вовсе не означает, что выполняются какие-либо сервисные программы высокого уровня, а говорит о том, что сеть находится в рабочем состоянии, питание проверяемой машины включено, и машина не отказала (не «висит»).

В Windows утилита ping имеется в комплекте поставки и представляет собой программу, запускаемую из командной строки.

Запросы утилиты ping передаются по протоколу ICMP (Internet Control Message Protocol). Получив такой запрос, программное обеспечение, реализующее протокол IP у адресата, посылает эхо-ответ. Если проверяемая машина в момент получения запроса была загружена более приоритетной работой (например, обработкой и перенаправлением большого объема трафика), то ответ будет отправлен не сразу, а как только закончится выполнение более приоритетной задачи. Поэтому следует учесть, что задержка, рассчитанная утилитой ping, вызвана не только пропускной способностью канала передачи данных до проверяемой машины, но и загруженностью этой машины.

Эхо-запросы посылаются заданное количество раз (ключ -n). По умолчанию передается четыре запроса, после чего выводятся статистические данные.

**Обратите внимание:** поскольку с утилиты ping начинается хакерская атака, некоторые серверы в целях безопасности могут не посылать эхо-ответы (например, [www.microsoft.com](http://www.microsoft.com)). Не ждите напрасно, введите команду прерывания (CTRL+C).

Формат команды: ping [-t] [-a] [-n] [-l] [-f] [-i TTL] [-v TOS] [-r] [] [имя машины] [[-j списокУзлов] | [-k списокУзлов]] [-w]

Параметры утилиты ping приведены в таблице 8.5.

Таблица 8.5

Параметры утилиты ping

Ключи	Функции
-t	Отправка пакетов на указанный узел до команды прерывания
-a	Определение имени узла по IP-адресу
-n	Число отправляемых запросов
-l	Размер буфера отправки
-f	Установка флага, запрещающего фрагментацию пакета
-i TTL	Задание времени жизни пакета (поле «Time To Live»)

На практике большинство опций в формате команды можно опустить, тогда в командной строке может быть: ping имя узла (для зацикливания вывода информации о соединении используется опция -t; для вывода информации n раз используется опция -n количество раз).

Пример: ping -n 20 peak.mountin.net

```
Обмен пакетами с peak.mountin.net [207.227.119.2] по 32 байт:
Превышен интервал ожидания для запроса.
Ответ от 207.227.119.2: число байт=32 время=734мс TTL=231
Ответ от 207.227.119.2: число байт=32 время=719мс TTL=231
Ответ от 207.227.119.2: число байт=32 время=688мс TTL=231
Ответ от 207.227.119.2: число байт=32 время=704мс TTL=231
Превышен интервал ожидания для запроса.
Ответ от 207.227.119.2: число байт=32 время=719мс TTL=231
Ответ от 207.227.119.2: число байт=32 время=1015мс TTL=231
```

Превышен интервал ожидания для запроса.

Ответ от 207.227.119.2: число байт=32 время=703мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=688мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=782мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=688мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=688мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=688мс TTL=231

Превышен интервал ожидания для запроса.

Ответ от 207.227.119.2: число байт=32 время=687мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=735мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=672мс TTL=231

Ответ от 207.227.119.2: число байт=32 время=704мс TTL=231

Статистика Ping для 207.227.119.2:

Пакетов: отправлено = 20, получено = 16, потеряно = 4 (20% потеря),

Приблизительное время передачи и приема:

наименьшее = 672мс, наибольшее = 1015мс, среднее = 580мс

Пример определения имени узла по IP-адресу

ping -a 194.67.57.26

Обмен пакетами с mail.ru [194.67.57.26] по 32 байт: ...

**Утилита tracert** позволяет выявлять последовательность маршрутизаторов, через которые проходит IP-пакет на пути к пункту своего назначения.

Формат команды: tracert имя\_машины

имя\_машины может быть именем узла или IP-адресом машины. Выходная информация представляет собой список машин, начиная с первого шлюза и заканчивая пунктом назначения.

Пример: tracert peak.mountin.net

Трассировка маршрута к peak.mountin.net [207.227.119.2]

с максимальным числом прыжков 30:

№	Пакет 1	Пакет 2	Пакет 3	DNS-имя узла и (или) его IP-адрес
1	<10 мс	<10 мс	<10 мс	SLAVE [192.168.0.1]
2	<10 мс	<10 мс	<10 мс	gw.b10.tpu.edu.ru [195.208.164.2]
3	<10 мс	<10 мс	<10 мс	195.208.177.62
4	<10 мс	<10 мс	<10 мс	news.runnet.tomsk.ru [195.208.160.4]
5	<10 мс	<10 мс	16 мс	ra.cctpu.tomsk.su [195.208.161.34]
6	781 ms	563 ms	562 ms	spb-2-gw.runnet.ru [194.85.33.9]
7	547 ms	594 ms	578 ms	spb-gw.runnet.ru [194.85.36.30]
8	937 ms	563 ms	562 ms	20.201.atm0-201.ru-gw.run.net [193.232.80.105]
9	1125 ms	563 ms	547 ms	fi-gw.nordu.net [193.10.252.41]
10	906 ms	1016 ms	578 ms	s-gw.nordu.net [193.10.68.41]
11	844 ms	828 ms	610 ms	dk-gw2.nordu.net [193.10.68.38]
12	578 ms	610 ms	578 ms	sl-gw10-cop-9-0.sprintlink.net [80.77.65.25]
13	610 ms	968 ms	594 ms	sl-bb20-cop-8-0.sprintlink.net [80.77.64.37]
14	641 ms	672 ms	656 ms	sl-bb21-msq-10-0.sprintlink.net [144.232.19.29]
15	671 ms	704 ms	687 ms	sl-bb21-nyc-10-3.sprintlink.net [144.232.9.106]
16	985 ms	703 ms	765 ms	sl-bb22-nyc-14-0.sprintlink.net [144.232.7.102]
17	719 ms	734 ms	688 ms	144.232.18.206
18	891 ms	703 ms	734 ms	p1-0.nycmnyl-nbr1.bbnplanet.net [4.24.8.161]

№	Пакет 1	Пакет 2	Пакет 3	DNS-имя узла и (или) его IP-адрес
19	719 ms	985 ms	703 ms	so-6-0-0.chcgil2-br2.bbnplanet.net [4.24.4.17]
20	688 ms	687 ms	703 ms	so-7-0-0.chcgil2-br1.bbnplanet.net [4.24.5.217]
21	719 ms	703 ms	672 ms	p1-0.chcgil2-cr9.bbnplanet.net [4.24.8.110]
22	687 ms	719 ms	687 ms	p2-0.nchicago2-cr2.bbnplanet.net [4.0.5.242]
23	781 ms	703 ms	672 ms	p8-0-0.nchicago2-core0.bbnplanet.net [4.0.6.2]
24	672 ms	703 ms	687 ms	fa0.wcnet.bbnplanet.net [207.112.240.102]
25	734 ms	687 ms	688 ms	core0-s1.rac.cyberlynk.net [209.100.155.22]
26	1188 ms	*	890 ms	peak.mountin.net [207.227.119.2]

Трассировка завершена.

Пакеты посылаются по три на каждый узел. Для каждого пакета на экране отображается величина интервала времени между отправкой пакета и получением ответа. Символ \* означает, что ответ на данный пакет не был получен. Если узел не отвечает, то при превышении интервала ожидания ответа выдается сообщение «Превышен интервал ожидания для запроса». Интервал ожидания ответа может быть изменен с помощью опции – w команды tracert.

Команда tracert работает путем установки поля времени жизни (числа переходов) исходящего пакета таким образом, чтобы это время истекало до достижения пакетом пункта назначения. Когда время жизни истечет, текущий шлюз отправит сообщение об ошибке на машину-источник. Каждое приращение поля времени жизни позволяет пакету пройти на один маршрутизатор дальше.

**Примечание:** для вывода информации в файл используйте символ перенаправления потока вывода «>». Данный символ справедлив и для утилит ping и tracert.

Пример: tracert 195.208.164.1 > tracert.txt

Отчет о трассировке маршрута до указанного узла будет помещен в файл tracert.txt.

**Сервис Whois.** При регистрации доменных имен второго уровня обязательным условием является предоставление верных сведений о владельце этого домена: для юридических лиц – название организации, для физических лиц – ФИО и паспортные данные. Также обязательным является предоставление контактной информации. Часть этой информации становится свободно доступной для любого пользователя сети Интернет через сервис Whois. Получить интересующую информацию о владельце домена можно через Whois-клиент, например, в Unix это консольная команда whois, в ОС Windows – приложение SmartWhois. Но проще всего отправить запрос можно через веб-форму онлайн-сервиса Whois, например через форму на странице <http://www.nic.ru/whois/>.

### Описание работы

1. С помощью утилиты ipconfig определить IP адрес и физический адрес основного сетевого интерфейса компьютера, IP адрес шлюза, IP адреса DNS-серверов и используется ли DHCP. Результаты представить в табличном виде.

2. Проверить состояние связи с любыми двумя узлами (работоспособными) в соответствии с вариантом задания. Число отправляемых запросов должно составлять не менее 20. В качестве результата отразить для каждого из исследуемых узлов в виде таблицы с полями:



- процент потерянных пакетов;
- среднее время приема-передачи;
- количество маршрутизаторов (с учетом шлюза) до опрашиваемого узла;

- IP адрес узла.

- класс сети, к которой принадлежит данный узел;

- имя узла, полученное по IP-адресу узла.

В отчете необходимо пояснить, как были определены значения.

3. Произвести трассировку двух работоспособных узлов в соответствии с вариантом задания. Результаты запротоколировать в таблице 8.6.

**Таблица 8.6**

№ узла	Время прохождения пакета №1	Время прохождения пакета №2	Время прохождения пакета №3	Среднее время прохождения пакета	DNS-имя маршрутизатора	IP-адрес маршрутизатора
--------	-----------------------------	-----------------------------	-----------------------------	----------------------------------	------------------------	-------------------------

*Если значения времени прохождения трех пакетов отличаются более чем на 10 мс либо есть потери пакетов, то для соответствующих узлов среднее время прохождения необходимо определять с помощью утилиты ping по 20 пакетам.*

По результатам таблицы в отчете привести график изменения среднего времени прохождения пакета. В отчёте привести одну копию окна с результатами команды tracert. Для каждого опрашиваемого узла определить участок сети между двумя соседними маршрутизаторами, который характеризуется наибольшей задержкой при пересылке пакетов. Для найденных маршрутизаторов с помощью сервиса Whois определить название организации и контактные данные (тел., e-mail). Полученную информацию необходимо указать в отчете.

### **Варианты заданий**

Исходные данные для выполнения работы приведены в таблице 8.7

**Таблица 8.7**

### **Исходные данные для выполнения работы**

№ варианта	Символьные адреса	№ варианта	Символьные адреса
1	www.informika.ru www.rfbr.ru www.ras.ru	11	www.tractor.ru www.rsci.ru www.astronet.ru
2	www.gpntb.ru www.rusmedserv.com www.nsc.ru	12	www.keldysh.ru www.fom.ru www.inauka.ru

Продолжение табл. 8.7

№ варианта	Символьные адреса	№ варианта	Символьные адреса
3	www.chemnet.ru www.rsl.ru www.philosophy.ru	13	www.gramota.ru www.csa.ru www.bionet.nsc.ru
4	www.rbc.ru www.membrana.ru www.osi.ru	14	www.inp.nsk.su www.scientific.ru www.med2000.ru
5	www.viniti.ru www.sostav.ru www.ioffe.ru	15	www.gpi.ru iki.cosmos.ru www.spsl.nsc.ru
6	www.fegi.ru www.elibrary.ru www.extech.ru	16	www.uiggm.nsc.ru hist.dcn-asu.ru www.cemi.rssi.ru
7	www.ripn.net www.shpl.ru sai.msu.su	17	psychology.net.ru www.irex.ru www.medlinks.ru
8	www.scsml.rssi.ru www.sccc.ru www.nlr.ru	18	www.viniti.ru www.sostav.ru www.gramota.ru
9	web.ru www.kamaz.ru www.rulex.ru	19	www.sccc.ru www.nlr.ru www.fom.ru
10	www.jinr.ru uic.nnov.ru www.ruthenia.ru	20	uic.nnov.ru www.ruthenia.ru www.rsl.ru

## 8.6. IP-адресация

**Цель работы:** Изучить эталонную модель протоколов ISO/OSI, стек протоколов TCP/IP и правила назначения IP-адресов.

### Общие сведения

**Формат IP-адреса.** IP-адрес представляет собой 32-разрядный номер, который уникально идентифицирует узел (компьютер или устройство, например, принтер или маршрутизатор) в сети TCP/IP.

IP-адреса обычно представлены в виде 4-х разрядов, разделенных точками, например 192.168.123.132. Чтобы понять использование масок подсетей для распознавания узлов, сетей и подсетей, обратите внимание на IP-адрес в двоичном обозначении.

Например, в виде разрядов, разделенных точками, IP-адрес 192.168.123.132 – это (в двоичном обозначении) 32-разрядный номер 110000000101000111101110000100. Такой номер сложно интерпретировать, поэтому разбейте его на четыре части по восемь двоичных знаков.

Эти 8-разрядные секции называются «октетты». Тогда данный IP-адрес будет иметь вид: 11000000.10101000.01111011.10000100. Этот номер ненамного понятнее, поэтому в большинстве случаев следует преобразовывать двоичный адрес в формат разделенных точками разрядов (192.168.123.132). Десятичные

числа, разделенные точками, и есть октеты, преобразованные из двоичного в десятичное обозначение.

Чтобы глобальная сеть TCP/IP работала эффективно как совокупность сетей, маршрутизаторы, обеспечивающие обмен пакетами данных между сетями, не знают точного расположения узла, для которого предназначен пакет. Маршрутизаторы знают только, к какой сети принадлежит узел, и используют сведения, хранящиеся в таблицах маршрутизации, чтобы доставить пакет в сеть узла назначения. Как только пакет доставлен в необходимую сеть, он доставляется в соответствующий узел.

Для осуществления этого процесса IP-адрес состоит из двух частей. Первая часть IP-адреса обозначает адрес сети, последняя часть – адрес узла. Если рассмотреть IP-адрес 192.168.123.132 и разбить его на эти две части, то получится следующее:

192.168.123. – сеть  
.132 – узел

или

192.168.123.0 – адрес сети;

0.0.0.132 – адрес узла.

Следующий элемент, необходимый для работы протокола TCP/IP, – это маска подсети. Протокол TCP/IP использует маску подсети, чтобы определить, в какой сети находится узел: в локальной подсети или удаленной сети.

В протоколе TCP/IP части IP-адреса, используемые в качестве адреса сети и узла, не зафиксированы, следовательно, указанные выше адреса сети и узла невозможно определить без наличия дополнительных сведений. Данные сведения можно получить из другого 32-разрядного номера под названием «маска подсети». В этом примере маской подсети является 255.255.255.0. Значение этого номера понятно, если знать, что число 255 в двоичном обозначении соответствует числу 11111111; таким образом, маской подсети является номер:

11111111.11111111.11111111.00000000

Расположив следующим образом IP-адрес и маску подсети, можно выделить составляющие сети и узла:

11000000.10101000.01111011.10000100 – IP-адрес (192.168.123.132)

11111111.11111111.11111111.00000000 – маска подсети (255.255.255.0).

Первые 24 разряда (число единиц в маске подсети) распознаются как адрес сети, а последние 8 разрядов (число оставшихся нулей в маске подсети) – адрес узла. Таким образом, получаем следующее:

11000000.10101000.01111011.00000000 – адрес сети (192.168.123.0)

00000000.00000000.00000000.10000100 – адрес узла (000.000.000.132)

Из данного примера с использованием маски подсети 255.255.255.0 видно, что код сети 192.168.123.0, а адрес узла 0.0.0.132. Когда пакет с конечным адресом 192.168.123.132 доставляется в сеть 192.168.123.0 (из локальной подсети или удаленной сети), компьютер получит его из сети и обработает.

Почти все десятичные маски подсети преобразовываются в двоичные числа, представленные единицами слева и нолями справа. Вот еще некоторые распространенные маски подсети:

Десятичные	Двоичные
255.255.255.192	1111111.11111111.1111111.11000000
255.255.255.224	1111111.11111111.1111111.11100000

Стандарт Internet RFC 1878 (доступен на <http://www.internic.net>) описывает действующие подсети и маски подсетей, используемые в сетевых протоколах TCP/IP.

**Классы сетей.** Интернет-адреса распределяются организацией InterNIC (<http://www.internic.net>), которая администрирует интернет. Эти IP-адреса распределены по классам. Наиболее распространены классы А, В и С. Классы D и E существуют, но обычно не используются конечными пользователями. Каждый из классов адресов имеет свою маску подсети по умолчанию. Определить класс IP-адреса можно по его первому октету. Ниже описаны интернет-адреса классов А, В и С с примером адреса для каждого класса.

- Сети класса А по умолчанию используют маску подсети 255.0.0.0 и имеют значения от 0 до 127 в первом октете. Адрес 10.52.36.11 является адресом класса А. Первым октетом является число 10, входящее в диапазон от 1 до 126 включительно.

- Сети класса В по умолчанию используют маску подсети 255.255.0.0 и имеют в первом октете значение от 128 до 191. Адрес 172.16.52.63 является адресом класса В. Первым октетом является число 172, входящее в диапазон от 128 до 191 включительно.

- Сети класса С по умолчанию используют маску подсети 255.255.255.0 и имеют в первом октете значение от 192 до 223. Адрес 192.168.123.132 является адресом класса С. В первом октете число 192, которое находится между 192 и 223 включительно.

В некоторых случаях значение маски подсети по умолчанию не соответствует потребностям организации из-за физической топологии сети или потому, что количество сетей (или узлов) не соответствует ограничениям маски подсети по умолчанию. В следующем разделе рассказывается, как можно распределить сети с помощью масок подсети.

**Подсети.** TCP/IP-сеть класса А, В или С может еще быть разбита на подсети системным администратором. Образование подсетей может быть необходимо при согласовании логической структуры адреса Интернета (абстрактный мир IP-адресов и подсетей) с физическими сетями, используемыми в реальном мире.

Системный администратор, выделивший блок IP-адресов, возможно, администрирует сети, организованные не соответствующим для них образом. Например, имеется глобальная сеть с 150 узлами в трех сетях (в разных городах), соединенных маршрутизатором TCP/IP. У каждой из этих трех сетей 50 узлов. Выделяем сеть класса С 192.168.123.0. (Для примера, на самом деле этот адрес из серии, не размещенной в Интернете.) Это значит, что адреса с 192.168.123.1 по 192.168.123.254 можно использовать для этих 150 узлов.

Два адреса, которые нельзя использовать в данном примере, – 192.168.123.0 и 192.168.123.255, так как двоичные адреса с составляющей узла из одних единиц и нулей недопустимы. Адрес с 0 недопустим, поскольку он

используется для определения сети без указания узла. Адрес с числом 255 (в двоичном обозначении адрес узла, состоящий из одних единиц) используется для доставки сообщения на каждый узел сети. Следует просто запомнить, что первый и последний адрес в любой сети и подсети не может быть присвоен отдельному узлу.

Теперь осталось дать IP-адреса 254 узлам. Это несложно, если все 150 компьютеров являются частью одной сети. Однако в данном примере 150 компьютеров работают в трех отдельных физических сетях. Вместо запроса на большее количество адресных блоков для каждой сети сеть разбивается на подсети, что позволяет использовать один блок адресов в нескольких физических сетях.

В данном случае сеть разбивается на четыре подсети с помощью маски подсети, которая увеличивает адрес сети и уменьшает возможный диапазон адресов узлов. Другими словами, мы «одалживаем» несколько разрядов, обычно используемых для адреса узла, и используем их для составляющей сети в адресе. Маска подсети 255.255.255.192 позволяет создать четыре сети с 62 узлами в каждой. Это возможно, поскольку в двоичном обозначении 255.255.255.192 – то же самое, что и 1111111.11111111.1111111.11000000.

Первые две цифры последнего октета становятся адресами сети, поэтому появляются дополнительные сети 00000000 (0), 01000000 (64), 10000000 (128) и 11000000 (192). (Некоторые администраторы применяют только две из этих подсетей, используя номер 255.255.255.192 в качестве маски подсети. Для получения дополнительной информации по этому вопросу см. RFC 1878.) В этих четырех сетях последние 6 двоичных цифр можно использовать в качестве адресов узлов.

Использование маски подсети 255.255.255.192 преобразует сеть 192.168.123.0 в четыре сети: 192.168.123.0, 192.168.123.64, 192.168.123.128 и 192.168.123.192. Эти четыре сети будут иметь следующие действующие адреса узлов:

- 192.168.123.1-62
- 192.168.123.65-126
- 192.168.123.129-190
- 192.168.123.193-254

Не забывайте, что двоичные адреса узлов с одними только единицами и нолями недействительны, поэтому нельзя использовать адреса со следующими числами в последнем октете: 0, 63, 64, 127, 128, 191, 192 или 255.

Обратите внимание на следующие два адреса узлов: 192.168.123.71 и 192.168.123.133. Если использовать по умолчанию маску подсети класса C 255.255.255.0, оба адреса будут в сети 192.168.123.0. Однако, если использовать маску подсети 255.255.255.192, они окажутся в разных сетях: 192.168.123.71 – в сети 192.168.123.64, в то время как 192.168.123.133 – в сети 192.168.123.128.

Описание работы

1. Заполните таблицу 8.8 «Характеристики сетей различных классов»

Таблица 8.8

**Характеристики сетей различных классов**

№ п.п	Характеристика сети	Класс сети		
		А	В	С
1	2	3	4	5
1	Формат первого байта IP-адреса			
2	Число байтов для номера сети			
3	Число байтов для номера хоста			
4	Минимальный номер сети в двоичном формате			
5	Минимальный номер сети в точечной нотации			
6	Минимальный номер сети в десятичном виде			
7	Максимальный номер сети в двоичном формате			
8	Максимальный номер сети в точечной нотации			
9	Максимальный номер сети в десятичном виде			
10	Число различных сетей			
11	Минимальный номер хоста в точечной нотации			
12	Максимальный номер хоста в двоичном формате			
13	Максимальный номер хоста в точечной нотации			
14	Максимальный номер хоста в десятичном виде			
15	Число различных хостов			
16	Маска сети по умолчанию			

2. Для IP-адреса, указанного в индивидуальном задании, считая, что маска сети задана по умолчанию, определить:

- класс сети;
- маску сети по умолчанию;
- номер сети;
- номер хоста;
- широковещательный адрес.

3. Указать, что означает запись, указанная в индивидуальном задании, и определить

- маску сети;
- номер сети;
- номер хоста;
- широковещательный адрес.

**Порядок выполнения работы**

На основе примера, разобранный для сетей класса А, заполните третью колонку таблицы 8.8. Выполните аналогичные расчеты и заполните четвертую и пятую колонки таблицы 8.8.

Для выполнения п. 2 задания необходимо выполнить следующие действия:

• Перевести каждое число IP-адреса в двоичную форму. Для перевода воспользуйтесь программой «Калькулятор», установив «Вид/Инженерный». Не забудьте, что в двоичном представлении каждого числа надо записывать ровно восемь битов!

- По первым битам IP-адреса определить класс сети.
- В соответствии с классом определить маску сети по умолчанию.

- Выписать только те биты IP-адреса, которые соответствуют единичным битам в маске сети. Представить эти биты в точечной нотации. Это будет номер сети.

- Выписать те биты IP-адреса, которые соответствуют нулевым битам в маске сети. Представить их в точечной нотации. Это будет номер хоста.

- В двоичном представлении IP-адреса биты, соответствующие номеру хоста, заменить единицами. Представить получившийся адрес в точечной нотации. Это будет широковещательный адрес.

Рассмотрим пример выполнения п. 2 задания.

Пусть IP-адрес 64.10.20.30. Переводим числа в двоичный формат:

$64_{10}=01000000_2$

$10_{10}=00001010_2$

$20_{10}=00010100_2$

$30_{10}=00011110_2$

Записываем двоичную форму представления IP-адреса:

01000000.00001010.00010100.00011110

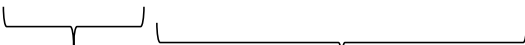
Первые биты адреса – 01, значит, это СЕТЬ КЛАССА А.

МАСКА СЕТИ ПО УМОЛЧАНИЮ: 255.0.0.0

Записываем в двоичной форме маску сети и IP-адрес:

Маска: 11111111. 00000000.00000000.00000000

IP-адрес: 01000000. 00001010.00010100.00011110

	
Эти биты	Эти биты
соответствуют	соответствуют
номеру сети	номеру хоста

Значит, НОМЕР СЕТИ –  $01000000_2$  или  $64_{10}$

НОМЕР ХОСТА –  $00001010.00010100.00011110_2$  или  $10.20.30_{10}$

Заменяем в IP-адресе номер хоста единицами, получим

ШИРОКОВЕЩАТЕЛЬНЫЙ АДРЕС

$01000000.11111111.11111111.11111111_2$  или 64.255.255.255

Следовательно:

IP-адрес 64.10.20.30

Класс сети А

Маска сети 255.0.0.0

Номер сети 64

Номер хоста 10.20.30

Широковещательный адрес 64.255.255.255

При выполнении п.3 задания необходимо вначале определить маску сети.

Маска содержит столько единичных битов, сколько указано в числе после дробной черты. Дальнейшие расчеты аналогичны выполнению п.2 задания.

### Содержание отчета

Отчет должен содержать заполненную таблицу характеристик сетей, и результаты выполнения заданий 2 и 3 с необходимыми пояснениями (по образцу).

Варианты индивидуальных заданий представлены в таблице 8.9.

Таблица 8.9

## Варианты индивидуальных заданий

Номер варианта	IP-адрес к заданию 2	IP-адрес к заданию 3
1	192.168.72.33	192.168.72.33/20
2	190.172.55.40	190.172.55.40/24
3	123.232.14.72	123.232.14.72/18
4	196.232.66.54	196.232.66.54/30
5	193.123.55.67	193.123.55.67/26
6	191.172.55.42	191.172.55.42/24
7	178.66.57.18	178.66.57.18/20
8	10.0.0.20	10.0.0.20/12
9	67.192.44.89	67.192.44.89/12
10	128.34.67.11	128.34.67.11/18
11	193.34.126.44	193.34.126.44/26
12	156.32.11.93	156.32.11.93/30
13	167.168.169.170	167.168.169.17/20
14	145.44.11.77	145.44.11.77/22
15	132.45.171.99	132.45.171.99/24
16	198.164.55.55	198.164.55.55/26
17	192.77.11.44	192.77.11.44/30
18	12.13.14.15	12.13.14.15/24
19	44.57.162.31	44.57.162.31/18
20	152.154.66.65	152.154.66.65/20

## 8.7. Аутентификация, авторизация и учет

**Цель работы:** ознакомление студентов с типовой структурой корпоративной сети и с процедурами аутентификации, авторизации и учета.

**Общие сведения**

Типовая структура корпоративной сети приведена на рис. 8.3. На ней изображены:

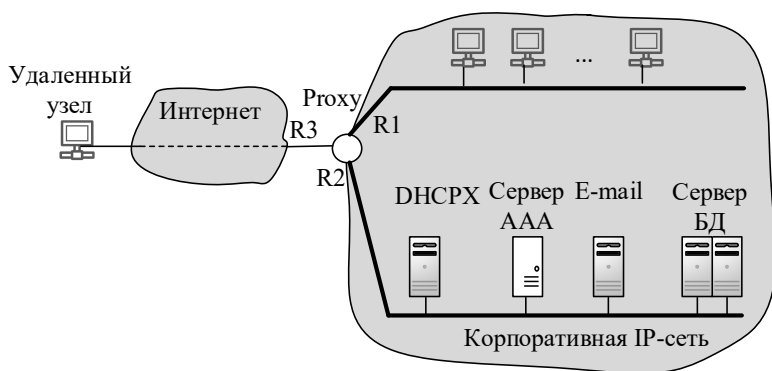
- вспомогательный (Proxu) сервер, основными функциями которого является коммутация трафика между интерфейсами Int-1, Int-2 и Int-3 в соответствии со списками доступа администратора;
- хосты локального и удаленного пользователя;
- серверы:
  - ДНСР для конфигурирования хостов,
  - ААА для аутентификации, авторизации и учета,
  - e-mail для обработки почтовых сообщений,
  - баз данных (БД) для хранения документации группового использования.

В корпоративной сети последовательно осуществляются три административные процедуры: аутентификация, авторизация и учет.

Аутентификация – установление легитимности абонента посредством запроса его имени и пароля. При попытке подключения пользователя к корпоративной сети проху-сервер запрашивает его имя и пароль. Полученный ответ



сравнивается с записью в списке доступа вида: имя пользователя (Name или User ID) – пароль (Password), которая внесена администратором сети и хранится на AAA-сервере. Аутентификация может осуществляться при помощи двух протоколов – Password Authentication Protocol (PAP) и Challenge Handshake Authentication Protocol (CHAP), являющимися составными частями протокола PPP.



**Рис. 8.3. Типовая схема корпоративной сети**

В протоколе PAP имя пользователя и пароль передаются в одном сообщении и в открытом виде. Они могут быть легко перехвачены злоумышленником, поэтому протокол PAP используется только для аутентификации локальных пользователей и не может быть использован для аутентификации удаленных пользователей.

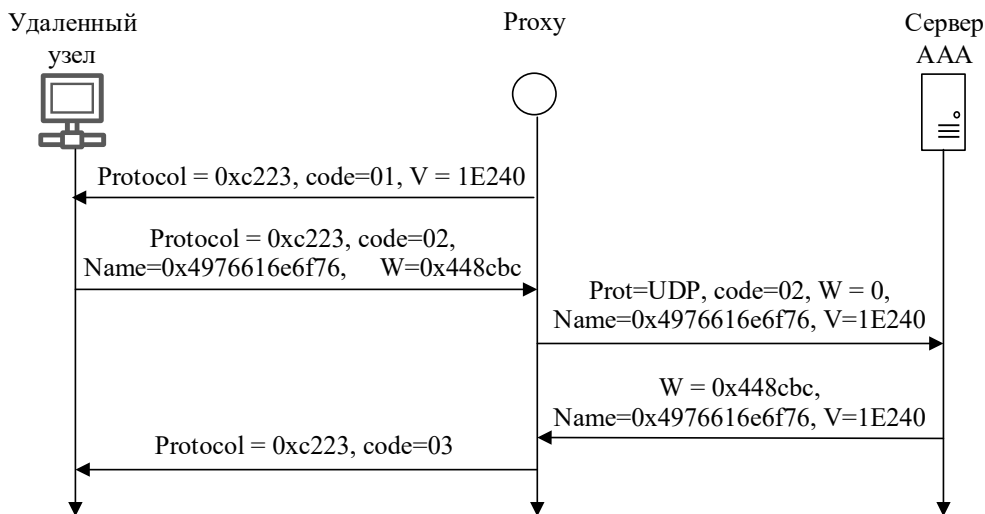
При использовании протокола CHAP прокxu-сервер посылает удаленному хосту пользователя некоторое случайное число  $V$ , а хост возвращает другое число  $W$ , вычисленное по заранее известной функции с использованием имени (Name) и пароля (Password). Иначе говоря,  $W = f(V, \text{Name}, \text{Password})$ . Предполагается, что злоумышленник в состоянии перехватить пересылаемые по сети значения  $V$ , Name и  $W$ , и ему известен алгоритм вычисления функции  $f$ . Существо формирования  $W$  состоит в том, что исходные элементы (биты) случайного числа  $V$  различным образом «перемешиваются» с неизвестным злоумышленнику элементами пароля Password. Затем полученный зашифрованный текст подвергается сжатию. Такое преобразование называется дайджест-функцией (digest function) или хэш-функцией, а результат – дайджестом. Точная процедура формирования дайджеста определена алгоритмом MD5 и описана в [RFC 1321, PS]. Прокxu-сервер запрашивает у AAA-сервера истинное значение  $W$ , пересылая ему значения Name и Challenge= $V$ . Сервер AAA на основании полученных от прокxu-сервера значений  $V$  и Name и имеющегося у него в базе данных пароля Password по тому же алгоритму вычисляет  $W$  и возвращает его прокxu-серверу. Прокxu-сервер сравнивает два значения  $W$ , полученные от хоста и от AAA-сервера: если они совпадают, то хосту посылается сообщение об успешной аутентификации.

После успешной аутентификации пользователя проху-сервер на основании списка управления доступом производит авторизацию, т.е. определяет, к каким серверам DB1 и DB2 группового использования может обращаться пользователь, а сервера DB1 и DB2 определяют, какие операции (только чтение или чтение/запись) он может осуществлять.

Процедура учета состоит в ведении записей истории соединений пользователей для последующего возможного анализа успешных и неуспешных соединений.

### Описание работы

На рис. 8.4 приведена процедура авторизации пользователя со следующими исходными данными: имя пользователя (Name) Ivanov, пароль (Password = K1m), случайное число (V) 123456. Процедура перемешивания состоит в последовательном перемешивании полубайтов пароля и случайного числа. Вычисление дайджеста состоит в вычислении остатка перемешенного числа по модулю Password.



**Рис. 8.4. Процедура аутентификации пользователя**

Рассмотрим вычисление полей протокола аутентификации подробно.

В первом сообщении проху-сервер запрашивает (code=01) по протоколу аутентификации CHAP (Protocol = 0xc223) у удаленного пользователя ответ на случайное число  $V = 123456 = 0x1E240$ . Хост удаленного пользователя производит следующие операции.

1. Подставляет имя пользователя, используя таблицу кодов ASCII (таблица 8.10).

Таблица 8.10

Таблица кодов ASCII

	(0) 000	(1) 001	(2) 010	(3) 011	(4) 100	(5) 101	(6) 110	(7) 111
(0) 0000	<b>NUL</b>	<b>DLE</b>	<b>SP</b>	<b>0</b>	<b>@</b>	<b>P</b>	<b>'</b>	<b>p</b>
(1) 0001	<b>SOH</b>	<b>DC1</b>	<b>!</b>	<b>1</b>	<b>A</b>	<b>Q</b>	<b>a</b>	<b>q</b>
(2) 0010	<b>STX</b>	<b>DC2</b>	<b>“</b>	<b>2</b>	<b>B</b>	<b>R</b>	<b>b</b>	<b>r</b>
(3) 0011	<b>ETX</b>	<b>DC3</b>	<b>#</b>	<b>3</b>	<b>C</b>	<b>S</b>	<b>c</b>	<b>s</b>
(4) 0100	<b>EOT</b>	<b>DC4</b>	<b>\$</b>	<b>4</b>	<b>D</b>	<b>T</b>	<b>d</b>	<b>t</b>
(5) 0101	<b>ENQ</b>	<b>NAK</b>	<b>%</b>	<b>5</b>	<b>E</b>	<b>U</b>	<b>e</b>	<b>u</b>
(6) 0110	<b>ACK</b>	<b>SYN</b>	<b>&amp;</b>	<b>6</b>	<b>F</b>	<b>V</b>	<b>f</b>	<b>v</b>
(7) 0111	<b>BEL</b>	<b>ETB</b>	<b>'</b>	<b>7</b>	<b>G</b>	<b>W</b>	<b>g</b>	<b>w</b>
(8) 1000	<b>BS</b>	<b>CAN</b>	<b>(</b>	<b>8</b>	<b>H</b>	<b>X</b>	<b>h</b>	<b>x</b>
(9) 1001	<b>HT</b>	<b>EM</b>	<b>)</b>	<b>9</b>	<b>I</b>	<b>Y</b>	<b>i</b>	<b>y</b>
(a) 1010	<b>LF</b>	<b>SUB</b>	<b>*</b>	<b>:</b>	<b>J</b>	<b>Z</b>	<b>j</b>	<b>z</b>
(b) 1011	<b>VT</b>	<b>ESC</b>	<b>+</b>	<b>;</b>	<b>K</b>	<b>[</b>	<b>k</b>	<b>{</b>
(c) 1100	<b>FF</b>	<b>IS4</b>	<b>,</b>	<b>&lt;</b>	<b>L</b>	<b>\</b>	<b>l</b>	<b> </b>
(d) 1101	<b>CR</b>	<b>IS3</b>	<b>-</b>	<b>=</b>	<b>M</b>	<b>]</b>	<b>m</b>	<b>}</b>
(e) 1110	<b>SO</b>	<b>IS2</b>	<b>.</b>	<b>&gt;</b>	<b>N</b>	<b>^</b>	<b>n</b>	<b>~</b>
(f) 1111	<b>S1</b>	<b>IS1</b>	<b>/</b>	<b>?</b>	<b>O</b>	<b>_</b>	<b>o</b>	<b>DEL</b>

Для определения двоичного кода символа следует к коду колонки приписать код строки, а для определения шестнадцатеричного – к значению кода колонки приписать значение кода строки. В соответствии с табл. 8.10 имя пользователя Ivanov представляется как 0x4976616e6f76, а пароль K1m – как 0x4b316d.

2. Перемешивает байты пароля 0x**4b316d** и случайного числа 0x01e240, получая перемешанное число F=0x**40b13e1264d0**.

3. Вычисляет ответ как  $W = F \bmod \text{Password} = 40b13e1264d0 \bmod 0x4b316d = 71129994781904 \bmod 4927853 = 4493476 = 0x448cbc$ .

Во втором сообщении хост возвращает ответ в виде Name=0x4976616e6f76 и W = 0x448cbc.

В третьем сообщении проху-сервер запрашивает истинное значение W у AAA-сервера, посылая ему те же значения Name и V.

В четвертом сообщении проху-сервер получает от AAA-сервера истинное значение W, соответствующее Name=0x4976616e6f76 и V=0x1E240.

В пятом сообщении проху-сервер подтверждает (code=03) легитимность пользователя.

### Порядок выполнения

1. В виде рис. 8.4 представить процедуру аутентификации при следующих исходных данных:

Имя пользователя (Name) – фамилия студента,

Пароль (Password) - Y1Y2,

Случайное число (V=Challenge) - Y3Y4.

2. Ответить на контрольный вопрос.

### Варианты для выполнения работы

Варианты для выполнения работы приведены в таблице 8.11.

Таблица 8.11

Варианты для выполнения работы

№ варианта	Y1	Y2	Y3	Y4
1	13	14	18	15
2	23	25	30	26
3	33	36	42	38
4	43	47	50	45
5	53	58	61	55
6	63	69	72	65
7	73	74	75	77
8	83	84	88	92
9	93	95	99	97
10	13	16	21	18
11	23	26	32	30
12	33	38	41	40

## 8.8. Маршрутизация в IP-сетях

**Цель работы:** Ознакомление с правилами маршрутизации в IP-сетях.

### Общие сведения

Маршрутизация пакета в публичной сети производится на основании классического IP-адреса номера сети, согласно табл. 4.2, приведенной в разделе 4.2.3 учебника.

Номер сети принято обозначать с помощью маски, количество лидирующих «единиц» в маске показывает число старших разрядов, которые определяют номер сети.

Запись маски производится в формате IP-адреса. Таким образом, для сети класса А стандартная маска имеет вид 255.0.0.0 (в двоичном коде 11111111.00000000.00000000.00000000), для сети класса В – 255.255.0.0 (11111111.11111111.00000000.00000000), для сети класса С – 255.255.255.0 (11111111.11111111.11111111.00000000).

Наличие только четырех классов адресов часто бывает неудобно. Если администратору необходимо создать сеть из определенного количества узлов, то эта проблема решается с помощью создания подсетей, путем переназначения части битов узла в качестве битов сети.

Для выполнения этой работы необходимо ознакомиться с материалом раздела 4.2.2.3 учебника.

### Описание работы

Пусть из публичной сети (рис. 8.5) поступили следующие пакеты с IP-адресами назначения: 135.38.16.15, 135.38.56.211, 135.38.92.10.

Определим, на какие интерфейсы они будут направлены.

Для определения номера адресуемой подсети складываем по правилу (4.1)

IP-адрес назначения первого пакета складываем с маской сети, получаем

IP = 135.38.16.15 → 10000111. 00100110. 00010000. 00001111

Mask = 255.255.224.0 → 11111111. 11111111. 11100000. 00000000

---

Destination = 10000111. 00100110. 00000000. 00000000<sub>2</sub> → 135. 38.0.0.

Такой записи в таблице маршрутизации нет, пакет уничтожается.

Для второго пакета

IP=135.38.56.211 → 10000111. 00100110. 00111000. 11010011

Mask= 255.255.224.0 → 11111111. 11111111. 11100000. 00000000

-----  
Destination = 10000111. 00100110. 00100000. 00000000<sub>2</sub> → 135.38.32.0.

Пакет будет направлен на интерфейс S1.

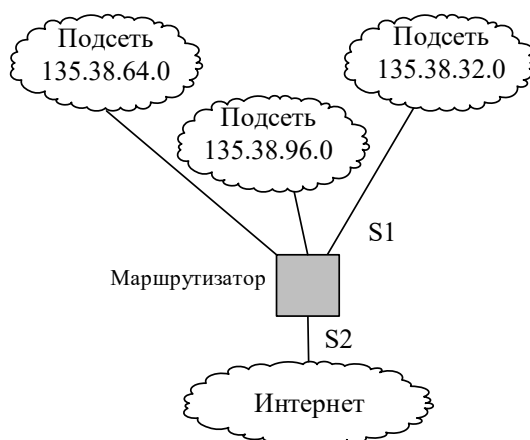
Для третьего пакета

IP=135.38.92.10 → 10000111. 00100110. 01011100. 00001010

Mask=255.255.224.0 → 11111111. 11111111. 11100000. 00000000

-----  
Destination = 10000111. 00100110. 01000000. 00000000<sub>2</sub> → 135.38.64.0.

Такой записи в таблице маршрутизации нет, пакет уничтожается.



**Рис. 8.5. Архитектура местоположения подсети 135.38.32.0**

### **Порядок выполнения работы**

Имеется сеть 131.40.0.0. Администратор разбил ее на 6 подсетей по 8100 узлов в каждой.

Изобразить деление адресов подсетей в виде табл. 4.3, приведенной в разделе 4.2.3.

Составить таблицу маршрутов маршрутизатора R для обслуживания всех подсетей в виде табл. 4.4, приведенной в разделе 4.2.3 учебника.

Определить на какие интерфейсы будут направлены пакеты с IP-адресами назначения 131.40.Y1.Y2, 131.40.Y2.Y3, 131.40.Y3.Y4.

### **Варианты для выполнения работы**

Варианты для выполнения лабораторной работы приведены в таблице 8.12.

Таблица 8.12

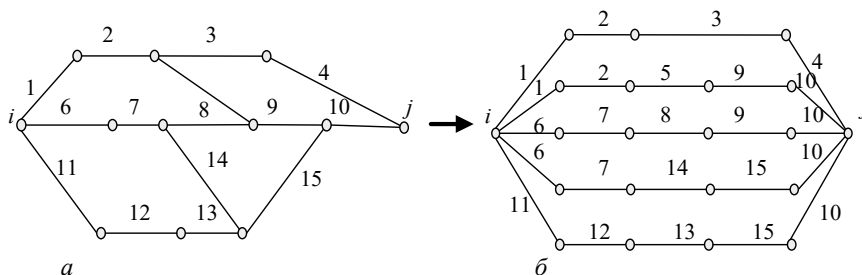
№ варианта	Y1	Y2	Y3	Y4
1	13	14	18	15
2	23	25	30	26
3	33	36	42	38
4	43	47	50	45
5	53	58	61	55
6	63	69	72	65
7	73	74	75	77
8	83	84	88	92
9	93	95	99	97
10	13	16	21	18
11	23	26	32	30
12	33	38	41	40

### 8.9. Статистическое оценивание функциональной надежности сети

**Цель работы:** оценить функциональную надежность установления логического соединения на сети.

#### Общие сведения

Установить логическое соединение – это значит построить логический канал, связывающий узел источника с узлом адресата для последующей передачи по нему данных. Таких логических каналов, связывающих источник и адресата, может быть несколько. Совокупность кратчайших путей между источником и адресатом образует гамак (рис. 8.6).



**Рис. 8.6. Соединение между источником  $i$  и адресатом  $j$ :  $a$  – направление передачи;  $b$  – гамак кратчайших путей**

Логический канал состоит из физических каналов, связывающих узловые точки (шлюзы и маршрутизаторы), входящие в данный логический маршрут. Построение логического канала выполняется посылкой управляющего сигнала-вызова на установление сеанса связи. При прохождении вызова от одного узла к другому состояние физического канала принимает одно из двух состояний: «1» – канал занят, и данные по нему не пройдут, либо «0» – канал обладает требуемой скоростью для передачи данных.

Повторные попытки установления соединения приводят к вынужденным возвращениям на предыдущие узлы. Поэтому число физических каналов, пройденных вызовом при его доставке адресату, оказывается случайным числом.

Учет данного аспекта позволяет прибегнуть к имитационному моделированию для установления логического канала между источником  $i$  и адресатом  $j$ .

Таким образом, время установления логического соединения  $t_{y.c}$  определим как случайную величину, которая может быть найдена выражением

$$t_{y.c} = \sum_{i=1}^{n_k} t_{\kappa_i} + \sum_{i=1}^{n_o} t_{o_i} + n_{\pi} t_{\pi}, \quad (8.1)$$

где  $n_k$  – число физических каналов в логическом канале, построенном от источника к адресату;

$n_o$  – число физических каналов, на которые вызов вернулся обратно при поиске альтернативного логического канала;

$n_{\pi}$  – число попыток в зафиксированной реализации процесса установления логического канала; в общем случае  $0 \leq n_{\pi} < n_{\text{доп}}$ , где  $n_{\text{доп}}$  – допустимое число попыток установления соединения;

$t_{\kappa_i}$  – время прохождения  $i$ -го физического канала;

$t_{o_i}$  – время обратного прохождения  $i$ -го физического канала;

$t_{\pi}$  – время переключения на другой логический канал.

Каждый эксперимент на имитационной модели дает реализацию трех случайных величин:  $n_k$ ,  $n_o$  и  $n_{\pi}$ , что позволяет оценить время установления соединения  $t_{y.c}$  в соответствии с выражением (8.1). Полученное значение  $t_{y.c}$  определяет результат установления соединения:

- если  $t_{y.c} \leq t_{\text{доп}}$ , то логический канал установлен;
- если  $t_{y.c} > t_{\text{доп}}$ , то логический канал установлен, но за время, превышающее допустимое, и для данных срочной доставки это имеет критическое значение, т. к. они могли потерять свою актуальность;
- если  $n_{\pi} > n_{\text{доп}}$ , то соединение не установлено.

### **Описание работы**

Исходными данными для моделирования являются:

- множество логических каналов, которые могут быть построены от источника  $i$  к адресату  $j$ , – обозначим как  $L_{ij}$ .
- характеристика физических каналов: время передачи сигнала-вызова по физическому каналу в прямом и обратном направлениях;
- значения вероятностей полной занятости физических каналов;
- допустимое время установления соединения;
- число попыток установления соединения;
- время, выделенное на повторную попытку – переключение на другой логический канал.

В модели накапливаются статистики, позволяющие оценить вероятность установления соединения за время, не превышающее допустимое, а также средние и среднеквадратические значения  $t_{y.c}$ , характеризующие процесс установления соединения в инфокоммуникационной сети.

На всем множестве логических каналов  $L_{ij}$  производится классификация по числу занятых физических каналов  $c$ , то есть ( $c = c_{\min}, \dots, c_{\max}$ ). В отдельном

эксперименте разыгрывается число  $c$  номеров занятых физических каналов во множестве  $L_{ij}$  и на полученной реализации имитируется процесс прохождения вызова от источника  $i$  к адресату  $j$ . По факту установления информационного взаимодействия фиксируются значения  $n_{тр}$ ,  $n_{от}$  и  $n_{п}$ . Процедура повторяется  $N$  раз. По результатам экспериментов производятся оценка времени установления информационного взаимодействия.

Реализация количества занятых физических каналов  $c$  сводится к «выбору наугад» номеров каналов из  $d$  возможных,  $c \in d$ . Очередной номер занятого физического канала  $z$  определяется по формуле  $z = \lceil Ud + 1 \rceil$ , где  $U$  – случайное число,  $U \in [0,1]$ , получаемое путем обращения к датчику случайных чисел. Скобки  $\lceil \rceil$  означают округление в меньшую сторону. Физическому каналу с номером  $z$  присваивается «1» во множестве  $L_{ij}$ . Процедура определения  $z$  повторяется  $c$  раз.

### Порядок выполнения работы

Работа выполняется в программе Planet. Сначала создается структурная модель инфокоммуникационной сети, а потом проводится имитационный эксперимент по оценке ее функциональной надежности.

1. Выбрать или создать структуру моделируемой сети, на которой будет производиться моделирование процесса установления соединения. Структура сети (топология) создается самостоятельно – задается нужное количество узлов коммутации, их взаимное соединение, каналы связи и их длина. Для создания структуры сети используются пункт меню **Создать** и команды **Добавить**, **Удалить** из меню **Net Editor**.

Положение узла может быть задано в любом месте поля экрана, при этом номер узла и его производительность отображаются в правой части графического редактора. Физические каналы связи (КС) создаются указанием номеров узлов, которые он соединяет. Для каждого КС необходимо выбрать: пропускную способность, в бодах; скорость прохождения физического канала в прямом  $t_{к_i}$  и обратном  $t_{о_i}$  направлениях

Сохранить структурную модель ИКС – (пункт меню **Сохранить**).

Выйти из графического редактора (**ALT-X** или пункт меню **Файл-выход**).

1. Составить план экспериментов с моделью установления соединения в ИКС. Цель эксперимента – построить зависимости абсолютных и вероятностных характеристик установления соединения, такие как количество установленных соединений, вероятность установления информационного взаимодействия  $P_{у.в}$  и времени установления соединения  $\bar{t}_{у.в}$  от

- количества непроводящих каналов  $c$ ;
- количества закрытых узловых точек  $N$ ;
- количества попыток  $n_{п}$ .

Результаты эксперимента позволят сделать рекомендации по выбору маршрутов, характеристик производительности узлов и каналов связи, позволяющих организовать взаимодействие между источником и адресатом с требуемыми значениями  $P_{у.с}$  и  $t_{доп}$



2. Провести численные эксперименты на выделенном направлении в соответствии с планом эксперимента:

- выбрать узел-источник и узел-адресат (они окрасятся в красный цвет);
- задать характеристики процесса установления соединения: время переключения на другой маршрут  $t_{п}$ , допустимое время на установление соединения  $t_{доп}$ , допустимое число попыток установления соединения  $n_{доп}$  (пункт меню **Параметры** → **Основные параметры**).
- задать параметры имитационной модели: скорости работы модели, приоритеты отправки пакетов, возможность визуализации (Да/Нет) процесса установления соединения (пункт меню **Параметры** → **Приоритет отправки, Анимация**).

3. Запустить имитационную модель установления соединения (пункт меню **Старт**). Вся последовательность событий и текущих состояний отображается в нижнем углу поля, гамак кратчайших путей в правой части экрана. После завершения работы модели, на экране появляется статистика по основным и дополнительным оценкам.

5. Оформить отчет о проделанной работе, в который должны войти выбранная структура сети, «гамак» кратчайших путей, набор исходных данных и полученные результаты в виде зависимостей п. 2. с разъяснениями и выводом.

## ЗАКЛЮЧЕНИЕ

Современные инфокоммуникационные технологии эволюционируют уже более 40 лет. Их задачи все эти годы постоянно усложнялись. Создавались и создаются новые технологии взаимодействия в распределенных информационных системах.

Централизация хранения и обработки данных способствовала интенсивному развитию и внедрению терминальных систем. Решение благоприятное с точки зрения защиты информации. Применение терминальных систем оправдано везде, где большое количество пользователей решают типовой набор задач, не требующих от локальной информационной системы максимальной производительности. В России наблюдается устойчивая тенденция внедрения терминальных систем практически у всех крупных компаний.

В последнее десятилетие большое внимание уделяется разработке программно-конфигурируемых сетей (Software-Defined Networks, SDN), работающих на основе протокола OpenFlow. Сети SDN позволяют отделить уровень данных от уровня управления сетью. Благодаря такому решению сетевые администраторы получают детализированный контроль над трафиком, а протокол конфигурации OpenFlow (OF-CONFIG) предоставляет возможность удаленного конфигурирования обмена данными, что позволяет вносить оперативные изменения в работу сетевой инфраструктуры. Протокол OpenFlow также поддерживает специфический метод инкапсуляции.

Это два примера того, что относительная легкость изменения функционального наполнения разного телекоммуникационного оборудования (преимущественно за счет его перепрограммирования) приводит к значительному возрастанию многообразия форм и способов воплощения сетевых технологий. Такое многообразие затрудняет понимание отличительных особенностей реализации взаимодействия в информационных сетях. В то же время основные принципы и модели построения инфокоммуникационных сетей за последние годы мало изменились. Приведенная в учебнике систематизация и разъяснение этих принципов на примере конкретных сетевых технологий способствуют осознанному усвоению учащимися курса «Инфокоммуникационные системы и сети».

## Приложение 1. Технологии построения глобальных сетей

### П1.1. Архитектура и технологии построения сетей X.25

Рекомендация (стандарт, технология) X.25 имеет следующее название: «Интерфейс между *оконечным оборудованием данных (ООД или DTE – Data Terminal Equipment)* и *аппаратурой окончания канала данных (АКД или DCE – Data Channel Equipment)* для оконечных установок, работающих в пакетном режиме и подключенных к сетям передачи данных общего пользования по выделенному каналу».

Рекомендация X.25 включает описание процедур (протоколов) трех нижних уровней ЭМВОС: физического, звена данных и сетевого (а также частично транспортного).

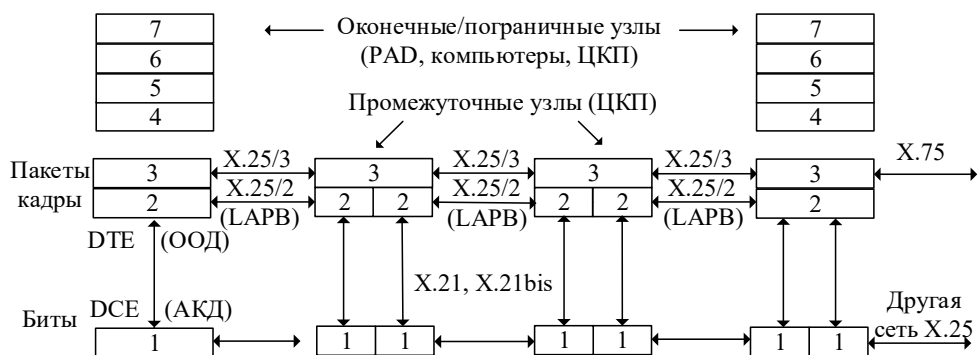
Стандарт X.25 только определяет пользовательский интерфейс с сетью. Взаимодействие узлов коммутации внутри сети не стандартизуется. Для согласованного взаимодействия разных сетей X.25 существует специальный стандарт – X.75.

Основными элементами сети являются оконечные устройства (асинхронные старт-стопные терминалы), *сборщики-разборщики пакетов (СРП или PAD – Packet Assembler-Disassembler)* и узлы коммутации (*packet switch*), именуемые обычно *центрами коммутации пакетов (ЦКП или PSE – Packet Switching Exchange)*, каналы связи.

PAD используется для доступа в сеть терминалов при асинхронном режиме обмена информацией (посимвольном). Он обычно имеет несколько асинхронных портов и один синхронный порт X.25. PAD накапливает поступающие через асинхронные порты данные, упаковывает их в пакеты и передает через порт X.25.

Компьютеры (мэйнфреймы) и локальные сети обычно подключаются к сети X.25 непосредственно через *адаптер X.25* или маршрутизатор, поддерживающий на своих интерфейсах протоколы X.25.

Стек (профиль) основных протоколов, от которого зависит архитектура сети X.25, показан на рис. П1.1.



**Рис. П1.1. Архитектура сети X.25**

**Физический уровень.** В Рекомендациях X.25 (X.25/1) для реализации физического уровня предлагается использовать протоколы **X.21** и **X.21bis**. Стыки между ООД и АКД, описываемые этими стандартами, содержат: механические характеристики; электрические характеристики; функциональные характеристики, задающие тип, число и назначение соединительных цепей стыка ООД/АКД; процедурные характеристики, определяющие последовательность изменения состояния цепей интерфейса ООД/АКД, то есть логику взаимодействия на физическом уровне.

**Уровень звена данных.** В Рекомендации X.25/2 указывается на необходимость использования на уровне звена данных процедуры управления звеном **LAPB** (*Link Access Protocol, Balanced*), при котором обеспечивается обмен по двухточечному соединению. Процедура сбалансированная (симметричная).

В соответствии с протоколом LAPB обмен данными осуществляется кадрами, формат которых приведен на рис. П1.2.

Флаг 01111110	Адрес	Управление	Данные (пакет)	Проверочная последовательность (CRC)	Флаг 01111110
1 байт	1 байт	1 – 2 байта	1 ... $2^{10}$ байта	2 байта	1 байт

**Рис. П1.2. Формат кадра X.25/2 (LAPB)**

Особенностью процедуры LAPB является обязательный запрос повторения кадров, в которых обнаруживаются ошибки. Для этого используется процедура отрицательного подтверждения с помощью служебных S-кадров REJ. Не убедившись в правильном приеме очередного кадра, уровень звена данных не будет принимать следующие кадры. Это является одной из причин случайных и длительных задержек передачи информации в сетях X.25.

**Сетевой уровень.** В соответствии с Рекомендацией X.25/3 протокол сетевого уровня (**PLP** – *Packet-Layer Protocol*) предоставляет пользователю возможность информационного взаимодействия с другими пользователями сети посредством *временных* (*SVC* – *Switch Virtual Circuit*) или *постоянных* (*PVC* – *Permanent Virtual Circuit*) виртуальных каналов. Наиболее распространены временные соединения (*SVC*).

Каждому виртуальному соединению присваиваются номера группы и логического канала. Число групп равно 15, в каждой из них содержится 255 логических каналов. Каждому пользователю при его постановке на учет администрацией сети может выделяться множество логических каналов.

Протокол PLP является протоколом управления маршрутизацией, т.е. управляющим коммутацией пакетов в ЦКП с целью их передачи через сеть X.25 от узла к узлу по заранее проложенному маршруту на основании маршрутных таблиц, хранящихся в памяти каждого ЦКП, и адресных признаков (адресных данных), записанных в специальном служебном поле пакетов. Специальных собственных протоколов маршрутизации, динамически выбирающих наилучшие маршруты и автоматически корректирующих маршрутные таблицы, в сетях X.25 нет.

## П1.2. Архитектура и технологии построения сетей Frame Relay

*Frame Relay* – это технология построения сети передачи данных с *ретрансляцией кадров*.

Стандарты FR описывают интерфейс доступа к сетям с быстрой коммутацией пакетов и включают процедуры (протоколы) двух нижних уровней ЭМ-ВОС – физического и звена данных (не полностью, но с дополнительными функциями сетевого уровня). Технология обеспечивает образование и поддержку множества независимых виртуальных каналов в одном звене, но не имеет средств коррекции и восстановления кадров при возникновении ошибок. Вместо средств управления потоком в протоколе FR реализованы функции извещения о перегрузках в сети.

FR позволяет эффективно передавать крайне неравномерно распределенный во времени трафик. Отличается малым временем задержки, высокими скоростями (до 2 Мбит/с). Недостаток – требует каналов высокого качества (с вероятностью ошибки  $10^{-7}$  и лучше).

Основными элементами сети FR являются оконечные устройства (терминалы), *устройства доступа* FRAD (*Frame Relay Access Device* или по аналогии с PAD – *Frame Relay Assembler / Disassembler*), узлы коммутации и каналы физической среды передачи. В литературе узлы коммутации FR, используемые для соединения локальных сетей, часто обозначают в виде мостов (bridge) или коммутаторов (switch).

В роли оконечных устройств сетей FR выступают компьютеры и локальные сети, подключаемые к сети FR через адаптер FR, коммутатор или маршрутизатор, поддерживающий на своих интерфейсах протоколы FR. Старт-стопные терминалы в сетях FR практически не используются.

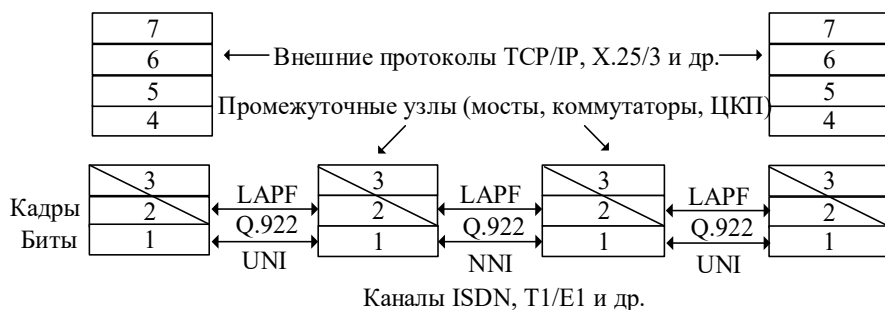
Сети FR сразу же были сориентированы на возможное объединение с другими сетями как путем инкапсуляции пакетов других (внешних) сетей (например, TCP/IP, X.25, LAN и др.) в кадры FR, так и путем инкапсуляции кадров FR в ПБД других (внутренних) сетей (в частности, ATM и др.). Передача речи в сетях FR побудила развитие технологий VoFR (Voice over Frame Relay – голос поверх Frame Relay) и создание шлюзов с ТфОП.

Сети FR очень популярны в корпоративных (ведомственных) сетях (в том числе специального назначения), в частности, для объединения локальных сетей, а также в сетях доступа отдельных компьютеров и локальных сетей к глобальным сетям (ATM, TCP/IP, ISDN).

Стек (профиль) основных протоколов, характеризующих архитектуру сети FR, показан на рис. П1.3.

Формат кадра протокола LAPF приведен на рис. П1.4.

В сетях FR на уровне звена данных используется процедура управления звеном **LAPF** (*Link Access Protocol, Frame Relay*), являющаяся, с одной стороны, упрощенным, а с другой – дополненным вариантом процедуры **HDLC** (*High-level Data Link Control*).



**UNI (User Network Interface)** – интерфейс «пользователь – сеть»  
**NNI (Network Network Interface)** – интерфейс межсетевое взаимодействия

**Рис. П1.3. Архитектура сети Frame Relay**

Кадр LAPF	Флаг 01111110	Заголовок	Данные (пакет)	(FCS)	Флаг 01111110
	1 байт	2 – 4 байта	1 – 4096 байт	2 байта	1 байт

**FCS (Frame Check Sequence)** – проверочная последовательность кадра

**Рис. П1.4. Формат кадра Frame Relay (LAPF)**

Одним из существенных отличий протокола FR от HDLC является то, что он не предусматривает передачу управляющих сообщений (нет нумерованных и супервизорных кадров, как в HDLC). Для передачи служебной информации используются специально выделенные виртуальные каналы сигнализации.

Другое важное отличие – отсутствие циклической нумерации передаваемых кадров. Это связано с тем, что в протоколе FR отсутствуют механизмы подтверждения правильно принятых кадров.

Основная процедура передачи кадров протокола FR состоит в том, что если кадр получен без искажений, он должен быть направлен далее по соответствующему маршруту (а если с искажениями, то он просто стирается).

В случае возникновения перегрузки в сети FR предусмотрено предупреждение источника и приемника, а также узлов коммутации вдоль маршрута следования пакетов начиная с узлов, соседствующих с узлом, на котором возникла перегрузка.

Особенностью технологии FR является отказ от коррекции обнаруженных в кадрах искажений. Протокол FR подразумевает, что конечные узлы будут выявлять и корректировать ошибки за счет работы протоколов транспортного или более высоких уровней.

### П1.3. Архитектура и технологии построения сетей ISDN

**ISDN (Integrated Services Digital Network)** – цифровая сеть с интеграцией служб (**ЦСИС**). Основной режим коммутации – коммутация каналов, а данные обрабатываются и передаются в цифровой форме. Отличительная черта ISDN (в сравнении ТФОП) – обеспечение полностью цифровых соединений между конечными устройствами и поддержка большого набора речевых и неречевых

служб, доступ к которым осуществляется через ограниченный набор стандартных многофункциональных интерфейсов.

Под ISDN подразумеваются узкополосные сети N-ISDN (Narrow-band ISDN) с максимальной предоставляемой скоростью 2 Мбит/с. Широкополосные сети с интеграцией служб B-ISDN (Broadband ISDN) или Ш-ЦСИС, построенные на основе технологии ATM, предоставляют скорости от 2 до 155 Мбит/с и более.

Сложность пользовательского интерфейса, отсутствие единых стандартов на многие важные функции, необходимость крупных капиталовложений для переоборудования телефонных АТС и каналов связи привели к тому, что период становления ISDN затянулся на многие годы.

В сущности, сети ISDN должны были выполнять де-юре роль массовой глобальной универсальной цифровой сети. Де-факто эту роль стала исполнять сеть Internet, построенная на принципах коммутации пакетов, а не каналов. Со временем стал все больше проступать недостаток базовой скорости доступа в ISDN по основному цифровому каналу 64 кбит/с. В итоге отдельные сети ISDN по всему миру продолжают разворачиваться, но остаются в основном на вторых ролях корпоративных сетей, фрагментов ТФОП и сетей доступа к другим, более высокоскоростным сетям, включая Internet.

Обобщенно сеть ISDN можно представить в виде совокупности сетей КК, КП и СС № 7, к которым через контрольные точки доступа подключается разнообразное терминальное оборудование (ТО), получая возможность интегрированного доступа к одной из этих сетей (рис. П1.5).

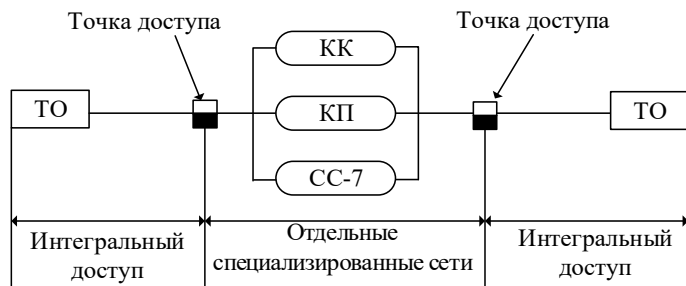


Рис. П1.5. Обобщенная структура сети ISDN

#### П1.4. Архитектура и технологии построения сетей ATM

ATM (Asynchronous Transfer Mode) – асинхронный режим доставки. Представляет собой метод коммутации, мультиплексирования и передачи пакетов (ячеек) постоянной длины (53 байта). Малая фиксированная длина ячеек и высокая скорость передачи (как правило, не менее 155 Мбит/с) обеспечивают малое время задержки передачи ячеек. Недостатки: высокая сложность (стоимость) и необходимость использования высококачественных каналов связи (с вероятностью ошибки  $10^{-9}$ – $10^{-12}$ ).

АТМ имеет трехуровневую архитектуру: физический уровень – PHY, уровень коммутации – АТМ и уровень адаптации – ААЛ (рис. П1.6).

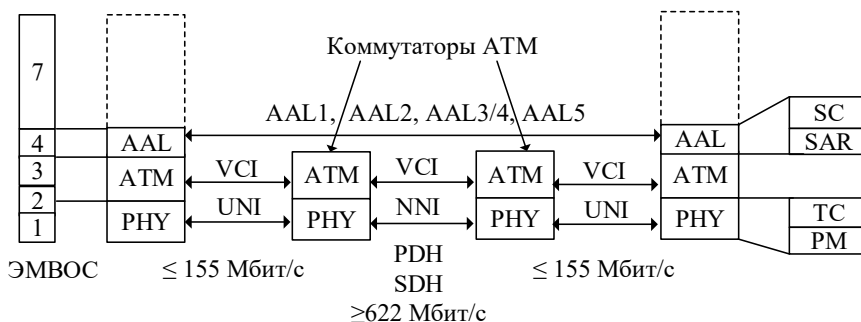


Рис. П1.6. Архитектура сети АТМ

**Физический уровень (PHY).** Стандарты АТМ для физического уровня определяют, каким образом отдельные биты должны проходить через среду передачи и как из последовательности бит выделять границы ячеек (согласно ЭМВОС, подобные функции выполняются на уровне звена данных).

Стандарт АТМ использует скорости 155 Мбит/с и 622 Мбит/с. На скорости 155 Мбит/с можно использовать волоконно-оптический кабель, незранированную витую пару (UTP) категории 5, РРЛ. На скорости 622 Мбит/с допустим только волоконно-оптический кабель, как одномодовый, так и многомодовый.

**Уровень АТМ (АТМ).** Стандарты для уровня АТМ регламентируют выполнение:

- 1) функций передачи информационных сигналов в виде ячеек определенного формата;
- 2) функций управления трафиком (очередностью, скоростью, задержками и отбрасыванием ячеек);
- 3) функций установления соединений и маршрутизации.

В сетях АТМ поддерживаются три типа виртуальных соединений:

**PVC** (*permanent virtual circuits*) – постоянные виртуальные соединения, которые образуются в результате соглашения об обслуживании между оператором и пользователем;

**SVC** (*switched virtual circuits*) – коммутируемые виртуальные соединения, которые устанавливаются по требованию протокола сигнализации между терминалом пользователя и коммутационным блоком доступа к сети;

**SPVC** (*smart permanent virtual circuits*) – интеллектуальные постоянные виртуальные соединения. Гибрид PVC и SVC. Подобно PVC, SPVC устанавливаются вручную на этапе конфигурирования сети. Провайдер или сетевой администратор задает только конечные станции. Для каждой передачи сеть определяет, через какие коммутаторы будут передаваться ячейки.

### Категории услуг протокола АТМ

Для поддержания требуемого качества обслуживания различных виртуальных соединений и рационального использования ресурсов в сети на уровне



протокола ATM реализовано несколько служб, предоставляющих услуги разных категорий (*service categories*) по обслуживанию пользовательского трафика. Всего на уровне протокола ATM определено пять категорий услуг, которые поддерживаются одноименными службами:

**CBR** (*constant bit rate*) – услуги для трафика с постоянной битовой скоростью;

**rtVBR** (*real time variable bit rate*) – услуги для трафика с переменной битовой скоростью, требующего соблюдения средней скорости передачи данных и синхронизации источника и приемника;

**nrtVBR** (*not real time variable bit rate*) – услуги для трафика с переменной битовой скоростью, требующего только соблюдения средней скорости передачи данных, но не синхронизации источника и приемника;

**ABR** (*available bit rate*) – услуги для трафика с переменной битовой скоростью, требующего соблюдения некоторой минимальной скорости передачи данных, а не синхронизации источника и приемника;

**UBR** (*unspecified bit rate*) – услуги для трафика, не предъявляющего требований к скорости передачи данных и синхронизации источника и приемника.

#### ***Уровень адаптации ATM (AAL)***

Стандарты для уровня адаптации ATM **AAL** (*ATM adaptation layer*) регламентируют выполнение следующих основных функций:

- 1) форматирование пакетов, передаваемых в поле данных ячеек;
- 2) предоставление управляющей информации для уровня ATM, необходимой для установления соединений с различным требуемым качеством сервиса QoS;
- 3) управление последовательностью и скоростью передачи пакетов (с целью предотвращения перегрузок).

Уровень AAL намного сильнее связан с областями использования, чем уровень ATM. Он повышает качество обслуживания, предоставляемого уровнем ATM, в соответствии с требованиями пользователя. Уровень AAL применяет протоколы из конца в конец, прозрачные для уровня ATM, что соответствует транспортному уровню ЭМВОС. Если ПБД (пакеты) уровня AAL превышают длину информационного поля ячейки ATM, то они сегментируются (фрагментируются).

Различные услуги требуют выполнения разных функций уровня адаптации. Однако, чтобы избежать слишком большого разнообразия, были выделены 4 класса обслуживания (A, B, C, D), отличающиеся сочетанием трех бинарных характеристик трафика:

- *битовая скорость передачи* может быть постоянной или переменной;
- *установление соединений* может требоваться или не требоваться;
- *строгая взаимосвязь между тактовыми частотами* источника и приемника может требоваться или не требоваться.

Для обеспечения качества услуг определены четыре протокола уровня адаптации AAL-1, AAL-2, AAL-3/4 и AAL-4.

**AAL-1** используется для передачи информации с постоянной битовой скоростью, которая требует строгой взаимосвязи между тактовыми частотами передачи и приема (например, для эмуляции речевого канала);

**AAL-2** применяется для передачи информации с переменной битовой скоростью, которая требует строгой взаимосвязи между тактовыми частотами передачи и приема (передача видео с переменной битовой скоростью);

**AAL-3/4** используется для передачи данных как с установлением соединений, так и без него;

**AAL-5** применяется для передачи данных только с установлением соединений.

Различные классы обслуживания через разные протоколы AAL опираются на соответствующие категории услуг, учитываемые на уровне ATM.

Уровень адаптации включает подуровни SAR и CS.

**Подуровень сегментации и сборки SAR** (*Segmentation And Reassembly sublayer*) отвечает за изменение формата блоков данных пользователя и полезной нагрузки ячеек. Поля AAL, соответствующие этому подуровню, представлены в каждой ячейке. Такой подуровень способен обнаруживать потерю или дублирование ячеек благодаря их нумерации, однако само восстановление является функцией подуровня сведения. И, наконец, подуровень SAR производит заполнение неполных ячеек.

**Подуровень сведения (конвергенции) CS** (*Convergence Sublayer*) выполняет большей частью специальные функции обслуживания пользователя. При необходимости отвечает за обработку ошибок. Применяет протоколы для повторной передачи ошибочных данных или защищает данные, предоставляя приемнику возможность исправлять ошибки. Этот метод прямого исправления ошибок применяется, в частности, в приложениях реального времени. Подуровень CS может также обеспечивать синхронизацию из конца в конец.

При использовании AAL-3/4 и AAL-5 пакеты, формируемые на подуровне CS, могут иметь длину или больше, или меньше поля данных ячейки, потому на подуровне SAR с ними выполняется процедура или сегментации (фрагментации), или группирования (блокирования).

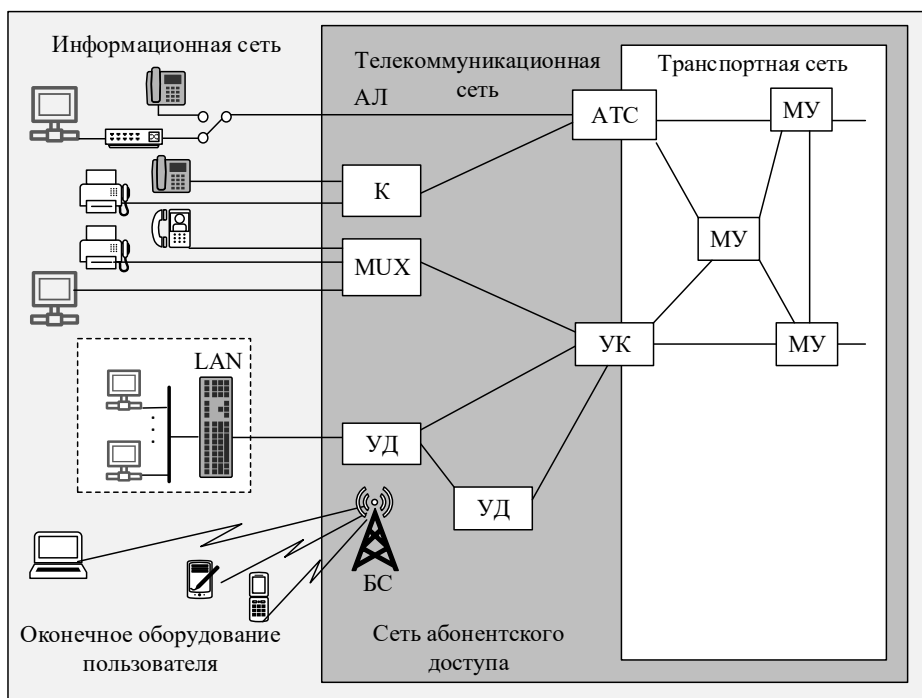
### **П1.5. Технологии построения сетей абонентского доступа**

От сети абонентского доступа требуется обеспечить персональный доступ к любым информационным и телекоммуникационным услугам любым абонентам независимо от их местонахождения, т.е. обеспечить персональную глобальную связь по принципу *«всегда и везде»*.

На рис. П1.7 показан фрагмент телекоммуникационной сети с выделенными типовыми элементами сети абонентского доступа (преимущественно на основе ТФОП).

В простейшем случае абонентская сеть состоит из трех основных элементов: абонентских терминалов (АТ); абонентских линий (АЛ); узла коммутации (УК).

Специальные технологии абонентского доступа нацелены на образование цифровых каналов на основе доступной физической среды *проводного и беспроводного доступа*.



**Рис. П1.7. Типовая структура и состав сетей абонентского доступа**

МУ – магистральный узел; УК – узел коммутации; К – удаленный концентратор; MUX – мультиплексор; УД – узел доступа, БС – базовая станция беспроводного доступа

Перспективные концепции построения САД ориентируются в основном на физические среды, позволяющие передавать высокоскоростные потоки информации, то есть на оптоволокно.

В последние годы получили широкое распространение технологии комплексного использования различных доступных физических сред.

В качестве подобной технологии построения САД может выступать любая технология LAN, способная обеспечить необходимую дальность связи, например FDDI или версии высокоскоростных технологий Ethernet.

Стали появляться смешанные технологии построения САД, включающие элементы технологий LAN, WAN и новых технологий образования цифровых абонентских линий (ЦАЛ) на основе уже проложенных медных пар.

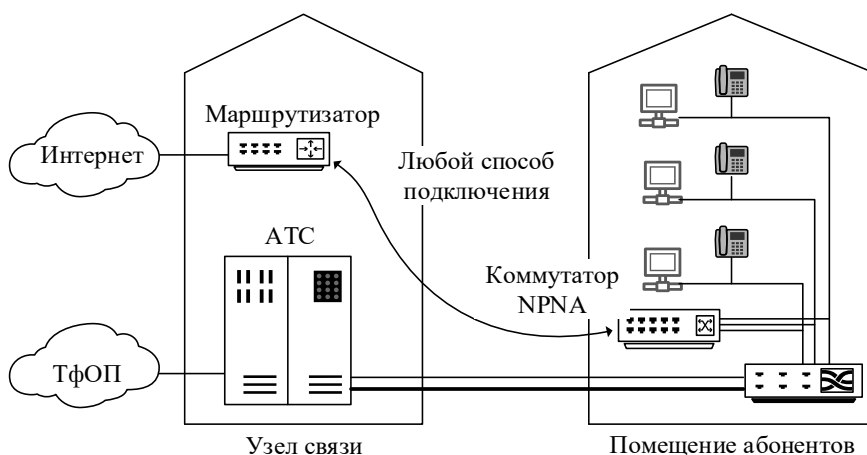
Например, технология HomePNA (Home Phoneline Networking Alliance) или HPNA (рис. П1.8), совмещающая в себе свойства технологии LAN Ethernet и технологии ЦАЛ xDSL.

В стандарте HomePNA 2.0 с целью повышения информационной скорости передачи до 10 Мбит/с применяется квадратурно-амплитудная модуляция передаваемого сигнала (QAM). При этом тип модуляции может изменяться так, что информационная скорость меняется от 2 до 8 бит/символ.

Ожидается появление новой версии HPNA, которая будет обеспечивать скорость передачи до 100 Мбит/с.

Сеть на основе оборудования HPNA может использовать топологии «звезда» и «шина».

В случае применения топологии «звезда» (рис. П1.8) используется коммутатор, который имеет несколько портов HPNA и WAN-порт для подключения к глобальной сети передачи данных. Такой вариант применяется, когда в здании уже существует телефонная проводка – коммутатор устанавливается вблизи телефонного кросса здания, подсоединяясь параллельно к телефонным линиям без частотных разделителей. Не требуются разделители и на стороне абонента. При этом каждому абоненту выделяется порт коммутатора, и абоненту гарантируется широкополосное подключение к Internet (скорости в 1 Мбит/с и выше). Порт WAN подключается к сети передачи данных оператора любым способом (выделенная линия, xDSL, оптика, радио).



**Рис. П1.8. Пример построения САД на основе технологии HPNA**

В случае использования топологии «шина» возможно объединение до 32 абонентов HPNA с помощью одной пары проводов. Такое решение является единственным выходом, если провайдер не имеет доступа к абонентской телефонной проводке. Но в этом случае полоса пропускания будет делиться между всеми абонентами.

Существуют варианты применения HomePNA типа «общей шины» для организации передачи данных *по сетям проводного вещания*. Низшее звено таких сетей – абонентская линия – уже имеет топологию «шина» (именно к ней подключаются абонентские громкоговорители).

Система проводного вещания является уникальной инфраструктурой. Такая система не только используется для передачи трех радиопрограмм, но и служит частью системы гражданской обороны и предупреждения о чрезвычайных ситуациях. Имеет смысл обратить внимание на нее и с точки зрения использования в системах абонентского доступа.

## Приложение 2. Описание основных протоколов семейства TCP/IP

**ARP** (Address Resolution Protocol, протокол определения адресов): конвертирует 32-разрядные IP-адреса в физические адреса вычислительной сети, например в 48-разрядные адреса Ethernet.

**FTP** (File Transfer Protocol, протокол передачи файлов): позволяет передавать файлы с одного компьютера на другой с использованием TCP-соединений. В родственном ему, но менее распространенном протоколе передачи файлов – Trivial File Transfer Protocol (TFTP) – для пересылки файлов применяется UDP, а не TCP.

**ICMP** (Internet Control Message Protocol, протокол управляющих сообщений Internet): позволяет IP-маршрутизаторам посылать сообщения об ошибках и управляющую информацию другим IP-маршрутизаторам и главным компьютерам сети. ICMP-сообщения «путешествуют» в виде полей данных IP-датаграмм и обязательно должны реализовываться во всех вариантах IP.

**IGMP** (Internet Group Management Protocol, протокол управления группами Internet): позволяет IP-датаграммам распространяться в циркулярном режиме (multicast) среди компьютеров, которые принадлежат к соответствующим группам.

**IP** (Internet Protocol, протокол Internet): низкоуровневый протокол, который направляет пакеты данных по отдельным сетям, связанным вместе маршрутизаторами для формирования Internet или интрасети. Данные «путешествуют» в форме пакетов, называемых IP-датаграммами.

**RARP** (Reverse Address Resolution Protocol, протокол обратного преобразования адресов): преобразует физические сетевые адреса в IP-адреса.

**SMTP** (Simple Mail Transfer Protocol, простой протокол обмена электронной почтой): определяет формат сообщений, которые SMTP-клиент, работающий на одном компьютере, может использовать для пересылки электронной почты на SMTP-сервер, запущенный на другом компьютере.

**TCP** (Transmission Control Protocol, протокол управления передачей): ориентирован на работу с подключениями и передает данные в виде потоков байтов. Данные пересылаются пакетами – TCP-сегментами, которые состоят из заголовков TCP и данных. TCP – «надежный» протокол, потому что в нем используются контрольные суммы для проверки целостности данных и отправка подтверждений, чтобы гарантировать, что переданные данные приняты без искажений.

**UDP** (User Datagram Protocol, протокол пользовательских дейтаграмм): не зависит от подключений, передает данные пакетами, называемыми UDP-датаграммами. UDP – «ненадежный» протокол, поскольку отправитель не получает информацию, показывающую, была ли в действительности доставлена датаграмма.

**Telnet** (*tele* – далеко, *net* – сеть) – протокол удаленного доступа, обеспечивающий побайтный обмен информацией между терминалами и внутрисетевыми элементами (узлами, компьютерами, хостами). Передача осуществляется

с использованием протокола TCP. Работа с протоколом на компьютере напоминает работу с программой Hyper Terminal в ОС Windows.

**NFS** (Network File System), сетевая файловая система: использует транспортные услуги UDP и позволяет монтировать в единое целое файловые системы нескольких машин с ОС UNIX. Бездисковые рабочие станции получают доступ к дискам файл-сервера так, как будто это их локальные диски.

**SNMP** (Simple Network Management Protocol, простой протокол управления сетью): использует транспортные услуги UDP; позволяет управляющим станциям собирать информацию о положении дел в сети Internet. Протокол определяет формат данных, их обработка и интерпретация остаются на усмотрение управляющих станций или менеджера сети.

**HTTP** (Hyper Text Transport Protocol, протокол передачи гипертекстовой информации): основа формирования всемирной информационной службы (всемирной информационной паутины) – **WWW** (*World Wide Web*). Использует транспортные услуги TCP.

**RTP** (Real-time Transport Protocol, протокол передачи данных в реальном времени): разработан для обеспечения передачи аудио- и видеосигналов по сети Internet с ограниченной допустимой задержкой (**IP-телефония**). Использует транспортные услуги UDP. Тесно связан с еще одним протоколом прикладного уровня – **RTCP** (Real-time Transport Control Protocol, протокол управления передачей в реальном времени). С помощью данного протокола прикладные программы могут приспосабливаться к изменению нагрузки на сеть. Например, в случае перегрузки, получив сигнал RTCP, алгоритм кодирования речи может увеличить сжатие (снизив качество).

## Листинг имитационной модели работы №1

\*\*\*\*\*

\*\* MODEL OF INFORMATION LINK \*  
 \*\*\*\*

20	INITIAL	X1,2000 ;message length
30	INITIAL	X2,512 ;standard packet length
40	INITIAL	X3,48 ;length of control packet
50	INITIAL	X4,1000 ;probability of error
60	INITIAL	X5,800 ;propagation delay
70	INITIAL	X6,1500 ;time-out of answerback
75	INITIAL	X7,100000 ;time of modeling
80	INITIAL	X8,0
85	INITIAL	X9,0
90	INITIAL	X10,1
93	INITIAL	X11,0
94	INITIAL	X12,1
95	INITIAL	X13,0
96	INITIAL	X14,0
100	BUF1	STORAGE 3 ; buffer for window
110	BUF2	STORAGE 1
120	LKF	VARIABLE X1@X2 ;length of non-standard packet
130	LKI	VARIABLE X1/X2 ;number of standard packet
140	LKS	VARIABLE X1/X2+1 ;number of packet
150	LKY	VARIABLE P3-1 ;number of packet - 1
160	LKX	VARIABLE X1/X2-1 ;number of copy
170	TKC1	VARIABLE P4+48 ; information frame length
180	TKC2	VARIABLE X3+48 ; control frame length
190	P_ER1	VARIABLE V\$TKC1#X4/1000 ; probability of
200	P_ER2	VARIABLE V\$TKC2#X4/1000 ; error per frame
211	HOM_R	VARIABLE X12-1 ;acknowledge frame number
212	CYCLE	VARIABLE (X2+48)+X5+(X3+48)+X5 ;stand.cycle
213	R_CYCLE	VARIABLE X14#100/V\$CYCLE ;relative rate
220	TIME	TABLE MP5,0,300,20 ;frame delivery time
230	TIM_M	TABLE MP1,0,4000,8;message delivery time
240	RATE	TABLE V\$R_CYCLE,0,10,10;relative rate

```

250      GENERATE    1,,1,1 ; first message
260 MBF  SAVEVALUE  9+,1   ; number of message
270      ASSIGN     1,C1    ; arrival time
280      ASSIGN     2,0     ; initial packet number
295      TEST E      V$LK F,0,MBF1 ; start of segmentation
300      ASSIGN     3,V$LKI ; flag of number segment
310      ASSIGN     4,X2    ; flag of segment length
320      SPLIT      1,KLS   ; start of packet selection
330      SPLIT      V$LKY,MBF3,2 ; generation of packets
340      TRANSFER    ,MBF3  ; on buffering
350 MBF1 ASSIGN     3,V$LKS ; flag of number segments

```

```

360  ASSIGN      4,X2      ;flag of segment length
370  SPLIT      1,KLS     ;start of packet selection
380  SPLIT      V$LKX,MBF3,2;generation of packet
390  SPLIT      1,MBF4     ;on non-standart packet
400  TRANSFER   ,MBF3     ;on buffering
410 MBF4 ASSIGN   2,V$LKS   ;number flag
420  ASSIGN      4,V$LKf   ;flag of segment length
430  TRANSFER   ,MBF3     ;on buffering
432 MBF3 SAVEVALUE 13+,1   ;counter of frames
435  ASSIGN      8,X13     ;frame numbering
436  ASSIGN      9,0       ;transmission flag
437 MBF5 LINK    BLOKS,P8  ;buffering
470 BF  ENTER    BUF1,1    ;output buffer
480 DL  SEIZE     DL1
490  SAVEVALUE   8,P8
492  TEST E      P9,0,DL11
494  ASSIGN      5,C1
496  ASSIGN      9,1
498 DL11 ADVANCE V$TKC1
500  RELEASE     DL1
501  SPLIT      1,YZ
502  SPLIT      1,TIM_A
503  SPLIT      1,DL12
504  PRIORITY    0,BU
505  ADVANCE     X5
506  TRANSFER    V$P_ER1,,KAN13
508  ASSIGN      6,0
510  TRANSFER    ,MOD2
512 KAN13 ASSIGN  6,1
514  TRANSFER    ,MOD2
515 YZ  LINK     WIND,P8
516 KLS  UNLINK   BLOKS,BF,1,,,MBF
517  TERMINATE
518 TIM_A TEST E  F$TMR,0,TIM_B
520 TIM_C SEIZE   TMR
522  ADVANCE     X6
524  RELEASE     TMR
526  TRANSFER    ,REJ_T
528 TIM_B PRIORITY 10
529  PREEMPT     TMR,PR,T_END,,RE
530  RETURN      TMR
531  PRIORITY     0
534  TRANSFER    ,TIM_C
536 T_END TERMINATE
538 DL12 TEST E  R$BUF1,0,KLS
540  TERMINATE
550 MD1  ADVANCE  1
555  TEST E      P6,0,ERR1
560  TEST E      P7,1,REJ
565  TEST LE     P8,X8,ERR2
570 MD_E UNLINK LE WIND,TIM,1,8,P8,MD31
571  PRIORITY     0,BU

```



```

572  TRANSFER  ,MD_E
575 MD31 UNLINK LE  BLOKS,OUT,1,8,P8,MD41
577  PRIORITY  0,BU
578  TRANSFER  ,MD31
579 MD41 TERMINATE
580 TIM  TEST E  F$TMR,1,TIM_F
581  PRIORITY  10
582  PREEMPT   TMR,PR,END_T,,RE
583  RETURN    TMR
584  PRIORITY  0
588  TEST G    CH$WIND,0,TIM_F
590  SPLIT     1,TIM_Z
592 TIM_F LEAVE  BUF1
594  TRANSFER  ,KLS
596 TIM_Z SEIZE  TMR
598  ADVANCE   X6
600  RELEASE   TMR
602  TRANSFER  ,REJ_T
604 END_T TERMINATE
606 ERR1 TERMINATE
608 ERR2 TRANSFER  ,MD_E
610 OUT  TERMINATE
612 REJ  TEST LE  P8,X8,ERR3
614 MD21 UNLINK LE  WIND,TRM,1,8,P8,MY11
616  PRIORITY  0,BU
618  TRANSFER  ,MD21
620 MY11 TEST NE  CH$WIND,0,TER1
622  TEST E    F$TMR,1,REJ_T
624  PRIORITY  10
626  PREEMPT   TMR,PR,ENS_T,,RE
628  RETURN    TMR
630  PRIORITY  0
632 REJ_T UNLINK  WIND,BL,ALL
634  PRIORITY  0,BU
636  TRANSFER  ,KLS
638 ENS_T TERMINATE
640 TRM  TEST E  F$TMR,1,REJ11
642  PRIORITY  10
644  PREEMPT   TMR,PR,EN_T,,RE
646  RETURN    TMR
648  PRIORITY  0
650 REJ11 LEAVE  BUF1
652  TERMINATE
654 EN_T TERMINATE
656 BL  LEAVE  BUF1
658  TRANSFER  ,MBF5
660 TER1 TEST E  R$BUF1,0,TSS
662  TERMINATE
664 TSS  TRANSFER  ,KLS
666 ERR3 TERMINATE
700 MOD2 ENTER  BUF2
710  ADVANCE   1

```

```

720     TEST E    P6,0,MOD25
721     TEST L    P8,X12,MOD26
722     TRANSFER  ,MOD21
724 MOD26 TEST E    X12,P8,MOD22
725     SAVEVALUE 12+,1
730     TEST NE   P2,X10,MOD24
731     LEAVE     BUF2
732     TERMINATE
740 MOD24 TABULATE TIME
750     SAVEVALUE 11+,P4
760     SAVEVALUE 14+,P4 ;received date in cycle
770     SAVEVALUE 10+,1
780     TEST E    P2,P3,MOD21
790     TABULATE  TIM_M
800     SAVEVALUE 10,1
810 MOD21 ASSIGN  7,1
820     TRANSFER  ,MOD23
830 MOD22 ASSIGN  7,0
835 MOD23 ASSIGN  8,V$HOM_R
840     TRANSFER  ,KAN2
842 MOD25 LEAVE   BUF2
844     TERMINATE
850 KAN2 SEIZE    DL2
860     LEAVE     BUF2
870     ADVANCE   V$TKC2
880     RELEASE   DL2
890     TRANSFER  V$P_ER2,,KAN21
892     ASSIGN    6,0
900     TRANSFER  ,KAN22
910 KAN21 ASSIGN  6,1
920 KAN22 ADVANCE X5
930     TRANSFER  ,MD1
932     GENERATE  V$CYCLE,,V$CYCLE
933     TABULATE  RATE
934     SAVEVALUE 14,0
935     TERMINATE
940     GENERATE  1,,X7
950     TERMINATE 1
960     WINDOW    TABLES
970     MICROWINDOW 1,AC1      ;MOD_TIME
980     MICROWINDOW 2,X9      ;MESSAGE
990     MICROWINDOW 3,X8      ;PACKET
1000    MICROWINDOW 4,X11     ;RECEIVE
1010    START     1

```

## Листинг имитационной модели работы №2

\* GPSS/PC Program File L2.GPS.

\* MODEL OF LAN

INPUT"СРЕДНЯЯ ИНТЕНСИВНОСТЬ", &AV1=1000

INPUT"ВРЕМЯ ПЕРЕДАЧИ", &AV2=20

```

INPUT"ЧИСЛО АБОНЕНТОВ", &AV3=10
INPUT"ВРЕМЯ МОДЕЛИРОВАНИЯ", &AV4=60000
*****

100 EXPON FUNCTION RN1,C24
0.,0./1.,104/.2.,222/.3.,355/.4.,509/.5.,69
.6.,915/.7,1.2/.75,1.38/.8,1.6/.84,1.83/.88,2.12
.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5
.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7.0/.9997,8.0
110 PAK FUNCTION RN2, D3
0.1,1.0/0.8,5.0/1.0,20.0
*****

120 INITIAL X1, &AV1
130 INITIAL X2, &AV2
140 INITIAL X3,4
150 INITIAL X4, &AV3
160 INITIAL X5,0
180 LKF VARIABLE X1/X4
182 LKZ VARIABLE Q$SUM/X4
190 TLAN TABLE M1,0,100,20
*****

*
SORCE
200 GENERATE V$LKF, FN$EXPON
201 AMD ASSIGN 2, X5
202 ASSIGN 2+,1
203 SAVEVALUE 5, P2
204 TEST LE P2, X4,NCYCL
205 SPLIT FN$PAK, KAN
210 TERMINATE
215 KAN QUEUE SUM
216 MARK
217 LINK P2, FIFO
218 NCYCL SAVEVALUE 5,0
219 TRANSFER ,AMD
*
MARKER
500 GENERATE 1,,, 1
501 ASSIGN 1, X4
502 MST ASSIGN 2,1
505 NEXT UNLINK P2, OUT, 1,,, NODT
510 ADVANCE X2
515 ADVANCE X3
519 NST ASSIGN 2+,1
520 TEST LE P2, P1, MST
522 TRANSFER ,NEXT
551 NODT ADVANCE X3
552 TRANSFER ,NST
560 OUT SEIZE LAN
561 ADVANCE X2
562 RELEASE LAN
564 DEPART SUM
565 TABULATE TLAN
566 TERMINATE
*
TIME

```

```

900    GENERATE    &AV4
901    TERMINATE   1
*****
960    WINDOW     TABLES
970    MICROWINDOW 1, AC1          ; MOD_TIME
980    MICROWINDOW 2, FR$LAN      ; FR_LAN
990    MICROWINDOW 3, V$LKZ       ; PC_BUF
1000   MICROWINDOW 4, FC$LAN      ; PAKETS
      START      1

```

### Листинг имитационной модели работы №3

```

* GPSS/PC Program File L3.GPS.
*      MODEL OF CSMA/CD LAN
      INPUT"СРЕДНЯЯ ИНТЕНСИВНОСТЬ",&AV1=200
      INPUT"ЧИСЛО АБОНЕНТОВ",&AV2=10
      INPUT"ВРЕМЯ ПЕРЕДАЧИ ПАКЕТА",&AV3=15
      INPUT"ИНТЕРВАЛ КОНФЛИКТОВ",&AV4=2
      INPUT"ВРЕМЯ МОДЕЛИРОВАНИЯ",&AV5=10000
*****
      TIN1 FUNCTION RN1,C2
0.,0./1.,2.
      TIN2 FUNCTION RN1,C2
0.,0./1.,4.
      TIN3 FUNCTION RN1,C2
0.,0./1.,8.
      TIN4 FUNCTION RN1,C2
0.,0./1.,16.
      TIN5 FUNCTION RN1,C2
0.,0./1.,32.
      TIN6 FUNCTION RN1,C2
0.,0./1.,64.
      TIN7 FUNCTION RN1,C2
0.,0./1.,128.
      TIN8 FUNCTION RN1,C2
0.,0./1.,256.
      TIN9 FUNCTION RN1,C2
0.,0./1.,512.
      TIN10 FUNCTION RN1,C2
0.,0./1.,1024.
      EXPON FUNCTION RN1,C24
0.,0./1.,104/2.,222/3.,355/4.,509/5.,69
.6.,915/7.,1.2/75,1.38/8,1.6/84,1.83/88,2.12
.9,2.3/92,2.52/94,2.81/95,2.99/96,3.2/97,3.5
.98,3.9/99,4.6/995,5.3/998,6.2/999,7.0/9997,8.0
*****
120    INITIAL    X1,&AV1
130    INITIAL    X2,&AV3
150    INITIAL    X4,&AV2
160    INITIAL    X5,0
170    INITIAL    X6,&AV4
180 LKF  VARIABLE  X1/X4

```

```

185 LKZ   FVARIABLE   QA$SUM#1000/X4
186 LKY   FVARIABLE   SC$BUS#1000/TC$TPAC
190 BUS   STORAGE     50
195 TPAC  TABLE      M1,0,50,20
*****
*          SORCE
200      GENERATE     V$LK,FN$EXPON
205 AMD   ASSIGN      2,X5
206      ASSIGN      2+,1
207      SAVEVALUE    5,P2
208      TEST LE     P2,X4,NCYCL
209      QUEUE        SUM
210      QUEUE        P2
211      SEIZE        P2
215      ASSIGN      1,1
220 BL1   TEST E      S$BUS,0,ZAD1
225      ADVANCE      X6
230      ENTER        BUS
232      BUFFER
235      TEST LE     S$BUS,1,OUT1
240      ADVANCE      X2,FN$EXPON
245      LEAVE        BUS
250      RELEASE      P2
255      DEPART       P2
256      DEPART       SUM
260      TABULATE     TPAC
261      SAVEVALUE    7,TC$TPAC
265      TERMINATE
270 ZAD1  ADVANCE     1
275      TRANSFER     ,BL1
280 OUT1  LEAVE        BUS
285      ASSIGN      1+,1
590 TS1   TEST E      P1,2,TS3
595      ADVANCE      FN$TIN1
600      TRANSFER     ,BL1
605 TS3   TEST E      P1,3,TS4
610      ADVANCE      FN$TIN2
615      TRANSFER     ,BL1
620 TS4   TEST E      P1,4,TS5
625      ADVANCE      FN$TIN3
630      TRANSFER     ,BL1
635 TS5   TEST E      P1,5,TS6
640      ADVANCE      FN$TIN4
645      TRANSFER     ,BL1
650 TS6   TEST E      P1,6,TS7
655      ADVANCE      FN$TIN5
660      TRANSFER     ,BL1
665 TS7   TEST E      P1,7,TS8
670      ADVANCE      FN$TIN6
675      TRANSFER     ,BL1
680 TS8   TEST E      P1,8,TS9
685      ADVANCE      FN$TIN7

```

```

690    TRANSFER    ,BL1
695 TS9    TEST E    P1,9,TS10
700    ADVANCE    FN$TIN8
705    TRANSFER    ,BL1
710 TS10    TEST E    P1,10,TS11
715    ADVANCE    FN$TIN9
720    TRANSFER    ,BL1
725 TS11    TEST LE    P1,16,TS12
730    ADVANCE    FN$TIN10
735    TRANSFER    ,BL1
740 TS12    SAVEVALUE    3+,1
745    RELEASE    P2
750    DEPART    P2
751    DEPART    SUM
755    TERMINATE

*****
800 NCYCL    SAVEVALUE    5,0
801    TRANSFER    ,AMD
*****
*          TIME
900    GENERATE    &AV5
901    SAVEVALUE    8,V$LKY
902    SAVEVALUE    9,V$LKZ
903    TERMINATE    1
*****

    WINDOW    TABLES
    MICROWINDOW    1,AC1            ;MOD_TIME
    MICROWINDOW    2,X7            ;PACKETS
    MICROWINDOW    3,SC$BUS        ;ATTEMPT
    MICROWINDOW    4,V$LKZ        ;AVER_Q
    START        1

```

## Листинг имитационной модели лабораторной работы №4

; GPSS/PC Program File L4SOI.GPS.

```

*****
**          MODEL OF TRANSPORT CONNECTION          *
    INPUT"Средний интервал для входных сегментов",&AV1=500
    INPUT"Средний интервал для нагрузки в ЦК",&AV2=7
    INPUT"Длина сегмента (байт)","&AV3=1024
    INPUT"Скорость абонентских каналов (бит/с)","&AV4=2400
    INPUT"Скорость магистральных каналов (бит/с)","&AV5=64000
    INPUT"Процент повторных передач в канале",&AV6=10
    INPUT"Срочная передача данных (да/нет = 1/0)","&AV8=0
    INPUT"Время прогона модели (1 ед. = 10 мс)","&AV7=20000
*****

    EXPON FUNCTION    RN1,C24
0.,0./1.,104/2.,222/3.,355/4.,509/5.,69
.6.,915/7.,1.2/75,1.38/8,1.6/84,1.83/88,2.12
.9,2.3/92,2.52/94,2.81/95,2.99/96,3.2/97,3.5
.98,3.9/99,4.6/995,5.3/998,6.2/999,7.0/9997,8.0
*****

```

```

      TIME TABLE      M1,0,250,20
110   INITIAL  X1,&AV1
111   INITIAL  X2,&AV2
114   INITIAL  X7,&AV7
115   INITIAL  X10,&AV3
116   INITIAL  X11,&AV4
117   INITIAL  X12,&AV5
118   INITIAL  X13,&AV6
130 ZKCA  FVARIABLE  X10#8#100/X11
131 ZKCM  FVARIABLE  X10#8#100/X12
132 PERS  VARIABLE   1000-X13#10
133      BBB                                     VARIABLE
Q$BUF1I+Q$BUF1O+Q$BUF2I+Q$BUF2O+Q$BUF3I+Q$BUF3O
*****
201   GENERATE  X1,FN$EXPON
202   ASSIGN    1,1
203   PRIORITY  &AV8
204   QUEUE     BUFA1
*      KAN1
210   SEIZE     KAN1
211   DEPART    BUFA1
212   TRANSFER  V$PERS,,MAIN1
215   ADVANCE   V$ZKCA
216 MAIN1 ADVANCE  V$ZKCA
220   RELEASE   KAN1
*      CC1
230 KOM1 QUEUE   BUF1I
235   SEIZE     CPU1
236   DEPART    BUF1I
240   ADVANCE   2
245   RELEASE   CPU1
250   TEST E    P1,1,OUT1
255 BF1O QUEUE   BUF1O
*      KAN2
300   SEIZE     KAN2
301   DEPART    BUF1O
302   TRANSFER  V$PERS,,MAIN2
305   ADVANCE   V$ZKCM
306 MAIN2 ADVANCE  V$ZKCM
310   RELEASE   KAN2
*      CC2
330 KOM2 QUEUE   BUF2I
335   SEIZE     CPU2
336   DEPART    BUF2I
340   ADVANCE   2
345   RELEASE   CPU2
350   TEST E    P1,1,OUT2
355 BF2O QUEUE   BUF2O
*      KAN3
400   SEIZE     KAN3
401   DEPART    BUF2O
402   TRANSFER  V$PERS,,MAIN3
405   ADVANCE   V$ZKCM

```

```

406 MAIN3 ADVANCE V$ZKCM
410 RELEASE KAN3
*      CC3
430 KOM3 QUEUE BUF3I
435 SEIZE CPU3
436 DEPART BUF3I
440 ADVANCE 2
445 RELEASE CPU3
450 TEST E P1,1,OUT3
455 BF3O QUEUE BUF3O
*      KAN4
500 SEIZE KAN4
501 DEPART BUF3O
502 TRANSFER V$PERS,,MAIN4
505 ADVANCE V$ZKCA
506 MAIN4 ADVANCE V$ZKCA
510 RELEASE KAN4
*
600 TABULATE TIME
601 SAVEVALUE 9+,1
610 TERMINATE
*
800 OUT1 TRANSFER .150,OUT11,OUT12
801 OUT11 TERMINATE
802 OUT12 TRANSFER ,BF1O
*
810 OUT2 TRANSFER .150,OUT21,OUT22
811 OUT21 TERMINATE
822 OUT22 TRANSFER ,BF2O
*
830 OUT3 SAVEVALUE 14,V$BBB
831 TERMINATE
*
900 GENERATE X2,FN$EXPON
901 ASSIGN 1,2
902 TRANSFER ,KOM1
910 GENERATE X2,FN$EXPON
911 ASSIGN 1,2
912 TRANSFER ,KOM2
920 GENERATE X2,FN$EXPON
921 ASSIGN 1,2
922 TRANSFER ,KOM3
*
940 GENERATE 1,,X7
950 TERMINATE 1
*****
960 WINDOW TABLES
970 MICROWINDOW 1,AC1 ;MOD_TIME
971 MICROWINDOW 2,X9 ;PACKETS
972 MICROWINDOW 3,FR$KAN1 ;FR_KAN1
973 MICROWINDOW 4,X14 ;QUEUES
*****
1010 START 1

```



## Список литературы

1. *Алексеев Е. Б.* Проектирование и техническая эксплуатация цифровых телекоммуникационных систем и сетей: Учебное пособие для вузов / Е. Б. Алексеев, В. Н. Гордиенко, В. В. Крухмалев. – 2-е изд., испр. – М. : Гор. линия-Телеком, 2012. – 392 с.
2. *Баринов, В. В.* Компьютерные сети: Учебник / В. В. Баринов, И. В. Баринов, А. В. Пролетарский. – М. : Academia, 2018. – 192 с.
3. *Гольдштейн, Б. С.* Сети связи пост-NGN / Б. С. Гольдштейн, А. Е. Кучерявый. – СПб. : БХВ-Петербург, 2014. – 160 с.
4. *Головин, Ю. А.* Методические указания к выполнению лабораторных работ по дисциплине «Сети ЭВМ и телекоммуникации». Часть 1 / Ю. А. Головин, О. И. Кутузов. – СПб. : ГЭТУ, 2006. – 14 с.
5. *Росляков, А. В.* Интернет вещей / А. В. Росляков, С. В. Ваяшин, А. Ю. Гребешков, М. Ю. Самсонов. – Самара : ПГУТИ, 2015. – 200 с.
6. *Росляков, А. В.* Будущие сети (Future networks) / А. В. Росляков, С. В. Ваяшин. – Самара : ПГУТИ, 2015. – 274 с.
7. *Крухмалев, В. В.* Цифровые системы передачи : Учебное пособие для вузов / В. В. Крухмалев, В. Н. Гордиенко, А. Д. Моченов. – 2-е изд., перераб. и доп. – М. : Гор. линия-Телеком, 2012. – 376 с.
8. *Куроуз, Д.* Компьютерные сети. Нисходящий подход / Д. Куроуз, К. Росс. – М. : Эксмо, 2016. – 912 с.
9. *Кутузов, О. И.* Инфокоммуникационные сети. Моделирование и оценка вероятностно-временных характеристик / О. И. Кутузов, Т. М. Татарникова. — СПб. : ГУАП, 2015.
10. *Кутузов, О. И.* Коммутаторы в корпоративных сетях. Моделирование и расчет / О. И. Кутузов, В. Г. Сергеев, Т. М. Татарникова – СПб. : Судостроение, 2003. – 170 с.
11. *Кутузов, О. И.* Основы проводной и радио связи. Конспект лекций, ЛЭТИ / О. И. Кутузов, В. В. Цехановский. – Л. : 1978. – 67 с.
12. *Кутузов, О. И.* Имитационное моделирование телекоммуникационных сетей. Учебное пособие с грифом УМО по спец. 071900 «Информационные системы и технологии» / О. И. Кутузов, Т. М. Татарникова. – СПб. : ГУТ, 2001. – 76 с.
13. *Кутузов, О. И.* Моделирование систем и сетей телекоммуникаций. Учебное пособие / О. И. Кутузов, Т. М. Татарникова. – СПб. : Изд-во РГГМУ, 2012. – 136 с.
14. *Кутузов, О. И.* Математические схемы и алгоритмы моделирования инфокоммуникационных систем : Учебное пособие / О. И. Кутузов, Т. М. Татарникова. – СПб. : ГУАП 2013. – 148 с.
15. *Кутузов, О. И.* Инфокоммуникационные сети. Моделирование и оценка вероятностно-временных характеристик: монография / О. И. Кутузов, Т. М. Татарникова. – СПб. ГУАП, 2015. – 382 с.
16. *Кутузов, О. И.* Модели и протоколы взаимодействия в информационных сетях : Учеб. пособие. – СПб. : Изд-во СПбГЭТУ «ЛЭТИ», 2014. – 114 с.

17. *Кутузов, О. И.* Моделирование систем. Методы и модели ускоренной имитации в задачах телекоммуникационных и транспортных сетей : Учебное пособие. – СПб. : Лань, 2018. – 132 с.
18. *Максимов, Н. В.* Компьютерные сети : Учебное пособие / Н. В. Максимов, И. И. Попов. – М. : Форум, 2017. – 320 с.
19. *Новожилов, Е. О.* Компьютерные сети: Учебное пособие. – М. : Academia, 2017. – 288 с.
20. *Олифер, В. Г.* Компьютерные сети. Принципы, технологии, протоколы: учеб. для вузов / В. Г. Олифер, Н. А. Олифер. – СПб. : Питер, 2016. – 318 с.
21. *Таненбаум, Э.* Компьютерные сети / Э. Таненбаум, Д. Уэзеролл. – СПб. : Питер, 2019. – 960 с.
22. *Татарникова, Т. М.* Защита информации в корпоративных вычислительных сетях : учеб. пособие. – СПб. : РГГМУ, 2011. – 107 с.
23. *Шахнович, И. В.* Современные технологии беспроводной связи. – М. : Техносфера, 2006. – 287 с.
24. *Шелухин, О. И.* Обнаружение вторжений в компьютерные сети (сетевые аномалии). – М. : ГЛТ, 2013. – 220 с.
25. *Чеппел, Л.* ТСР/ПР. Учебный курс / Л. Чеппел, Э. Титтел ; пер. с англ. Ю. Гороховский. – СПб. : БХВ-Петербург, 2003. – 976 с.
26. Современные сетевые технологии в телекоммуникационных системах / А. В. Боговик, Н. А. Зюзин, В. А. Керко [и др.] ; под общ. ред. проф. А. А. Сикарева. – СПб. : СПбГУВК, 2008. – 477 с.

*Олег Иванович КУТУЗОВ,  
Татьяна Михайловна ТАТАРНИКОВА,  
Владислав Владимирович ЦЕХАНОВСКИЙ*  
**ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ И СЕТИ**  
*У ч е б н и к*

Зав. редакцией  
литературы по информационным технологиям  
и системам связи *О. Е. Гайнутдинова*

ЛР № 065466 от 21.10.97  
Гигиенический сертификат 78.01.10.953.П.1028  
от 14.04.2016 г., выдан ЦГСЭН в СПб

**Издательство «ЛАНЬ»**  
lan@lanbook.ru; www.lanbook.com  
196105, Санкт-Петербург, пр. Юрия Гагарина, д.1, лит. А.  
Тел.: (812) 336-25-09, 412-92-72.  
Бесплатный звонок по России: 8-800-700-40-71

Подписано в печать 13.07.20.  
Бумага офсетная. Гарнитура Школьная. Формат 70×100<sup>1</sup>/<sub>16</sub>.  
Печать офсетная. Усл. п. л. 19,83. Тираж 30 экз.

Заказ № 716-20.

Отпечатано в полном соответствии  
с качеством предоставленного оригинал-макета  
в АО «Т8 Издательские технологии»  
109316, г. Москва, Волгоградский пр., д. 42, к. 5.