

Издательство «ПРЕСС-HERSON»

Локальные вычислительные сети

Издание второе, переработанное и дополненное

*Допущено учебно-методическим объединением
по образованию в области коммерции
в качестве учебного пособия для студентов
высших учебных заведений, обучающихся
по специальностям:*

060400 «Финансы и кредит»;

060500 «Бухгалтерский учет, анализ и аудит»;

351400 «Прикладная информатика (в экономике)»

080301 (351300) «Коммерция (торговое дело)»;

080111 (061500) «Маркетинг»;

032401 (350700) «Реклама»



Москва, 2009

УДК 681.142.2
ББК 32.97
Ч37

Чекмарев Ю. В.
Ч37 Локальные вычислительные сети. Издание второе, исправленное и дополненное. – М.: ДМК Пресс, 2009. – 200 с. : ил.

ISBN 978-5-94074-460-3

В издании описываются вопросы организации локальных вычислительных сетей (ЛВС), а также задачи, решаемые аппаратными и программными средствами локальной сети. Показаны используемые в ЛВС протоколы передачи данных, сетевые организационные системы, распределенные базы данных и методы администрирования ЛВС. Даны понятия сети Инtranет и корпоративных информационных приложений.

Учебное пособие предназначено для студентов нетехнических высших учебных заведений, обучающихся по экономическим и другим специальностям.

УДК 681.142.2
ББК 32.97

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-94074-460-3

© Чекмарев Ю. В., 2008
© Оформление, ДМК Пресс, 2009

Содержание

1. ЛВС как основа информационных систем	6
1.1. Состав и структура вычислительных сетей	6
1.2. Эволюция развития информационно-вычислительных систем	11
1.3. Архитектура компьютерных сетей	13
1.4. Состав и основные функции элементов ЛВС	16
2. Принципы построения ЛВС	20
2.1. Организационные аспекты и компоненты ЛВС	20
2.2. Основные характеристики и классификация ЛВС	23
2.3. Сетевые операционные системы ЛВС	27
2.4. Эталонная модель взаимодействия открытых систем (ВОС)	28
2.5. Уровневые услуги ВОС в ЛВС	34
2.6. Протокол сети передачи данных X.25	38
2.7. Методы доступа к передающей среде в ЛВС	39
3. Топология и способы организации ЛВС	43
3.1. Топологии ЛВС	43
3.2. Средства реализации ЛВС	47
3.3. Модем ЛВС	56
3.4. Способы организации ЛВС	58
3.5. Методы доступа в ЛВС	62
3.6. Централизованные и одноранговые ЛВС	66
3.7. Построение ЛВС на базе ОС Windows	68
4. Система протоколов TCP/IP	72
4.1. Назначение и организация стека TCP/IP	72
4.2. Многоуровневая структура стека TCP/IP	75
4.3. Сетезависимые и сетезависимые уровни стека TCP/IP	80

4.4. Адресация в IP-сетях. Типы адресов стека TCP/ IP	83
4.5. Формы записи IP-адресов	84
4.6. Разрешение локального IP-адреса	90
4.7. Служба доменных имен	92
4.8. Протокол межсетевое взаимодействия IP	94
4.9. Протокол доставки сообщений TCP	96

5. Организация и администрирование ЛВС 101

5.1. Задачи и методы администрирования ЛВС	101
5.2. Основные цели управления ЛВС	104
5.3. Основные принципы управления ЛВС	107
5.4. Многоуровневое представление задач управления	109
5.5. Средства и протоколы управления ЛВС	114
5.6. Службы безопасности ЛВС	116
5.7. Организация одноранговой сети в среде «Microsoft Windows»	122
5.8. Организация работы в сети Novel NetWare	126

6. Корпоративные информационные приложения 132

6.1. Корпоративно-функциональные компоненты информационной системы	132
6.2. Классификация корпоративных сетей	136
6.3. Программное обеспечение корпоративных сетей	142
6.4. Корпоративные информационные приложения	145

7. Сети интранет 148

7.1. Назначение сети интранет	148
7.2. Архитектура интранет	151
7.3. Средства построения сети интранет	152
7.4. Обзор языков программирования в интранет	157
7.5. Интегрированный язык HTML в интранет	163
7.6. Связь интранет с Интернетом	165
7.7. Услуги глобальной сети Интернет	168
7.8. Протоколы модуля сетевого управления	170

8. Распределенные базы данных 177

8.1. Понятия базы данных и базы знаний	177
8.2. Система управления базой данных	180
8.3. Распределенная система управления базой данных	183
8.4. Классы СУБД	188
8.5. Обеспечение целостности данных на уровне базы данных	189
8.6. Методы доступа к данным, находящимся в базах	192
8.7. Инструментальные средства СУБД	195

ЛВС как основа информационных систем



1.1. Состав и структура вычислительных сетей

Сеть (*network*) – взаимодействующая совокупность объектов, образуемых устройствами передачи и обработки данных. Различают два понятия сети: *коммуникационная сеть* и *информационная сеть*.

Первая в основном предназначена для передачи данных и, кроме этого, обеспечивает дополнительный сервис (VAS – Value Added Service). Более того, она нередко выполняет и задачи, связанные с преобразованием данных. Например, сборку потоков символов в пакеты и разборку пакетов на потоки символов. Благодаря интеграции обработки и передачи данных строятся интеллектуальные сети. Сети объединяются друг с другом, образуя ассоциации. Коммуникативные сети различаются по типу используемых физических средств соединения.

Информационная сеть создается подключением к коммуникационной сети абонентских систем. При этом на базе коммуникационной сети может быть построена не только одна, но и группа информационных сетей (рис. 1).

Под **системой** понимают любой объект, который одновременно рассматривается и как единое целое, и как объединенная в интересах достижения поставленной цели совокупность разнородных элементов.

Системы значительно отличаются между собой как по составу, так и по главным целям.

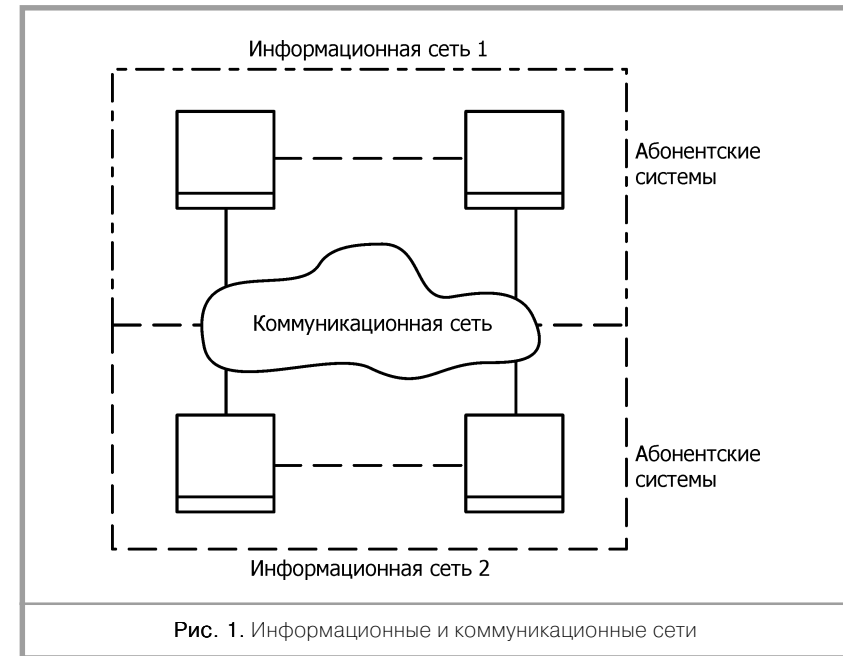


Рис. 1. Информационные и коммуникационные сети

Информационная система – человеко-компьютерная система для поддержки принятия решения и производства информационных продуктов, использующая компьютерную информационную технологию.

Расширение среды использования вычислительной техники влечет за собой необходимость постоянного повышения производительности и расширения функциональных возможностей компьютеров, которые по сути дела превратились в сложные вычислительные системы.

Под **вычислительной системой (ВС)** понимается совокупность взаимосвязанных и взаимодействующих процессоров или ЭВМ, периферийного оборудования и программного обеспечения, предназначенная для сбора, хранения, обработки и распределения информации. Естественно, вычислительная система должна оставаться интерактивной, то есть обеспечивать каждому пользователю возможность оперативного взаимодействия с системой на всех этапах решения задач.

Создание ВС преследует следующие основные цели:

- повышение производительности системы за счет ускорения процессов обработки данных;

- повышение надежности и достоверности вычислений;
- предоставление пользователям дополнительных сервисных услуг.

Основные принципы построения, закладываемые при создании ВС:

- возможность работы в разных режимах;
- модульность структуры технических и программных средств, что позволяет совершенствовать и модернизировать вычислительные системы без коренных переделок;
- унификация и стандартизация технических и программных решений;
- иерархия в организации управления процессами;
- способность систем к адаптации, самонастройке и самоорганизации;
- обеспечение необходимым сервисом пользователей при выполнении вычислений.

Структура ВС – совокупность комплексированных элементов и их связей. В качестве элементов ВС выступают отдельные ЭВМ и процессоры.

Классифицируют ВС по:

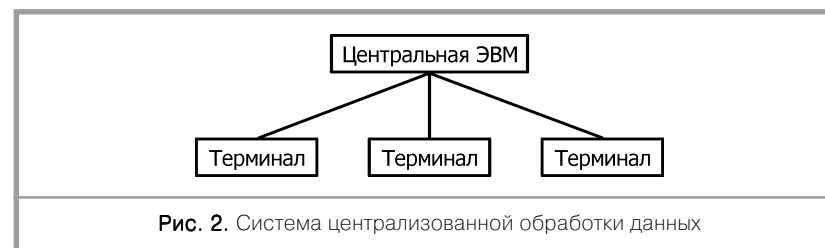
- целевому назначению и выполняемым функциям;
- типам и шагу ЭВМ или процессоров;
- архитектуре системы;
- режимам работы;
- методам управления элементами системы;
- степени разобщенности элементов ВС.

По назначению ВС делят на **универсальные** и **специализированные**. Универсальные ВС предназначены для решения самых различных задач. Специализированные системы ориентированы на решение узкого класса задач.

По типу ВС можно разделить на **многомашинные** и **многопроцессорные**.

В эпоху централизованного использования ЭВМ с пакетной обработкой информации пользователи предпочитали приобретать компьютеры, на которых можно было бы решать почти все классы их задач. Однако принцип централизованной обработки данных не отвечал всем требованиям надежности процесса обработки, затруднял развитие систем и не мог обеспечить необходимыми временными параметрами при диалоговой обработке данных в многопользовательском решении (рис. 2).

Кратковременный выход из строя центральной ЭВМ приводит к негативным последствиям для системы в целом, так как приходится дублировать функции центральной ЭВМ.



Многомашинные вычислительные системы (ММС) появились исторически первыми. Уже при использовании ЭВМ первых поколений возникали задачи повышения производительности, надежности и достоверности вычислений. Для этих целей используется комплекс машин, схематически показанный на рис. 3а.

Положения 1 и 3 электронного ключа (ЭК) обеспечивают режим повышенной надежности. При этом одна из машин выполняет вычисления, а другая находится в «горячем» или «холодном» резерве, то есть в готовности заменить основную ЭВМ. Положение 2 электронного ключа соответствует случаю, когда обе машины обеспечивают параллельный режим вычислений.

Здесь возможны две ситуации:

- обе машины решают одну и ту же задачу и периодически сверяют результаты решения. Тем самым обеспечивается режим повышенной достоверности, уменьшается вероятность появления ошибок в результатах вычисления;
- обе машины работают параллельно, но обрабатывают собственные потоки заданий. Возможность обмена информацией между машинами сохраняется. Этот вид работы относится к режиму повышенной производительности.

Основные отличия ММС заключаются в организации связи и обмена информацией между ЭВМ комплекса. Каждая из них сохраняет возможность автономной работы и управляется собственной операционной системой (ОС).

Многопроцессорные вычислительные системы (МПС) строятся при комплексировании нескольких процессоров (рис. 3, б). В качестве общего ресурса они имеют общую оперативную память (ООП). Параллельная работа процессоров и использование ООП обеспечиваются под управлением единой общей информационной системы. По сравнению с ММС здесь достигается наивысшая оперативность взаимодействия вычислителей-процессоров.

По типу ЭВМ или процессоров, используемых для построения ВС, различают однородные и неоднородные системы.

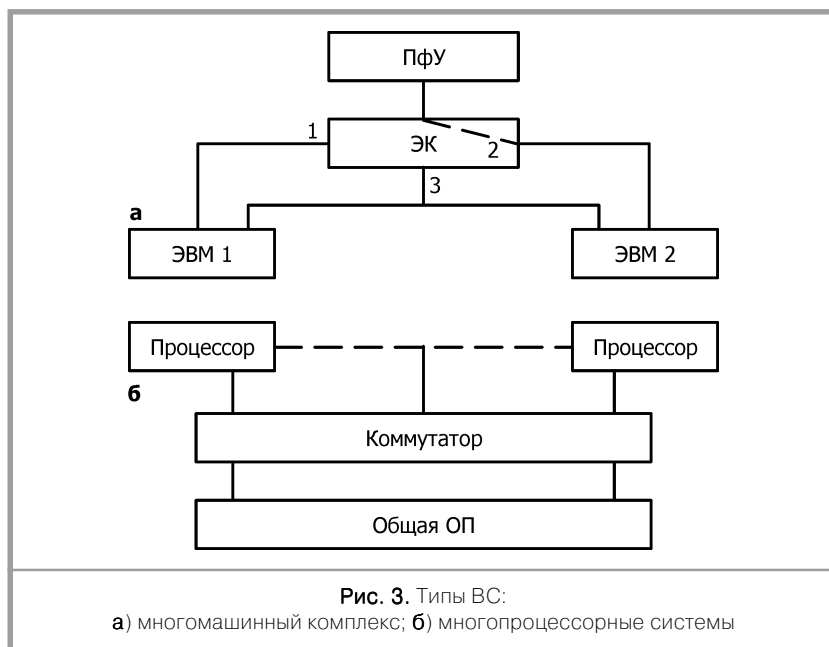


Рис. 3. Типы ВС:

а) многомашинный комплекс; б) многопроцессорные системы

По степени территориальной разобщенности вычислительных модулей ВС делятся на системы совмещенного (сосредоточенного) и распределенного (разобщенного – рис. 4) типов. Обычно такое деление касается только ММС. Многопроцессорные системы относятся к системам совмещенного типа. Совмещенные и распределен-

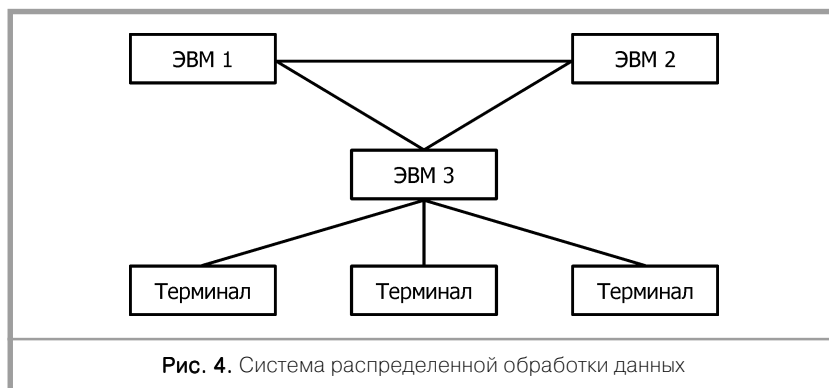


Рис. 4. Система распределенной обработки данных

ные ММС сильно различаются по оперативности взаимодействия в зависимости от удаленности ЭВМ.

1.2. Эволюция развития информационно-вычислительных систем

Компьютерные сети, называемые также вычислительными сетями, или сетями передачи данных, являются логическим результатом эволюции двух важнейших научно-технических отраслей современной цивилизации – компьютерных и телекоммуникационных технологий. С одной стороны, сети представляют собой частный случай распределенных вычислительных систем, в которых группа компьютеров согласованно выполняет набор взаимосвязанных задач, обмениваясь данными в автоматическом режиме. С другой – компьютерные сети могут рассматриваться как средство передачи информации на большие расстояния, для чего в них применяются методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах.

Таким образом, **компьютерная (вычислительная) сеть** – это совокупность компьютеров и терминалов, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных.

В силу ряда объективных причин становление и развитие компьютерных сетей шло по двум основным направлениям. Первое направление связано с развитием и совершенствованием систем телеобработки. Это направление в основном поддерживалось фирмами – производителями вычислительной техники. Так, наиболее характерным примером является сеть SNA (Systems Network Architecture – системная сетевая архитектура), разработанная фирмой IBM. По сути, данная сеть представляет множество систем телеобработки, объединенных между собой каналами передачи данных. В этом случае основная нагрузка по организации коммуникаций возлагается на процессоры телеобработки данных, сама же сеть передачи данных имеет относительно простую структуру.

Второе направление: компьютерная сеть рассматривается как сеть передачи данных, абонентами которой являются компьютеры. В этом случае основное внимание уделяется организации сети передачи данных на основе существующих сетей связи общего пользования. Постепенно эти два направления стали сближаться, и в настоящее время компьютерные сети можно рассматривать как результат объединения систем телеобработки на основе развитой сети передачи данных, что позволило получить качественно новые возможности в сфере информатики.

Таким образом, компьютерные сети позволяют решать такие качественно новые проблемные задачи, как, например:

- обеспечение распределенной обработки данных и параллельной обработки многими ЭВМ;
- возможность создания распределенной базы данных (РБД), размещаемой в памяти различных ЭВМ;
- возможность обмена большими массивами информации между ЭВМ, удаленными друг от друга на различные расстояния;
- коллективное использование дорогостоящих ресурсов: прикладных программных продуктов (ППП), баз данных (БД), баз знаний (БЗ), печатающих устройств (ПУ), сетевых операционных систем (ОС);
- предоставление большего перечня услуг, в том числе таких, как электронная почта (ЭП), телеконференции, электронные доски объявлений (ЭДО), дистанционное обучение, организация безбумажного документооборота, электронная подпись, принятие управленческих решений;
- повышение эффективности использования средств вычислительной техники и информатики (СВТИ) за счет более интенсивной и равномерной их загрузки, а также надежности обслуживания запросов пользователей;
- возможность оперативного перераспределения вычислительных мощностей между пользователями сети в зависимости от изменения их потребностей, а также резервирование этих мощностей и средств передачи данных на случай выхода из строя отдельных элементов сети;
- сокращение расходов на приобретение и эксплуатацию СВТИ (за счет коллективного их использования);
- облегчение работ по совершенствованию технических, программных и информационных средств.

Компьютерные сети являются высшей формой многомашинных ассоциаций. Основные отличия компьютерных сетей от многомашинного вычислительного комплекса следующие:

- размерность, то есть большое количество ЭВМ (от десятка до нескольких сотен), расположенных на расстоянии друг от друга от десятков метров до нескольких сотен и даже тысяч километров;
- разделение функций ЭВМ, то есть обработка данных и управление системой, анализ и хранение информации распределены между различными ЭВМ сети;
- необходимость решения в сети задачи маршрутизации сообщений, то есть сообщение от одной ЭВМ к другой в сети может быть передано по различным маршрутам в зависимости от приоритета и состояния каналов связи, соединяющих ЭВМ друг с другом.

1.3. Архитектура компьютерных сетей

Как и все сложные системы, компьютерные сети характеризуются определенными, присущими только им принципами организации.

Эти вопросы рассматриваются в рамках архитектуры, которая определяет общие принципы построения, топологию, функциональные характеристики системы. В частности, архитектура компьютерных сетей охватывает вопросы организации логической и физической структуры (топологии) сети, структурную организацию аппаратных и программных средств, правила (протоколы) их взаимодействия. В компьютерных сетях широко используется многоуровневый принцип структурной организации, при котором все множество сетевых функций распределяется по определенным уровням. При этом взаимодействие между уровнями осуществляется стандартным образом, что обеспечивает определенную независимость функций, принадлежащих различным уровням. В первую очередь это необходимо для реализации принципа открытости вычислительных сетей, являющегося неотъемлемой частью современных сложных систем.

По функциональному признаку все множество систем компьютерной сети можно разделить на абонентские, коммутационные и главные (Host) системы.

Абонентская система представляет собой компьютер, ориентированный на работу в составе компьютерной сети и обеспечивающий пользователям доступ к ее вычислительным ресурсам.

Коммутационные системы являются узлами коммутации сети передачи данных и обеспечивают организацию каналов передачи данных между элементами системы. В качестве управляющих элементов узлов коммутации используются процессоры телеобработки или специальные коммутационные (сетевые) процессоры.

Большим разнообразием отличаются **главные** (Host) системы, или сетевые серверы.

Сервером принято называть специальный компьютер, выполняющий основные сервисные функции: управление сетью, сбор, обработку, хранение и предоставление информации абонентам компьютерной сети.

В зависимости от территориальной рассредоточенности абонентских систем компьютерные (вычислительные) сети разделяют на три основных класса:

- глобальные сети (WAN – Wide Area Network);
- региональные сети (MAN – Metropolitan Area Network);
- локальные сети (LAN – Local Area Network).

Глобальная вычислительная сеть (ГВС) объединяет абонентские системы, рассредоточенные на большой территории, охваты-

вающие различные страны и континенты. ГВС решают проблему объединения информационных ресурсов всего человечества и организации доступа к ним. Взаимодействие абонентских систем (АС) осуществляется на базе различных территориальных сетей связи, в которых используются телефонные линии связи, радиосвязь, системы спутниковой связи.

Региональная вычислительная сеть (РВС) объединяет абонентские системы, расположенные друг от друга на значительном расстоянии: в пределах отдельной страны, региона, большого города.

Локальная вычислительная сеть (ЛВС) связывает абонентские системы, расположенные в пределах небольшой территории. К классу ЛВС относятся сети предприятий, фирм, банков, офисов, учебных заведений и т. д. Протяженность ЛВС ограничивается несколькими километрами.

Отдельный класс составляют **корпоративные вычислительные сети (КВС)**, или Intranet (интранет). Их также называют сетями масштаба предприятий (корпораций), что соответствует термину «enterprise – wide network». Им принадлежит ведущая роль в реализации задач планирования, организации и осуществления производственно-хозяйственной деятельности корпорации.

Другими словами, Intranet – это версия Интернета на уровне компании, адаптация некоторых технологий, созданных для Интернета, применительно к частным локальным (LAN) и глобальным (WAN) сетям организаций.

Корпоративную сеть можно рассматривать как модель группового сотрудничества, вариант решения прикладного программного обеспечения для рабочих групп, основанного на открытых стандартах Интернета. Она основана на технологии «клиент–сервер», то есть сетевое приложение делится на стороны: клиента, запрашивающего данные или услуги, и сервера, обслуживающего запросы клиента.

Типовая структура КВС приведена на рис. 5. Здесь выделено оборудование сети, размещенное в центральном офисе корпорации и в ее региональных отделениях. В центральном офисе (ЦОФ) имеются локальная сеть и учрежденческая автоматическая телефонная станция (УАТС) с подключением к ней телефонными аппаратами (Т). Через мультиплексор-коммутатор и модемы КДС и УАТС имеют выход на территориальную сеть связи (ТСС) типа Fram Relay или X.25, где используются выделенные телефонные линии связи. Такое же оборудование имеется в каждом региональном отделении (РО-1, ..., РО-N). Удаленные персональные компьютеры (УПК) через сервер доступа и ТСС имеют прямую связь с ЛВС центрального офиса.

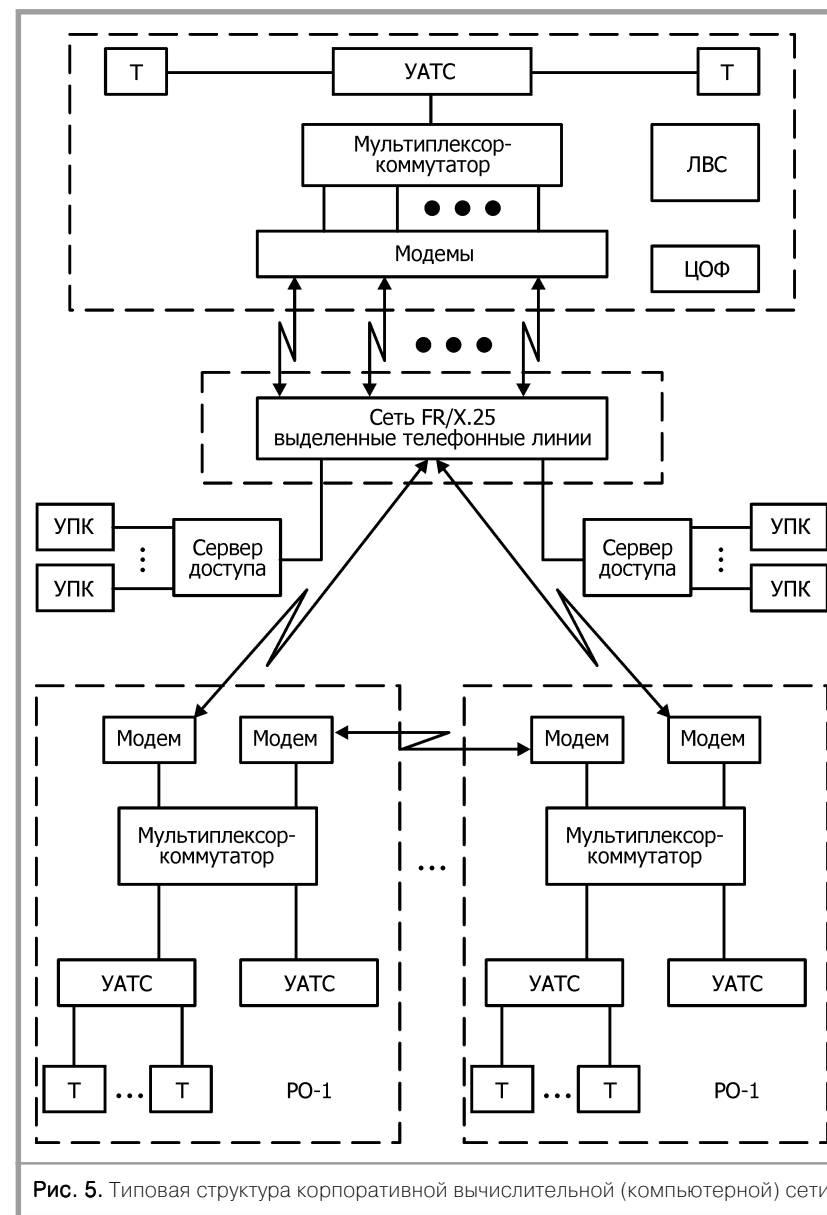


Рис. 5. Типовая структура корпоративной вычислительной (компьютерной) сети

Для установления Intranet необходимы следующие компоненты:

- компьютерная сеть для совместного использования ресурсов, или сеть взаимосвязанных ЛВС и УПК;
- сетевая операционная система, поддерживающая протокол TCP/IP (Unix, Windows NT, Netware, OS/2);
- компьютер-сервер, который может работать как сервер Интернета;
- программное обеспечение сервера, поддерживающее запросы браузеров в формате протокола передачи интерфейсных сообщений (HTTP – HyperText Transfer Protocol);
- служба глобального соединения (WWW). Для удобства редактирования объектов используется браузер объектов (Browser) – интегрированный отладчик, позволяющий выполнять пошаговую трассировку кода, задавать точки остановок (Break points);
- компьютеры-клиенты, на которых имеется сетевое программное обеспечение, позволяющее посылать и принимать пакетные данные по протоколу TCP/IP;
- программное обеспечение браузера для различных компьютеров-клиентов (Netscape Navigator, Microsoft Internet Explorer).

Эти требования к оборудованию и программному обеспечению Intranet дополняются требованиями к знанию технологии составления документов на языке написания гипертекста (HTML – HyperText Markup Language – гипертекстовый высокоуровневый язык).

1.4. Состав и основные функции элементов ЛВС

Изучение компьютерных сетей в целом предполагает знание принципов работы их отдельных элементов:

- компьютеров;
- коммуникационного оборудования;
- операционных систем;
- сетевых приложений.

Весь комплекс программно-аппаратных средств компьютерной (вычислительной) сети может быть описан многослойной моделью. В основе любой сети, в том числе и ЛВС, лежит аппаратный слой стандартизированных компьютерных платформ. В настоящее время в сетях широко и успешно применяются компьютеры различных классов – от персональных компьютеров до мэйнфреймов и супер-ЭВМ. Набор компьютеров в сети должен соответствовать набору разнообразных задач, решаемых сетью.

Второй слой – это коммуникационное оборудование. Хотя компьютеры и являются центральным элементом обработки данных в сетях, в последнее время не менее важную роль стали играть коммуникационные устройства. Кабельная система, повторители, мосты, коммутаторы, маршрутизаторы и модульные концентраторы из вспомогательных компонентов сети превратились в основные наряду с компьютерами и системным программным обеспечением как по влиянию на характеристики сети, так и по стоимости. Сегодня коммуникационное устройство может представлять собой сложный специализированный мультипроцессор, который нужно конфигурировать, оптимизировать и администрировать. Изучение принципов работы коммуникационного оборудования требует знакомства с большим количеством протоколов, используемых как в локальных, так и в глобальных сетях.

Третьим слоем, образующим программную платформу сети, являются операционные системы. **Операционная система (ОС)** (operating system) – комплекс программ, обеспечивающий в системе выполнение других программ, распределение ресурсов, планирование, ввод-вывод и управление данными. От того, какие концепции управления локальными и распределенными ресурсами положены в основу сетевой ОС, зависит эффективность работы всей сети. Сетевая операционная система (СОС) – network operating system (NOS) – комплекс программ, обеспечивающих в сети обработку, хранение и передачу данных. Сетевая операционная система (NOS) выполняет роль прикладной платформы, предоставляет разнообразные виды сетевых служб и поддерживает работу прикладных программ, реализуемых в прикладных системах.

NOS определяет взаимосвязанную группу протоколов верхних уровней, обеспечивающих основные функции сети. К ним в первую очередь относятся:

- адресация объектов сети;
- функционирование сетевых служб;
- обеспечение безопасности данных;
- управление сетью.

При проектировании сети важно учитывать, насколько просто данная ОС может взаимодействовать с другими ОС сети, насколько она обеспечивает безопасность и защищенность данных, до какой степени она позволяет наращивать число пользователей, можно ли ее перенести на компьютер другого типа и многие другие соображения.

Самым верхним слоем сетевых средств являются различные сетевые приложения, такие как сетевые базы данных, почтовые системы, средства архивирования данных, системы автоматизации коллективной работы и др. Очень важно представлять диапазон

возможностей, предоставляемых для различных областей применения, а также знать, насколько они совместимы с другими сетевыми приложениями и ОС.

Возможны следующие варианты структур СОС ЛВС:

- каждая ЭВМ сети реализует все функции СОС, то есть хранит в своей основной памяти резидентскую часть СОС и имеет доступ к любой нерезидентской части, хранящейся на внешних носителях;
- каждая ЭВМ сети имеет копии программ только части реализуемых функций СОС, копии программ редко реализуемых функций имеются в памяти только одной (или нескольких) ЭВМ;
- каждая ЭВМ сети выполняет только определенный набор функций СОС, причем этот набор является либо индивидуальным, либо некоторые функции будут общими для нескольких ЭВМ.

Различия в структурах СОС обусловлены принятыми способами управления ЛВС (децентрализованное или централизованное управление). Отличительной особенностью СОС ЛВС является наличие слоя операционных систем, обеспечивающего обмен информацией между ЭВМ сети.

В сетях с централизованным управлением СОС, называемая также ОС сервера, обеспечивает выполнение базовых функций, таких как поддержка файловой системы, планирование задач, управление памятью. СОС и ОС рабочей станции абонентской системы не совместимы, поэтому для обеспечения взаимодействия сервера и РС в рабочую программу вводится специальная программа, называемая сетевой оболочкой. Оболочка загружается в оперативную память РС как резидентная программа. Она воспринимает прикладные запросы пользователей сети и определяет место их обработки – в локальной ОС станции или с СОС на сервере.

В качестве сетевой оболочки ОС рабочей станции ЛВС используются следующие:

- сетевая оболочка Net Ware для взаимодействия с СОС Net Ware фирмы Novell;
- MS Windows фирмы Microsoft;
- X Windows;
- X.TreeNet предназначена для совместной работы с многопользовательской СОС с разделением времени Net Ware LAN фирмы Novell;
- программа Norton Commander.

Наиболее распространенными для ЛВС типа «клиент–сервер» являются следующие СОС:

- Net Ware фирмы Novell;
- LAN Server фирмы IBM;
- LAN Manager фирмы Microsoft;
- Vines фирмы Banyan на базе UNIX.

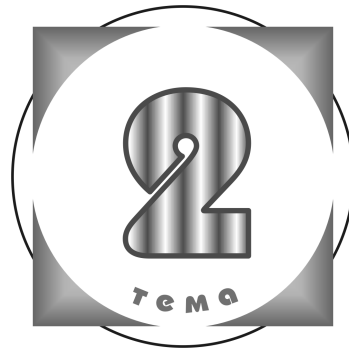
В последние годы широко используются СОС Windows NT, Windows 2000, Net Ware 5.

В сетях с децентрализованным управлением, или одноранговых сетях, объединяются компьютеры, каждый из которых может быть и сервером, и клиентом. В такой сети любой компьютер работает под управлением обычной дисковой ОС, а для выполнения сетевых функций в его оперативную память загружаются программы одноранговой СОС.

Для одноранговых ЛВС наиболее популярными СОС являются Net Ware Lite фирмы Novell и LAN-tastie фирмы Artisoft. Большинство этих систем, как и СОС для ЛВС с централизованным управлением, базируются на ОС ПЭВМ типа MS DOS, OS/2, Unix и Windows. Также в одноранговых ЛВС применяются СОС Windows for Workgroups, Personal NetWare, POWERLan.

СОС обеспечивают выполнение лишь общих функций ЛВС (поддержка файл-сервера, обеспечение многопользовательской работы, безопасности, секретности данных и т. д.), но они не могут самостоятельно реализовать многочисленные прикладные процессы. Например, не все СОС имеют собственные средства программирования электронной почты – одного из основных приложений ЛВС. Поэтому важным требованием к большинству современных пакетов прикладных программ (ППП) является их способность работать в условиях локальных сетей, то есть выполнять функции прикладных программ сети (ППС).

Принципы построения ЛВС



2.1. Организационные аспекты и компоненты ЛВС

Локальная сеть (LAN) – сеть, системы которой расположены на небольшом расстоянии друг от друга. Она охватывает небольшое пространство и характеризуется большими скоростями передачи данных. Важно то, что в локальной сети каналы имеют высокое качество и принадлежат одной организации.

Локальную вычислительную сеть (ЛВС) модно рассматривать как коммуникационную систему, которая поддерживает, как правило, в пределах одного здания или ограниченной территории (до 10 км), один или несколько высокоскоростных каналов передачи информации, предоставляемых подключаемым абонентским системам (АС) для кратковременного использования.

В обобщенной структуре ЛВС выделяются совокупность абонентских узлов или систем (их число может быть от десятков до сотен), серверов и коммутационная подсеть (КП).

Основными компонентами сети являются (рис. 6):

- физическая среда передачи данных (сетевой кабель);
- рабочие станции (АРМ пользователей сети) на базе компьютеров;
- платы интерфейса среды (сетевые адаптеры);
- серверы сети.

Рабочими станциями (РС) в ЛВС служат, как правило, персональные компьютеры (ПК). Из-за своей небольшой сложности и невысокой стоимости ПК используются при автоматизации коммерчес-

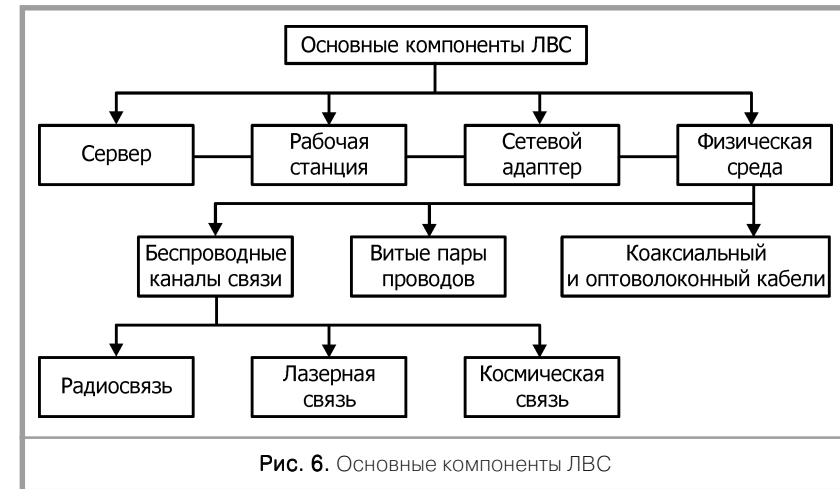


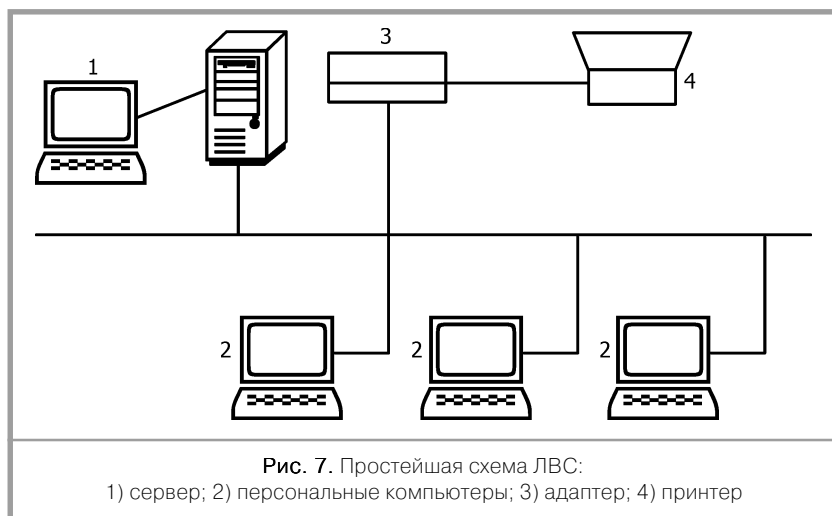
Рис. 6. Основные компоненты ЛВС

кой, бухгалтерской, банковской деятельности, а также для создания распределенных, управляющих и информационно-справочных систем. На РС пользователями сети реализуются прикладные задачи, выполнение которых связано с понятием вычислительного процесса.

Международный комитет IEEE 802 (институт инженеров по электронике и электротехнике, США), специализирующийся на стандартизации в области ЛВС, дает следующее определение этим системам: «ЛВС отличаются от других видов сетей тем, что обычно ограничены умеренной географической областью, такой как группа рядом стоящих зданий, и в зависимости от каналов связи осуществляют передачу данных в диапазонах скоростей от умеренных до высоких с низкой степенью ошибок» (рис. 7).

Структурно локальную сеть можно представить в виде множества абонентских систем, объединенных высокоскоростными каналами передачи данных.

Серверы сети – это аппаратно программные системы, выполняющие функции управления распределением сетевых ресурсов общего доступа, которые могут работать и как обычная абонентская система. В качестве аппаратной части сервера используются достаточно мощный ПК, мини-ЭВМ, большая ЭВМ или компьютер, спроектированный специально как сервер. В ЛВС может быть несколько различных серверов для управления сетевыми ресурсами, однако всегда имеется один (или более) файл-сервер (сервер без данных) для управления внешними запоминающими устройствами (ЗУ) общего доступа и организации распределенных баз данных (РБД).



Рабочие станции и серверы соединяются с кабелем коммуникационной подсети с помощью интерфейсных плат – сетевых адаптеров (СА). Основные функции СА: организация приема (передачи) данных из (в) РС, согласование скорости приема (передачи) информации (буферизация), формирование пакета данных, параллельно-последовательное преобразование (конвертирование), кодирование (декодирование) данных, проверка правильности передачи, установление соединения с требуемым абонентом сети, организация собственно обмена данными.

В ряде случаев перечень функций СА существенно увеличивается, и тогда они строятся на основе микропроцессоров и встроенных модемов.

В ЛВС в качестве физических передающих средств используются витая проволочная пара, коаксиальный и оптоволоконный кабель, радио, и оптические средства связи.

Кроме указанного выше в ЛВС используется следующее **сетевое оборудование**:

- приемопередатчики (трансиверы) и повторители (репитеры) – для объединения сегментов локальной сети с шинной технологией;
- концентраторы (хабы) – для формирования сети производительной технологии (используются активные и пассивные концентраторы);
- мосты – для объединения локальных сетей в единое целое и повышения производительности этого целого путем регули-

рования трафика (данных пользователя) между отдельными подсетями;

- маршрутизаторы и коммутаторы – для реализации функций коммутации и маршрутизации при управлении трафиком в сегментированных (состоящих из взаимосвязанных сегментов) сетях. В отличие от мостов, обеспечивающих сегментацию сети на физическом уровне, маршрутизаторы выполняют ряд «интеллектуальных» функций при управлении трафиком. Коммутаторы, выполняя практически те же функции, что и маршрутизаторы, превосходят их по производительности и обладают меньшей латентностью (аппаратная временная задержка между получением и пересылкой информации);
- модемы (модуляторы-демодуляторы) – для согласования цифровых сигналов, генерируемых компьютером, с аналоговыми сигналами типичной современной телефонной линии;
- анализаторы – для контроля качества функционирования сети;
- сетевые тестеры – для проверки кабелей и отыскания неисправностей в системе установленных кабелей.

2.2. Основные характеристики и классификация ЛВС

Основные **характеристики ЛВС**:

- территориальная протяженность сети (длина общего канала связи);
- максимальная скорость передачи данных;
- максимально возможное расстояние между рабочими станциями сети;
- топология сети;
- вид физической среды передачи данных;
- максимальное число каналов передачи данных;
- тип передачи сигналов (синхронный или асинхронный);
- метод доступа абонентов в сети;
- структура программного обеспечения сети;
- возможности передачи речи и видеосигнала;
- условия надежной работы сети;
- возможности связи ЛВС между собой и с сетью более высокого уровня;
- возможность использования процедуры установления приоритетов при одновременном подключении абонентов к общему каналу.

К наиболее типичным областям применения ЛВС относятся следующие:

- обработка текстов – одна из наиболее распространенных функций средств обработки информации, используемых в ЛВС. Передача и обработка информации в сети, развернутой на предприятии (в организации, вузе, банке и т. д.), обеспечивают реальный переход к «бесбумажной» технологии, вытесняя полностью и частично пишущие машинки;
- организация собственных информационных систем, содержащих собственные базы данных – индивидуальные и общие, сосредоточенные и распределенные. Такие базы данных могут быть в каждой организации или фирме;
- обмен информацией между абонентами сети – важное средство сокращения до минимума бумажного документооборота. Передача данных и связь занимают особое место среди приложений сети, так как это главное условие нормального функционирования современных организаций;
- обеспечение распределенной обработки данных, связанное с объединением АРМ всех специалистов данной организации в сеть. Несмотря на существенные различия в характере и объеме расчетов, проводимых на АРМ специалистами различного профиля, используемая при этом информация в рамках одной организации, как правило, находится в одной (интегрированной) базе данных. Поэтому объединение таких АРМ в сеть является целесообразным и весьма эффективным решением;
- поддержка принятия управленческих решений, предоставляющая руководителям и управленческому персоналу организации достоверную и оперативную информацию, необходимую для оценки ситуации и принятия оптимальных решений;
- организация электронной почты – один из видов услуг ЛВС, позволяющих руководителям и всем сотрудникам предприятия оперативно получать возможные сведения, необходимые в его производственно-хозяйственной, коммерческой и торговой деятельности;
- коллективное использование дорогостоящих ресурсов – необходимое условие снижения стоимости работ, выполняемых в порядке реализации вышеуказанных применений ЛВС. Речь идет о таких ресурсах, как высокоскоростные печатающие устройства, запоминающие устройства большой емкости, мощные средства обработки информации, прикладные программные системы, базы данных, базы знаний. Очевидно, что такие средства нецелесообразно (вследствие невысокого коэффициента использования и дороговизны) иметь каждой абонент-

ской системе сети. Достаточно, если в сети эти средства имеются в одном или нескольких экземплярах, но доступ к ним обеспечивается для всех АС.

В зависимости от характера деятельности организации, в которой развернуты одна или несколько ЛВС, указанные функции реализуются в определенной комбинации. Кроме того, могут выполняться и другие функции, специфические для данной организации.

Наличие высокоскоростных каналов передачи данных до недавнего времени являлось основной отличительной особенностью ЛВС. Первоначально скорость передачи в локальных сетях составляла от 1 до 16 Мбит/с, что было значительно больше скорости передачи в глобальных сетях. Современные ЛВС обеспечивают скорость передачи свыше 100 Мбит/с. Такая высокая скорость передачи информации в локальных сетях достигается еще и за счет максимального упрощения процедуры выбора маршрута, коммутации и промежуточного хранения информации в узлах сети.

ЛВС можно классифицировать по:

- назначению – делятся на управляющие (организационными, технологическими, административными и другими процессами), информационные (информационно-поисковые), расчетные, информационно-расчетные, обработки документальной информации и др.;
- типам используемых ЭВМ – делятся на однородные и неоднородные. Однородные содержат одинаковые модели и одинаковый состав абонентских средств, а неоднородные ЛВС содержат различные классы (микро-, мини-, большие) и модели (внутри классов) ЭВМ, а также различное абонентское оборудование;
- организации управления – однородные ЛВС в зависимости от наличия центральной абонентской системы делятся на две группы.

К первой группе относятся сети с *централизованным управлением*, когда выделяются одна или несколько машин (центральных систем или органов) для управления работой сети. Для таких сетей характерны обилие служебной информации и приоритетность подключаемых станций (по расположению или принятому приоритету). В общем случае ЛВС с централизованным управлением (необязательно на основе многоканала) имеет централизованный сервер (ЭВМ), управляющий работой сети.

Прикладной процесс центральной системы организует проведение сеансов, связанных с передачей данных, осуществляет диагностику сети, ведет статистику и учет работы. В ЛВС с мо-

ноканалом центральная система реализует также общую защиту от конфликтов, возникающих при перегрузке и сбоях в сети. При выходе из строя центральной системы вся ЛВС прекращает работу. Сети с централизованным управлением отличаются простотой обеспечения функций взаимодействия между отдельными ЭВМ и, как правило, характеризуются тем, что большая часть информационно-вычислительных ресурсов сосредоточивается в центральном сервисе.

Применение ЛВС с централизованным управлением целесообразно применять при небольшом числе абонентских систем. Если информационно-вычислительные ресурсы ЛВС равномерно распределены по большому числу абонентских систем, то централизованное управление малоэффективно, так как не обеспечивает требуемой надежности сети и приводит к резкому увеличению служебной (управляющей) информации.

Ко второй группе относятся сети с *децентрализованным*, или *распределенным, управлением*, или *одноранговые*. В таких сетях нет выделенных серверов, функции управления сетью распределены между отдельными серверами или рабочими станциями. Однако для проведения диагностики, сбора статистики и проведения других административных функций в сети используется специально выделенная абонентская система. К недостаткам одноранговых (децентрализованных) сетей следует отнести зависимость эффективности функционирования сети от количества АС, сложность управления сетью, сложность обеспечения защиты информации от несанкционированного доступа. Но при распределенном управлении каждый узел коммутации самостоятельно на основе хранящейся в нем управляющей информации определяет направление передачи пакетов (кадров), что приводит к увеличению сложности узлов коммутации, однако система обладает более высокой живучестью, так как выход из строя какого-либо узла коммутации не влечет за собой потери работоспособности всей сети;

- организации передачи информации – ЛВС делятся на сети с маршрутизацией и селекцией информации. Взаимодействие абонентских систем с маршрутизацией информации обеспечивается определением путей передачи блоков данных по адресам их назначения. Этот процесс выполняется всеми коммутационными системами, имеющимися в сети. При этом абонентские системы могут взаимодействовать по различным путям (маршрутам) передачи блоков данных, и для сокращения времени передачи осуществляется поиск кратчайшего по времени маршрута.

Под селекцией в ЛВС подразумевается процесс выбора очередной абонентской системы для подключения ее к сети передачи данных с целью обмена информацией;

- по топологическим признакам – ЛВС делятся на сети с произвольной, кольцевой, древовидной конфигурацией, сети типа «общая шина» (моноканал, звезда) и др.

2.3. Сетевые операционные системы ЛВС

Кроме топологии ЛВС, процесс передачи данных во многом определяется программным обеспечением ЭВМ абонентских систем, в основном их операционными системами, поскольку каждая из них поддерживает соответствующий метод доступа со стороны терминалов. Моноканал рассматривается как один из терминалов, поэтому надо знать, насколько различаются операционные системы и методы теледоступа всех абонентских комплексов, подключенных к сети. Различают ЛВС с единой операционной поддержкой и единичными методами теледоступа и ЛВС, использующие различные физические носители сигнала.

Следует иметь в виду, что сетевые операционные системы можно разделить на классы.

Класс системы клиент-сервер. Системный слой обычно распределен между классами процессоров-серверов, а пользовательский слой СОС размещается на процессорах клиентов-пользователей, то есть это способ взаимодействия компьютеров в локальной сети, при котором один из компьютеров (сервер) представляет свои ресурсы другому компьютеру (клиенту) (рис. 8).

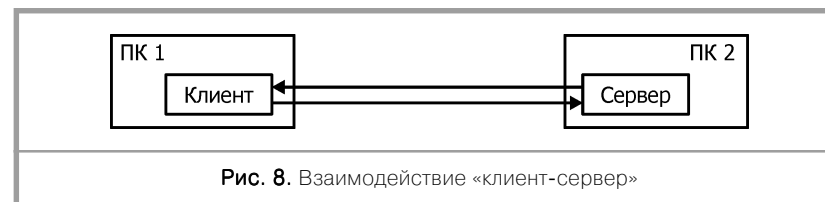


Рис. 8. Взаимодействие «клиент-сервер»

Класс системы автономных филиалов. Все слои обычно присутствуют на каждой машине. Ядром чаще является модифицированная ОС, разделенная по времени. Каждая система способна к независимой работе в случае, если она становится отсоединенной от сети. Ресурсы обычно разбиты на части, и каждой частью управляет одна машина.

Класс распределенных СОС. Общее ядро ОС выполняется на каждом процессоре сети, но другие функции СОС могут дублироваться и распределяться по процессорам сети. Единообразное именование ресурсов обычно является функцией ядра СОС, а не пользовательского слоя СОС.

Класс децентрализованных СОС, когда ресурсы необязательно распределяются на части, а распределяются сами функции ядра СОС и отсутствуют какие-либо центральные компоненты. Реализация рассредоточенных и взаимодействующих процессов в ЛВС осуществляется на основе двух концепций, одна из которых устанавливает связи между процессорами без функциональной среды между ними, а другая определяет связь только через функциональную среду.

В первом случае правильность понимания действий, происходящих в рамках соединяемых процессов взаимодействующих АС, обеспечивается соответствующими средствами теледоступа в составе СОС. Однако предусмотреть такие средства на все случаи соединения нереально. Поэтому взаимодействующие процессы в сетях соединяются с помощью функциональной среды, обеспечивающей выполнение определенного свода правил – протокола связи процессов. Обычно эти протоколы реализуются с учетом принципа пакетной коммутации, в соответствии с которым перед передачей сообщение разбивается на блоки – пакеты определенной длины. Каждый пакет представляет собой независимую единицу передачи информации, содержащую, кроме собственно данных, служебную информацию (адреса отправителя и получателя, номер пакета в сообщении, информацию для контроля правильности принятых данных).

2.4. Эталонная модель взаимодействия открытых систем (ВОС)

Практика создания и развития ВС привела к необходимости разработки стандартов по всему комплексу вопросов организации сетевых систем. В 1978 г. Международная организация по стандартизации (МОС) предложила семиуровневую эталонную модель взаимодействия открытых систем (ВОС), которая получила широкое распространение и признание. Данный стандарт определяет:

- понятия и основные термины, используемые при построении открытых систем;
- описание возможностей и набора конкретных услуг, которые должна предоставлять открытая система;
- логическую структуру открытых систем; протоколы, обеспечивающие услуги открытых систем.

В соответствии со стандартом 7498 открытой системой считается система, отвечающая требованиям эталонной модели взаимодействия открытых систем, реализующая стандартный набор услуг и поддерживаемая стандартными протоколами.

Основной задачей модели ВОС является описание множества функций, определяющих правил взаимодействия открытых систем. При этом широко используется понятие «процесс», определяемый как динамический объект, реализующий целенаправленный акт обработки информации.

Принято подразделять процессы на прикладные и системные. Прикладной процесс отождествляется с реализацией определенных процедур, связанных с переработкой информации при решении пользовательских задач. Системные процессы определяют выполнение вспомогательных функций, связанных с обеспечением прикладных процессов.

Ввод данных, необходимых процессу, и вывод данных производятся в форме сообщений через логические (программно-организованные) точки, называемые портами. Различают входные и выходные порты. Через входные порты осуществляется ввод данных для данного процесса, через выходные порты текущий процесс выдает результат обработки данных. Промежуток времени, в течение которого взаимодействуют процессы, называется сеансом обмена, или сессией.

Основу модели ВОС составляет концепция многоуровневой организации протоколов, которую можно рассматривать в качестве дальнейшего развития многоуровневой организации протоколов систем телеобработки. Существенной особенностью модели взаимодействия открытых систем являются разработка и использование единого подхода к организации протоколов и интерфейсов различных уровней. В соответствии с данной концепцией каждому уровню ставится в соответствие набор определенных функций, связанных с решением конкретной задачи по организации взаимодействия открытых систем. Нумерация уровней осуществляется относительно физических средств соединения, то есть первый номер присваивается физическому уровню, а наибольший номер – прикладному (пользовательскому) уровню. Каждый уровень с меньшим номером считается вспомогательным для смежного с ним более высокого уровня и предоставляет ему определенный набор услуг, называемых сервисом. Следует подчеркнуть, что эталонная модель не определяет средства реализации протоколов, а только специфицирует их. Таким образом, функции каждого уровня могут быть реализованы различными программными и аппаратными средствами. Основным условием при этом является то, что взаимо-

действие между любыми смежными уровнями должно быть четко определено, то есть должно осуществляться через точку доступа посредством стандартного межуровневого интерфейса.

Точка доступа является портом, в котором объект N-го уровня предоставляет услуги (N + 1)-му уровню. Это достаточно важное условие определяет возможность изменения системы в целом, что является одним из основных условий построения открытых систем.

Структурной единицей данных, передаваемых между уровнями, является так называемый протокольный блок данных, состоящий из управляющего поля, называемого заголовком, и поля данных. Заголовок N-го блока содержит управляющую информацию, формируемую на N-ом уровне. Содержимое поля данных N-го уровня представляет собой блок данных (N + 1)-го уровня. Таким образом, формируется вложенная структура, при которой протокольные блоки данных, начиная с верхнего уровня, вкладываются друг в друга. При передаче данных в обратном направлении происходит обратная процедура «распаковки» блоков.

При разработке эталонной модели число ее уровней определялось из следующих соображений:

- разбивка на уровне должна максимально отражать логическую структуру компьютерной сети;
- межуровневые границы должны быть определены таким образом, чтобы обеспечивались минимальное число и прирост межуровневых связей;
- большое количество уровней, с одной стороны, упрощает внесение изменений в систему, с другой – увеличивает количество межуровневых протоколов и затрудняет описание модели в целом.

С учетом вышеизложенного МОС была предложена 7-уровневая модель ВОС (рис. 9).

Основным с точки зрения пользователя является *прикладной* (седьмой) уровень, который обеспечивает поддержку прикладных процессов конечных пользователей и определяет семантику данных, то есть смысловое содержание информации, которой обмениваются открытые системы в процессе их взаимодействия. С этой целью данный уровень, кроме протоколов взаимодействия, прикладных процессов, поддерживает протоколы передачи файлов, виртуального терминала, электронной почты и им подобные.

Следующий (шестой) уровень – представительский (уровень представления данных) – определяет единый для всех открытых систем синтаксис (то есть представление данных) передаваемой информации. Необходимость данного уровня обусловлена различной формой представления информации в сети передачи данных и ком-

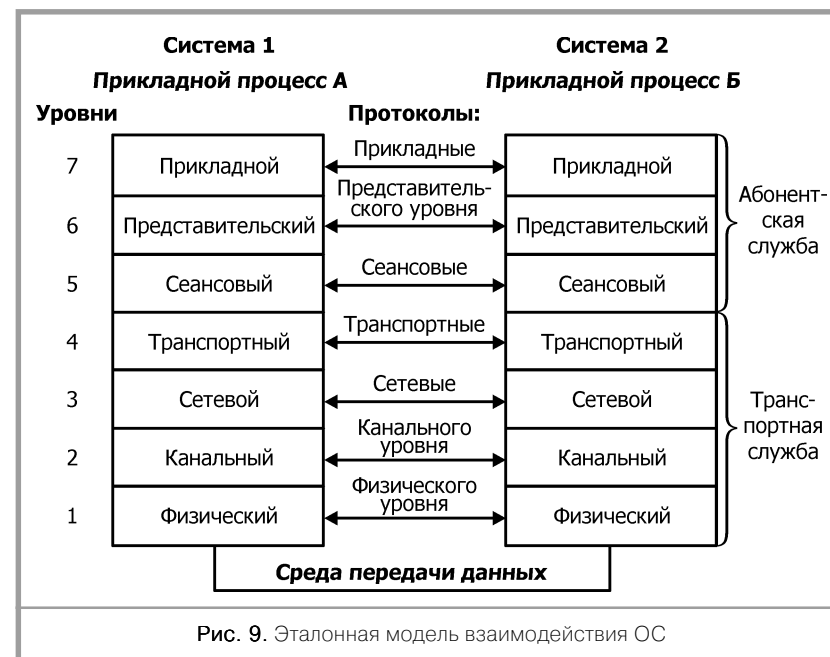


Рис. 9. Эталонная модель взаимодействия ОС

пьютерах. Данный уровень играет важную роль в обеспечении «открытости» систем, позволяя им общаться между собой независимо от их внутреннего языка, и гарантирует представление данных в кодах и форматах, принятых в данной системе.

Сеансовый (пятый) уровень реализует организацию сеансов связи между прикладными процессами, расположенными в различных абонентских системах. На данном уровне создаются порты для приема и передачи сообщений и организуются соединения – логические каналы между процессами. Необходимость протоколов данного уровня определяется относительной сложностью сети передачи данных и стремлением обеспечить достаточно высокую надежность передачи данных. Сеансовый уровень поддерживает и завершает сеанс связи.

Три верхних уровня объединяются под общим названием – процесс, или прикладной процесс. Эти уровни определяют функциональные особенности вычислительной сети как прикладной системы.

Транспортный (четвертый) уровень (уровень сквозной передачи) служит для обеспечения передачи данных между двумя взаимодействующими открытыми системами и организации процедуры

сопряжения абонентов сети с системой передачи данных. На этом уровне определяется взаимодействие абонентских систем – источника и адресата данных, организуется и поддерживается логический канал (транспортное соединение) между абонентами.

Сетевой (третий) уровень предназначен для обеспечения процессов маршрутизации информации и управления сетью передачи данных. Здесь решаются вопросы управления сетью передачи данных, в том числе маршрутизация и управление информационными потоками. Он также отвечает за информатизацию пакетов в коммутационной сети и за связь между сетями – реализует межсетевое взаимодействие.

Канальный (второй) уровень обеспечивает функциональные и процедурные средства для установления, поддержания и расторжения соединений на уровне каналов передачи данных. Процедуры канального уровня – обнаружение и, по возможности, исправление ошибок, возникающих на физическом уровне.

Физический (первый) уровень обеспечивает технические, электрические, функциональные и процедурные средства организации физических соединений при передаче данных между физическими объектами. Его основная задача – управление аппаратурой передачи данных и подключенным к ней каналом связи.

Четыре нижних уровня эталонной модели образуют транспортную службу компьютерной сети, которая обеспечивает передачу («транспортировку») информации между абонентскими системами, освобождая более высокие уровни от решения этих задач. В свою очередь, три верхних уровня, обеспечивающих логическое взаимодействие прикладных процессов, функционально объединяются в абонентскую службу.

Работы по совершенствованию эталонной модели ВОС для ЛВС привели к декомпозиции уровней 1 и 2 (рис. 10). Канальный уровень разделен на два подуровня: подуровень управления логическим каналом LLC (Local Link Control), то есть выполняет функции соответственно канального уровня (передача кадров между рабочими станциями, включая исправление ошибок, диагностику работоспособности узлов сети), и подуровень управления доступом к передающей среде MAC (Media Access Control), то есть реализация алгоритма доступа к среде и адресация станции сети.

Физический уровень делится на три подуровня: передачи физических сигналов, интерфейса с устройством доступа и подключения к физической среде.

При передаче информации от прикладного процесса в сеть происходит ее обработка уровнями модели ВОС (рис. 10). Смысл этой обработки заключается в том, что каждый уровень добавляет к ин-

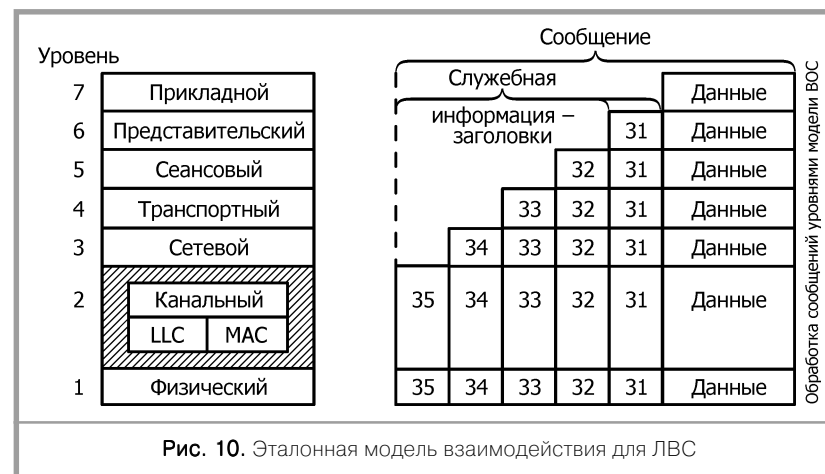


Рис. 10. Эталонная модель взаимодействия для ЛВС

формации процесса свой заголовок – служебную информацию, которая необходима для адресации сообщений и для некоторых контрольных функций. Канальный уровень, кроме заголовка, добавляет еще и концевик – контрольную последовательность, которая используется для проверки правильности приема сообщений из коммуникационной сети.

Физический уровень заголовка не добавляет. Сообщение, обрамленное заголовками и концевиком, уходит в коммуникационную сеть и поступает на абонентскую ЭВМ вычислительной сети. Каждая абонентская ЭВМ, принявшая сообщение, дешифрует адреса и определяет, предназначено ли ей данное сообщение. При этом в абонентской ЭВМ происходит обратный процесс – чтение и отсечение заголовков уровнями модели ВОС. Каждый уровень реагирует только на свой заголовок. Заголовки верхних уровней нижними уровнями не воспринимаются и не изменяются – они «прозрачны» для нижних уровней. Так, перемещаясь по уровням модели ВОС, информация наконец поступает к процессору, которому она была адресована.

В ЛВС процедуры управления на физическом, канальном и транспортном уровнях не отличаются сложностью, в связи с чем эти уровни управления реализуются в основном техническими средствами, называемыми станциями локальной сети (СЛС) и адаптерами ЛВС. По существу, адаптер вместе с физическим каналом образует информационный моноканал, к которому подключаются системы сети, выступающие в качестве абонентов моноканала.

Основное достоинство семиуровневой модели ВОС заключается в том, что в процессе развития и совершенствования любой системы вызывает потребность изменять ее отдельные компоненты, а это вызывает изменение в других компонентах системы, что существенно усложняет и затрудняет процесс модернизации системы. В семиуровневой модели ВОС, где между уровнями определены однозначно интерфейсы, изменение одного из уровней не влечет за собой необходимости внесения изменений в другие уровни. Таким образом, существует относительная независимость уровней друг от друга.

Функции, описываемые уровнями модели, должны быть реализованы либо в аппаратуре, либо в виде программ.

Функции физического уровня реализуются в аппаратуре. Это адаптеры, мультиплексоры передачи данных, сетевые платы и т. д.

Функции остальных уровней реализуются в виде программных модулей – драйверов.

2.5. Уровневые услуги ВОС в ЛВС

В рамках эталонной модели ВОС определяются следующие уровневые услуги:

- физический уровень – установление и идентификация физических соединений; организация последовательности передачи отдельных битов информации; оповещение об окончании связи;
- канальный уровень – организация требуемой последовательности блоков данных и их передача; управление потоками между степенными узлами; идентификация конечных пунктов канальных соединений; обнаружение и исправление ошибок; оповещение об ошибках, которые не исправлены на канальном уровне;
- сетевой уровень – идентификация конечных точек сетевых соединений; организация сетевых соединений; управление потоками блоков данных; обеспечение последовательности доставки блоков данных; обнаружение ошибок и формирование сообщений о них; разъединение сетевых соединений;
- транспортный уровень – установление и разъединение транспортных соединений; формирование блоков данных; обеспечение взаимодействия сеансовых соединений с транспортными соединениями; управление последовательностью передачи блоков данных; обеспечение целостности блоков данных во время передачи; обнаружение и исправление ошибок, сообщения о неисправленных ошибках, представление приоритетов

в передаче блоков; передача подтверждений о принятых блоках; ликвидация тупиковых ситуаций;

- сеансовый уровень – обслуживание сеансов и обеспечение передачи данных в диалоговом режиме; установление сеансового соединения; обмен данными; управление обменом; синхронизация сеансового соединения; сообщения об исключительных ситуациях; отображение сеансового соединения на транспортный уровень; завершение сеансового соединения;
- представительский уровень – выбор вида представления данных; интерпретация и преобразование передаваемых данных к виду, удобному для прикладных процессов; преобразование синтаксиса данных; формирование данных;
- прикладной уровень – управление терминалами – файлами, диалогом, задачами, сетью в целом; целостность информации; ряд дополнительных услуг. К дополнительным услугам уровня относятся услуги по организации электронной почты, передаче массовых сообщений и т. д.

Услуги различных уровней модели ВОС определяются с помощью протоколов, которые представляют собой правила взаимодействия объектов одноименных уровней открытых систем.

Протокол – это набор правил, определяющих взаимодействие двух одноименных уровней моделей ВОС в различных абонентских ЭВМ.

В соответствии с 7-уровневой моделью ВОС вводится 7 типов протоколов, которые именуются так же, как уровни. При этом по функциональному назначению все протоколы целесообразно разделить на три группы.

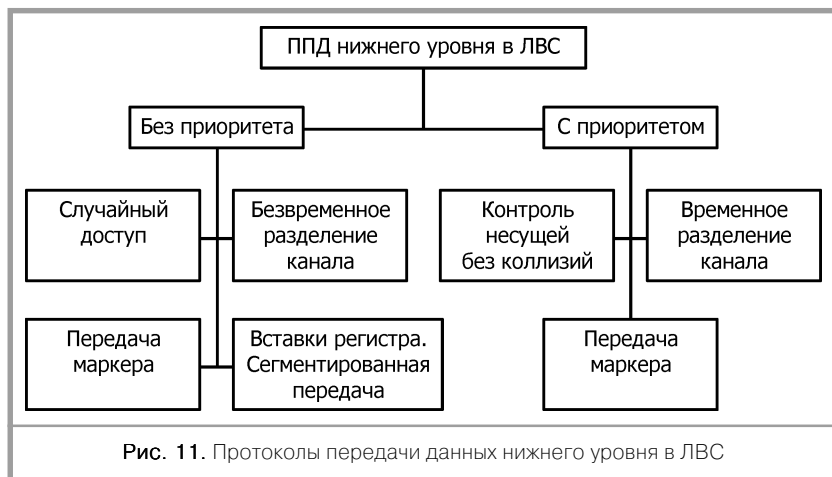
Первую группу составляют протоколы абонентской службы, соответствующие прикладному, представительскому и сеансовому уровням модели ВОС. Протоколы этой группы являются сетезависимыми, то есть их характеристики и структура не зависят от используемой сети передачи данных. Они определяются лишь структурой абонентских систем и решаемыми задачами обработки данных. Две другие группы протоколов описывают транспортную службу компьютерной сети и различаются между собой процедурой доступа к передающей среде. Одна из этих групп определяет систему передачи данных с *маршрутизацией информации*, а другая – с *селекцией информации*.

Маршрутизация представляет собой процедуру определения пути передачи информации в сетях передачи данных и характерна для глобальных и региональных компьютерных сетей, в рамках которых и рассматривается соответствующая группа протоколов.

Селекция определяет процесс выбора очередной абонентской системы для подключения ее к сети передачи данных с целью обмена

информацией. Селекция информации в основном используется в системах передачи данных локальных компьютерных сетей, где и рассматривается третья группа протоколов.

Протоколы передачи данных нижнего уровня, получившего распространение в ЛВС, приведены на рис. 11.



Проще всего представить особенности сетевых протоколов на примере протоколов канального уровня, которые делятся на две основные группы:

- байт-ориентированные;
- бит-ориентированные.

Байт-ориентированный протокол обеспечивает передачу сообщения по информационному каналу в виде последовательности байтов. Кроме информационных байтов, в канал передаются также управляющие и служебные байты. Такой тип протокола удобен для ЭВМ, так как она ориентирована на обработку данных, представленных в виде двоичных байтов. Для коммуникационной среды байт-ориентированный протокол менее удобен, так как разделение информационного потока в канале на байты требует использования дополнительных сигналов, что в конечном счете снижает пропускную способность канала связи.

Наиболее известным и распространенным байт-ориентированным протоколом является протокол двоичной синхронной связи BSC (Binary Synchronous Communication), разработанный фирмой IBM.

Бит-ориентированный протокол предусматривает передачу информации в виде потока битов, не разделяемых на байты. Поэтому для разделения кадров используются специальные последовательности – флаги. В начале кадра ставится флаг открывающий, а конце – флаг закрывающий. Этот протокол удобен относительно коммуникационной среды, так как канал связи как раз и ориентирован на передачу последовательности битов.

Типичным представителем группы бит-ориентированных протоколов являются протоколы HDLC (High-level Data Link Control – высший уровень управления каналом связи) и его множества.

Работы по стандартизации вычислительных сетей ведутся большим количеством организаций. В зависимости от статуса организаций различают следующие виды стандартов:

- стандарты отдельных фирм;
- стандарты специальных комитетов и объединений, создаваемых несколькими фирмами;
- национальные стандарты (например, стандарт безопасности для операционных систем, разработанный Национальным центром компьютерной безопасности (NCSC) Министерства обороны США);
- международные стандарты (например, стандарт на сети с коммутацией пакетов X.25).

Для протоколов физического уровня стандарты определены рекомендациями МККТТ (Международного консультативного комитета по телеграфии и телефонии). Цифровая передача предусматривает использование протоколов X.21 и X.21-бис.

Канальный уровень определяет протокол HDLC и его подмножества, а также протокол X.25/3.

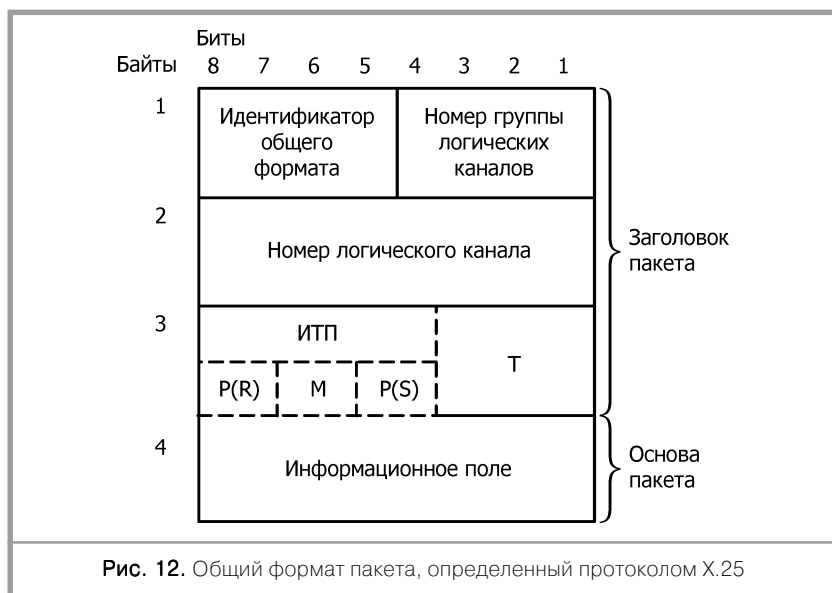
Широкое распространение ЛВС потребовало разработки стандартов для этой области. В настоящее время для ЛВС используются стандарты, разработанные Институтом инженеров по электротехнике и радиоэлектронике – ИИЭР (IEEE – Institute of Electrical and Electronics Engineers).

Комитеты IEEE 802 разработали ряд стандартов, часть из которых принята МОС (ISO) и другими организациями. Для ЛВС разработаны следующие стандарты:

- 802.1 – верхние уровни и административное управление;
- 802.2 – управление логическим звеном данных (LLC);
- 803.3 – случайный метод доступа к среде (CSMA/CD – Carries Sense Multiple Access with Collision Detection – множественный доступ с контролем передачи и обнаружением столкновений);
- 802.4 – маркерная шина;
- 802.5 – маркерное кольцо;
- 802.6 – городские сети.

2.6. Протокол сети передачи данных X.25

К наиболее распространенным протоколам сети передачи данных относится группа протоколов, определенная рекомендацией X.25 для сетей коммутации пакетов. Для облегчения передачи информации протокол X.25/3 определяет 19 типов пакетов, общий формат которых представлен на рис. 12. Первые три бита являются общими для всех типов пакетов. Первый байт содержит поле идентификатора общего формата и поле группового номера логического канала. Поле идентификатора общего формата (четыре бита) предназначено для указания общего формата (структуры) остальной части заголовка, который зависит от типа пакета.



Рекомендация X.25 допускает одновременное существование некоторого множества сгруппированных между собой логических каналов. Для указания номера группы логического канала, ответственного абонентской системе, и предназначено поле группового номера логического канала. Это поле занимает остальные четыре бита первого байта. Второй байт пакета содержит поле номера логического канала в рамках определенной группы. Таким образом обеспечивается логическая идентификация 4096 логических каналов, из которых 4095 предоставляются пользователю, а один с но-

мером «0» зарезервирован для управления (пакеты рестарта и диагностики). При установлении виртуального соединения абонентская система выбирает свободный логический канал из числа доступных ей каналов.

Содержимое третьего байта для управляющих пакетов трактуется как поле идентификатора типа пакета (ИТП), а для информационных – байтом последовательности передачи пакетов. К информационным относятся пакеты «данные» и «дейтаграммы», содержащие информацию пользователя. Остальные пакеты являются управляющими и обеспечивают информационными пакетами по сети коммутации пакетов. Идентификация типов пакета осуществляется на основании бита типа пакета (Т), расположенного в третьем байте пакета. Для информационных пакетов значение этого бита равно нулю, а для управляющих пакетов – единице. В свою очередь, по функциональному назначению управляющие пакеты делятся на пакеты:

- установления и завершения соединения;
- управления потоком и повторной установки;
- прерывания;
- повторного пуска.

Бит «М» является признаком продолжения данных и используется для указания цепочки пакетов. Значение M=1 указывает на то, что передается промежуточный пакет и передача пакетов данных на этом не заканчивается. При M=0 пакет данных является последним в цепочке пакетов.

Длина поля данных в информационных пакетах может быть равной 16, 32, 64, 128, 256, 512, 1024, 2048 или 4096 байтам. По умолчанию длина поля данных считается равной 128 байтам. В управляющих пакетах поле данных вообще может отсутствовать.

Рекомендация X.25 позволяет организовать четыре способа обмена данными:

- виртуальный вызов;
- постоянная виртуальная цепь;
- вызов с быстрым выбором;
- вызов с быстрым выбором и немедленной считкой.

2.7. Методы доступа к передающей среде в ЛВС

Типичными методами доступа к передающей среде в современных ЛВС являются:

- множественный доступ с контролем несущей и обнаружением конфликтов (CSMA/CD), иначе называемый методом доступа

Ethernet, так как именно в этой сети он получил наибольшее распространение;

- маркерное кольцо (метод доступа Token Ring);
- маркерная шина (метод доступа ARCnet).

Метод доступа Ethernet (метод случайного доступа) разработан фирмой Xerox в 1975 г. и используется в ЛВС с шинной технологией, обеспечивает высокую скорость передачи данных и надежность. Это метод множественного доступа с прослушиванием несущей и разрешением конфликтов (коллизий). Каждая рабочая станция (PC) перед началом передачи определяет, свободен канал или занят. Если канал свободен, PC начинает передачу данных, осуществляемую пакетами, упакованными в кадры. Из-за различных системных задержек могут возникнуть коллизии. В этом случае станция задерживает передачу на определенное время. Для каждой PC устанавливается свое время ожидания перед повторной передачей кадра. Коллизии приводят к снижению быстродействия сети только при сравнительно большом количестве активных PC (до 80–100).

Метод доступа Token Ring разработан фирмой IBM и рассчитан на кольцевую топологию сети. Это селективный метод доступа в кольцевой моноканал, именуемый «маркерное кольцо». В качестве маркеров используется уникальная последовательность битов. Маркер не имеет адреса и может находиться в одном из двух состояний – свободном или занятом. Если ни одна PC не готова к передаче данных, свободный маркер циркулирует по кольцу. Станция, имеющая кадр для передачи, ждет подхода свободного маркера, захватывает его, изменяет состояние маркера на «занятый» и добавляет к нему кадр. Занятый маркер с кадром перемещается по кольцу и возвращается к станции-отправителю, причем при прохождении через узел назначения снимается конец кадра. Станция-отправитель удаляет свой кадр из кольца, изменяет состояние маркера на «свободный» и передает его дальше по кольцу. С этого момента любая станция может изменить состояние маркера на «занятый» и начать передачу данных. Описанная процедура характерна для сети, в которой все станции имеют одинаковый приоритет. В рамках метода «маркерное кольцо» предусматривается возможность передачи кадров станции с учетом их приоритета. Тогда станции с низким приоритетом могут захватывать кольцо в случае неактивности станций с более высоким приоритетом.

Для различных видов сообщений, передаваемых кадрам, могут назначаться различные приоритеты от «0» (низший) до «7» (высший). Решение о приоритете конкретного кадра принимает передающая станция (протокол Token Ring получает этот параметр через межуровневые интерфейсы от протоколов верхнего уровня, например прикладного). Маркер также всегда имеет некоторый уровень

текущего приоритета. Станция имеет право захватить переданный ей маркер только в том случае, если приоритет кадра, который она хочет передать, выше (или равен) приоритета маркера. В противном случае станция обязана передать маркер следующей по кольцу станции.

Метод доступа ARCnet разработан фирмой Datapoint Corp. и используется в ЛВС с топологией «звезда» и «общая шина». Это селективный метод доступа в моноканал, называемый «маркерная шина». Маркер создается одной из станций сети и имеет адресное поле, где указывается номер (адрес) станции, владеющий маркером. Передачу производит только та станция, которая в данный момент владеет маркером (эстафетной палочкой). Остальные станции работают на прием. Последовательность передачи маркера от одной станции к другой задается управляющей станцией сети. Станции, последовательно получающие маркер для передачи кадров, образуют «логическое кольцо». Станция, получившая маркер (полномочия на передачу информации), передает свой подготовленный кадр в шину. Если кадра для передачи нет, она сразу посылает маркер другой станции согласно установленному порядку передачи полномочий. Так продолжается до тех пор, пока управляющая станция не инициирует новую последовательность передач маркера. Станция назначения, получившая маркер с кадром, «отцепляет» кадр от маркера и передает маркер той станции, которая является следующей в установленной последовательности передач. При таком методе доступа в моноканал имеется возможность обеспечить приоритетное обслуживание абонентов, например в течение одного цикла, когда маркер совершает полный оборот по «логическому кольцу». Станции с более высоким приоритетом получают маркер не один, а несколько раз.

В качестве примера приведем структуру пакета по стандарту IEEE 802.3 с указанием длины каждого поля в байтах.

Преамбула	Признак начала пакета	Назначение	Длина	Данные	Источник	Набивка	CRC-сумма
7	1	2 или 6	2	0–1500	2 или 6	?	4

Преамбула – это поле, содержащее семь одинаковых байтов 101010110, предназначенных для синхронизации.

Признак начала пакета – однобайтовое поле для обозначения начала пакета.

Назначение – поле длиной 2 или 6 байт (в зависимости от типа ЛВС) указывает, для какой рабочей станции пакет предназначен.

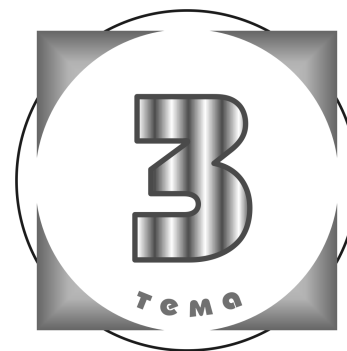
Длина – здесь содержится информация о длине данных и пакете. Данные – в это поле записываются данные, составляющие передаваемое сообщение.

Источник – в этом поле содержится адрес отправителя пакета.

Набивка – сюда вставляют пустые символы для доведения длины пакета до минимально допустимой величины. При достаточно большой длине поля данных поле набивки может отсутствовать.

CRC-сумма – здесь содержится контрольное число, используемое на приемном пункте для выявления ошибок в данных принятого пакета. В качестве контрольного числа применяется остаток избыточной циклической суммы, вычисленной с помощью полинома типа CRC-32. На приемном пункте также производится вычисление этого остатка и затем его сравнение с содержимым рассматриваемого поля с целью обнаружения ошибок в принятых данных.

Общая длина пакета стандарта IEEE 802.3 находится в диапазоне от 64 до 1518 байт, не считая преамбулы и признака начала пакета.



Топология и способы организации ЛВС

3.1. Топологии ЛВС

Фирмой Novell была разработана операционная система NetWare. Она может использоваться в любой распространенной в настоящее время физической структуре ЛВС, поэтому сеть, поддерживаемая сетевой ОС NetWare, может иметь шинную, кольцевую и звездообразную топологии. Из-за большой популярности структуры сети Ethernet в дальнейшем рассматривается использование Novell NetWare для этого типа топологии.

ЛВС фирмы Novell представляет собой сеть шинной топологии, для реализации которой используется аппарататура Ethernet.

ЛВС состоит из *файлового сервера* и *рабочих станций*, включенных в *сетевой сегмент*. *Сетевой сегмент* представляет собой отрезок коаксиального кабеля с подключенными к нему компьютерами. Он может быть отдельной локальной компьютерной сетью либо частью сети. Основным типом передающей среды для ЛВС является коаксиальный кабель. Максимальная длина сетевого сегмента составляет 185 м, с помощью повторителей можно соединить до пяти сегментов. В состав одного сегмента входит до 30 рабочих станций.

В последнее время большую популярность получил вариант сети на базе витой пары проводов. Он предусматривает подключение рабочих станций к концентратору. Например, один концентратор

в состоянии поддержать работу 12 станций, расположенных на расстоянии 120 м от него. Концентраторы можно соединять каскадами, и максимальное число сегментов в одной сети может составлять 1024.

Таким образом, реализация локальной сети фирмы Novell возможна на двух типах топологии: шинной и звездообразной. За рубежом предпочтение отдается витой паре из-за дешевизны. В России используется главным образом тонкий коаксиальный кабель.

Структурные схемы ЛВС на тонком кабеле и витой паре соответственно приведены на рис. 13.

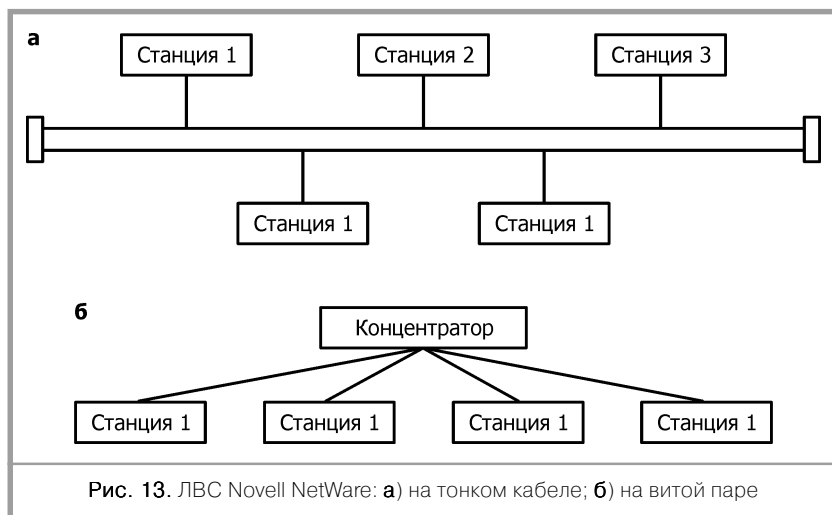


Рис. 13. ЛВС Novell NetWare: а) на тонком кабеле; б) на витой паре

В локальной сети с централизованным управлением большую роль играет *выделенный сервер*, который может выполнять разные функции: файл-сервера, сервера печати, сервера баз данных и т. д.

Основным ресурсом ЛВС Novell является *файловый сервер*. На нем размещаются сетевая операционная система, база данных и прикладные программы пользователей. Файл-сервер должен быть самым мощным компьютером в сети, так как от него зависят производительность и функциональные возможности сети в целом.

Для реализации файл-сервера необходим персональный компьютер с объемом оперативной памяти не менее 8 Мбайт. Желательно, чтобы объем оперативной памяти составлял 16–32 Мбайт с учетом возможности расширения сети. Емкость винчестера файл-сервера – главного разделяемого ресурса поддерживаемой им сети – должна составлять 500–800 Мбайт.

Ввиду того, что надежность работы файл-сервера определяет надежность работы всей сети, необходимо принимать специальные меры для защиты информации на жестком диске от сбоев и потерь. Одним из таких мер является зеркальное отображение диска. К контроллеру жесткого диска файл-сервера подключаются два дисководы, и информация записывается на оба диска одновременно. При отказе одного из дисководов автоматически осуществляется переход на другой. Но этот способ все же не спасает при выходе из строя контроллера.

Большую надежность обеспечивает способ дублирования дисков. В этом случае на файл-сервере устанавливаются два контроллера, каждый из которых обслуживает свой диск. Таким образом создаются два независимых канала записи на жестких дисках. Информация на обоих дисках дублируется. Поэтому при выходе из строя одного контроллера начинает работать другой. А вероятность отказа двух контроллеров одновременно очень мала. Используется также включение в состав одного сетевого сегмента двух файловых серверов. Кроме того, файл-серверы подключаются к электрической сети через источник бесперебойного питания.

Необходимым ресурсом в ЛВС является также и принтер, обеспечивающий выполнение функций сетевой печати. Для этой роли можно использовать отдельный компьютер, а можно совмещать функции печати с функциями файл-сервера.

Компьютер, выполняющий функции *рабочей станции*, должен обеспечить пользователю возможность решения всех его прикладных задач.

Если *рабочая станция* ориентирована только на сетевой режим работы, то ей, в сущности, не нужны ни винчестер, ни гибкие диски. Появляется возможность использовать бездисковые рабочие станции. Операционная система на такой станции загружается дистанционно из файл-сервера под управлением постоянного запоминающего устройства, установленного в сетевой плате рабочей станции.

Бездисковые рабочие станции значительно дешевле дисковых, работа на них исключает возможность занесения в сеть вируса. В то же время в ЛВС, построенной на базе бездисковых рабочих станций, резко возрастает нагрузка на файл-сервер и исключается возможность работы на станции в автономном режиме. Вполне естественно, что требования к рабочим станциям более скромные, чем к файл-серверу. Большую часть пользователей вполне удовлетворяет объем оперативной памяти 8–16 Мбайт и винчестер 650 Мбайт.

Технические характеристики ЛВС определяются ее функциональным назначением, сложностью прикладных задач пользователей и экономическими возможностями предприятия, использующего ЛВС.

Простейшей физической средой в ЛВС, соединяющей рабочие станции, концентраторы и серверы, является витая проводная пара. Ее использование снижает стоимость ЛВС, во-первых, по причине дешевизны самого носителя, а во-вторых, благодаря наличию на многих объектах резервных пар в телефонных кабелях, которые могут быть выделены для передачи данных. К недостаткам витой пары как среды передачи данных относятся плохая защищенность от электрических помех, простота несанкционированного подключения, ограничения на дальность (сотни метров и скорость передачи данных (несколько сотен килобитов в секунду)).

Единая операционная поддержка, включая метод теледоступа, предусмотрена в однородных ЛВС. Сложнее обстоит дело с ЛВС, использующими ЭВМ различных классов и моделей, например мини-ЭВМ и большие вычислительные машины.

Методы теледоступа поддерживают многоуровневые системы интерфейсов. Различают многоуровневые (модель открытых систем и двухуровневые) ЛВС. К двухуровневым применяют двухуровневые терминальные комплексы со стандартными методами теледоступа (базисный телекоммуникационный метод доступа – БТМД).

Многожильные кабели значительно дороже, чем витая пара, хотя и обладают примерно такими же свойствами.

Следующей распространенной средой передачи данных в современных ЛВС является коаксиальный кабель. Он прост по конструкции, имеет небольшую массу и умеренную стоимость и в то же время обладает хорошей электрической изоляцией, допускает работу на довольно больших расстояниях (сотни метров – километры) и высоких скоростях (десятки мегабитов в секунду).

В последнее время все большее применение находят оптоволоконные кабели (световоды), которые обладают рядом преимуществ. Они имеют небольшую массу, способны передавать информацию с очень высокой скоростью (свыше 1 Гбит/с), не восприимчивы к электрическим помехам, сложны для несанкционированного подключения, пожаро- и взрывобезопасны. В то же время с ними связан ряд проблем: сложность технологии сращивания, возможность передачи данных только по одному направлению, высокая стоимость модемов, ослабление сигнала при подключении ответвителей и др.

Радиосреда в ЛВС используется мало из-за экранированности зданий, ограничений организационно-юридического плана, низких скоростей передачи (до 2 Мбит/с), слабой помехозащищенности, относительно высокой стоимости оборудования и низкой защищенности от несанкционированного доступа (снятия информации). Основное достоинство радиоканала – отсутствие кабеля, возможность обслуживания мобильных станций, работа с неограниченным количеством рабочих станций.

Для передачи информации в ЛВС может также использоваться инфракрасный диапазон волн, хорошо обеспечивающих передачу цифровых сигналов в пределах одного помещения. Установленная на потолке «интеллектуальная лампочка» служит интерфейсом с сетью здания, а также управляет сигналами на локальной «инфракрасной шине»

3.2. Средства реализации ЛВС

Рассмотрим подробнее оборудование, используемое в локальных сетях.

Сетевые адаптеры (СА). Основные функции адаптеров и их технические характеристики определяются поддерживаемым уровнем протокола ЛВС в соответствии с архитектурой семиуровневой эталонной модели ВОС.

Адаптер (adapter) – устройство либо программа, предназначенные для согласования параметров входных и выходных сигналов в целях сопряжения объектов.

По выполняемым функциям СА разделяются на две группы:

- 1) реализующие функции физического и канального уровней. Такие адаптеры, выполненные в виде интерфейсных плат, отличаются технической простотой и невысокой стоимостью. Они применяются в сетях с простой технологией, где почти отсутствует необходимость выполнения таких функций, как маршрутизация пакетов, формирование из поступающих пакетов сообщений, согласование протоколов различных сетей и др.;
- 2) реализующие функции первых четырех уровней модем ВОС – физического, канального, сетевого и транспортного. Эти адаптеры, кроме функций СА первой группы, могут выполнять функции маршрутизации, ретрансляции данных, формирования пакетов из передаваемого сообщения (при передаче), сборки пакетов в сообщения (при приеме), согласования протоколов передачи данных различных сетей, сокращая таким образом затраты вычислительных ресурсов ЭВМ на организацию сетевого обмена. Технически они могут быть выполнены на базе микропроцессоров. Естественно, что такие адаптеры применяются в ЛВС, где имеется необходимость в реализации перечисленных функций.

Адаптеры ориентированы на определенную архитектуру локальной сети и ее технические характеристики, поэтому по топологии ЛВС адаптеры разделяются на следующие группы: поддерживающие шинную топологию, кольцевую, звездообразную, древовидную, комбинированную (звездно-кольцевую, звездно-шинную).

Дифференциация адаптеров по выполненным функциям и ориентации их на определенную архитектуру ЛВС привели к большому многообразию типов адаптеров и разбросу их характеристик.

Концентраторы (хабы) – концентратор – устройство либо функциональный блок, у которого суммарная пропускная способность входных каналов выше пропускной способности выходного канала. Так как потоки входных данных в концентраторе больше выходного потока, то главной его задачей является концентрация данных. При этом случаются ситуации, когда число блоков данных, поступающее на входы концентратора, превышает его возможности. Тогда концентратор ликвидирует часть этих блоков. Ядром концентратора является процессор. Для объединения входной информации чаще всего используется множественный доступ с разделением времени (ТДМА). TDMA – Time Division Multiple Access – множественный доступ по принципу временного мультиплексирования каналов. Функции, выполняемые концентратором, близки к задачам, возложенным на мультиплексор. Нарастиваемые (модульные) концентраторы позволяют выбирать их компоненты, не думая о совместимости с уже используемыми. Современные концентраторы имеют порты для подключения к разнообразным локальным сетям. Они удобны для формирования сети произвольной топологии. Выпускается ряд типов концентраторов – пассивных и активных с автономным питанием, выполняющих роль повторителя. Они отличаются по количеству, типу и длине подключаемых кабелей и могут автоматически управлять подсоединенными сегментами (включать и выключать их в случае обнаружения сбоев и обрывов).

Концентраторы характерны практически для всех базовых технологий локальных сетей – Ethernet, Arcnet, Token Ring, FDOI, Fast Ethernet, Gigabit Ethernet, 100VG – AnyLAN.

Следует подчеркнуть, что в работе концентраторов любых технологий много общего – они повторяют сигналы, пришедшие с одного из своих портов, на других своих портах. Разница состоит в том, на каких именно портах повторяются входные сигналы. Так, концентратор Ethernet повторяет входные сигналы на всех своих портах, кроме того, с которого сигналы поступают (рис. 14, а). А концентратор Token Ring (рис. 14, б) повторяет входные сигналы, поступающие с некоторого порта, только на одном порту – на том, к которому подключен следующий в кольце компьютер. В качестве концентратора может выступать как компьютер, так и специализированное устройство, такое как многовходовый повторитель, коммутатор или маршрутизатор. Повторитель, который имеет несколько портов и соединяет несколько физических сегментов, называют концентратором, или хабом (*hub* – основа, центр деятельности), который отра-

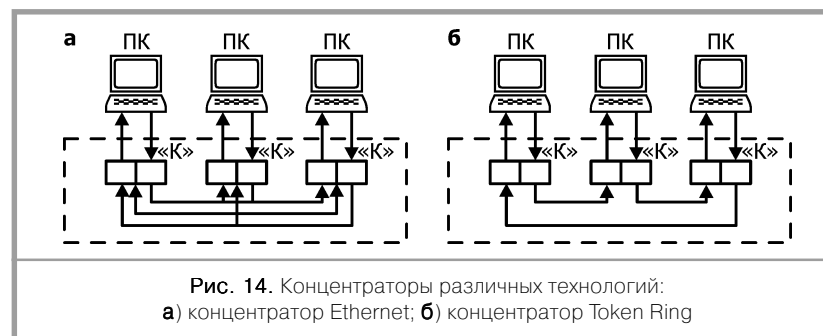


Рис. 14. Концентраторы различных технологий:
а) концентратор Ethernet; б) концентратор Token Ring

жает тот факт, что в данном устройстве сосредоточиваются все связи между сегментами сети.

Приемопередатчики (трансиверы) и повторители (репитеры). С помощью этих устройств можно объединить несколько сегментов сети с шинной технологией, увеличивая таким образом общую протяженность сети.

Трансивер (приемопередатчик) – (*transmitter + receiver = transceiver*) – это устройство, предназначенное для приема пакетов от контроллера рабочих станций и передачи их в шину. Он также разрешает коллизии в шине. Конструктивно приемопередатчик и контроллер могут объединяться на одной плате или находиться в различных узлах.

Трансивер соединяется с сетевым адаптером интерфейсным кабелем длиной 50 м. Допускается подключение к одному сегменту не более 100 трансиверов, причем расстояние между подключениями трансиверов не должно быть меньше 2,5 м.

Трансивер – это часть сетевого адаптера, которая выполняет следующие функции:

- прием и передача данных с кабеля на кабели;
- определение коллизий на кабеле;
- электрическая развязка между кабелем и остальной частью адаптера;
- защита кабеля от некорректной работы адаптера.

Репитер (повторитель) (repeater) – устройство с автономным питанием, обеспечивающее передачу данных между сегментами определенной длины. Он служит для объединения в одну сеть нескольких сегментов кабеля и увеличения тем самым общей длины сети. Повторитель принимает сигналы из одного сегмента кабеля и побитно синхронно повторяет их в другом сегменте, улучшая форму и мощность импульсов, а также синхронизируя импульсы. По-

вторитель состоит из двух или нескольких трансиверов, которые присоединяются к сегментам кабеля, а также блока повторения со своим тактовым генератором.

Разрешается использовать в сети не более 4 повторителей и, собственно, не более 5 сегментов кабеля. При максимальной длине сегмента кабеля в 500 м это дает максимальную длину сети в 2500 м. Только 3 сегмента из пяти могут быть нагруженными, то есть такими, к которым подключаются конечные узлы. Между нагруженными сегментами должны быть ненагруженные сегменты, так что максимальная конфигурация сети представляет собой два нагруженных крайних сегмента, которые соединяются ненагруженными сегментами еще с одним центральным нагруженным сегментом. На рис. 15 приведена сеть Ethernet, состоящая из трех сегментов, объединенных двумя повторителями. Крайние сегменты являются нагруженными, а промежуточные – ненагруженными.

Мосты – это устройства для логической структуризации сети.

Мост (bridge) – ретрансляционная система, соединяющая два канала передачи данных. Они используются для соединения в основном идентичных сетей, имеющих некоторые физические различия на физическом и канальном уровнях. Мост делит разделяющую среду передачи сети на части (часто называемые логическими сегментами), передавая информацию из одного сегмента в другой только в том случае, если такая передача действительно необходима, то есть если адрес компьютера назначения принадлежит другой подсети. Тем самым мост изолирует трафик одной подсети от трафика другой, повышая общую производительность передачи данных в сети. Промышленностью выпускается довольно широкая номенклатура мостов.

Мосты используются для локализации трафика аппаратного адреса компьютеров. Это затрудняет распознавание принадлежности того или иного компьютера к определенному логическому сегменту – сам адрес не содержит никакой информации. Поэтому мост достаточно упрощенно представляет деление сети на сегменты – он запоминает, через какой порт на него поступает кадр данных от каждого компьютера сети, и в дальнейшем передает кадры, предназначенные для этого компьютера, на этот порт.

На рис. 16 показана логическая структуризация сети с помощью моста. Сети 1-го и 2-го отделов состоят из отдельных логических сегментов. Каждый логический сегмент построен на базе концентратора и имеет простейшую физическую структуру, образованную отрезками кабеля, связывающими компьютеры с портами концентратора.

В соответствии с базовой эталонной моделью ВОС мост описывается протоколами физического и канального уровней, над которыми

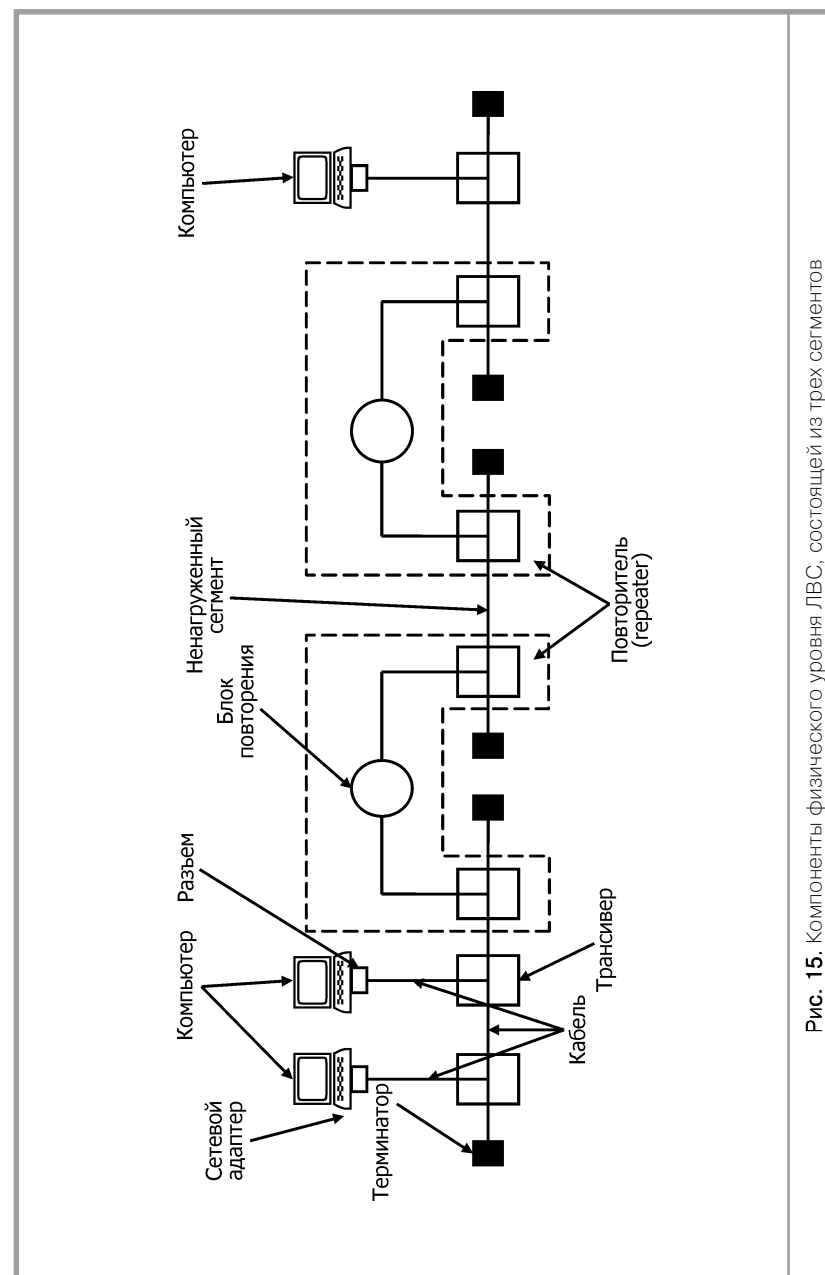
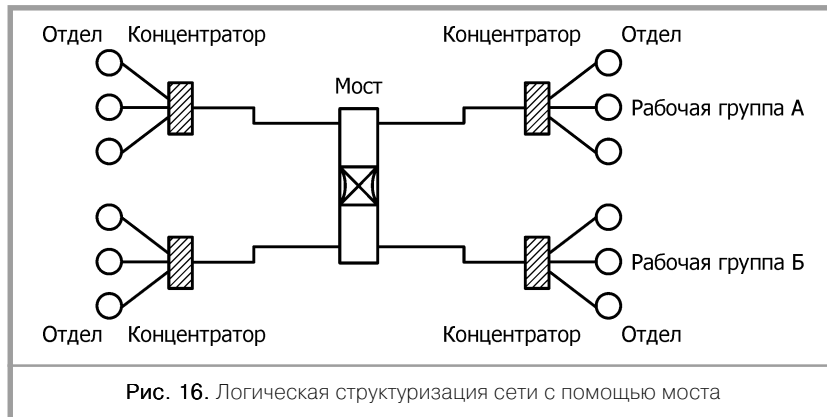
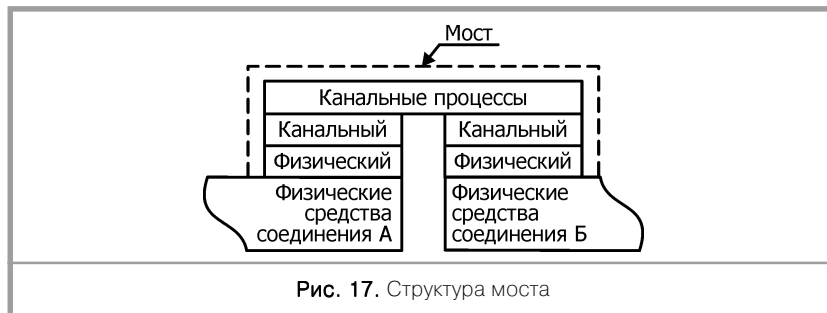


Рис. 15. Компоненты физического уровня ЛВС, состоящей из трех сегментов



ми располагаются каналные процессы, – рис. 17 (структура моста). Мост опирается на пару связанных им физических средств соединения, которые в этой модели представляют физические каналы. Мост преобразует физический (1А, 1В) и каналный (2А, 2В) уровни различных типов.



Коммутатор (switch) по принципу обработки кадров ничем не отличается от моста. Основное его отличие от моста состоит в том, что он является своего рода коммуникационным мультипроцессором, так как каждый его порт оснащен специализированным процессором, который обрабатывает кадры по алгоритму моста независимо от процессоров других портов. За счет этого общая производительность коммутаторов обычно намного выше производительности традиционного моста, имеющего один процессорный блок. Можно сказать, что коммутаторы – это мосты нового поколения, которые обрабатывают кадры в параллельном режиме. Они

обладают большей пропускной способностью, что важно для интерактивного трафика между взаимодействующими рабочими станциями. В сети Ethernet коммутаторы обрабатывают полученный пакет в реальном масштабе времени, обеспечивая нужную латентность и высокую скорость коммутации.

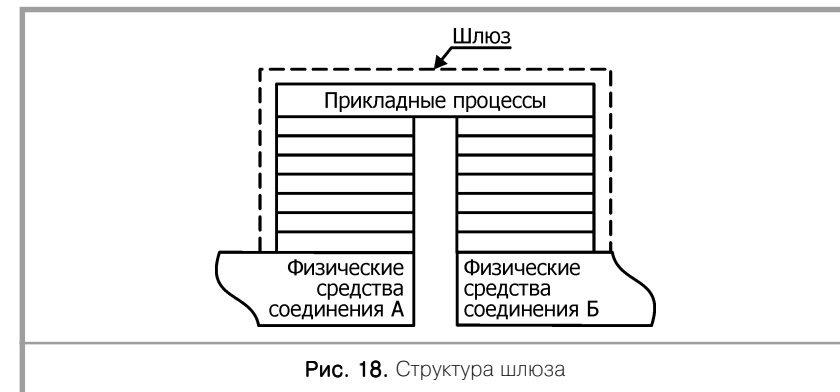
Коммутатор локальной сети (*local-area network switch*) – устройство, обеспечивающее взаимодействие сегментов одной либо группы локальных сетей. Он, как и обычный коммутатор, обеспечивает взаимодействие подключенных к нему абонентов. Но в дополнение к этому он осуществляет преобразование интерфейса, если соединяются различные типы сегментов локальных сетей. В перечень функций, выполняемых коммутатором локальных сетей, входят:

- обеспечение сквозной коммутации;
- наличие средств маршрутизации;
- поддержка простого протокола управления сетью (SNMP);
- имитация моста либо маршрутизатора;
- организация виртуальных сетей;
- скоростная ретрансляция блоков данных.

Отдельные части локальной сети может соединять шлюз.

Шлюз (gateway) – ретрансляционная система, обеспечивающая взаимодействие двух информационных систем. Шлюзы применяются для различных сетей. Они дают возможность объединить сети с разными типами системного и прикладного программного обеспечения и выполняют протокольные преобразования для всех семи уровней модели ВОС, в частности маршрутизацию пакетов, преобразование сообщения из одного формата в другой или из одной системы кодирования в другую.

Шлюз – наиболее сложная ретрансляционная система, обеспечивающая взаимодействие двух (рис. 18 – структура шлюза) сетей



с различными (1А–7А и 1В–7В) наборами протоколов всех семи уровней. В свою очередь, наборы протоколов могут опираться на различные типы физических средств соединения. Штатные протоколы и, естественно, информационные сети объединяются в единое целое специальными прикладными процессами шлюза.

Шлюз осуществляет свои функции на уровне выше сетевого. Он не зависит от используемой передающей среды, но зависит от используемых протоколов обмена данными. Обычно шлюз выполняет преобразование между двумя протоколами. С помощью шлюзов можно подключить ЛВС к главному компьютеру или к глобальной сети.

В тех случаях, когда соединяются информационные сети, созданные по стандартам Международной организации по стандартизации (ISO), в них часть уровней может иметь одни и те же протоколы. Ограничения, связанные с применением мостов и коммутаторов – по топологии связей, а также ряд других причин приводят к тому, что сети соединяются с еще одним коммуникационным устройством – маршрутизатором.

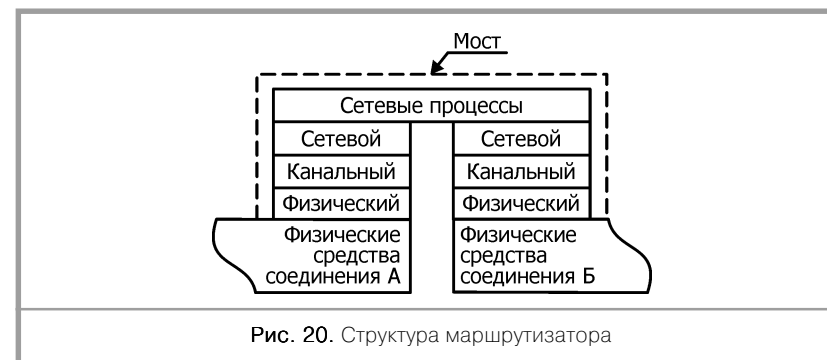
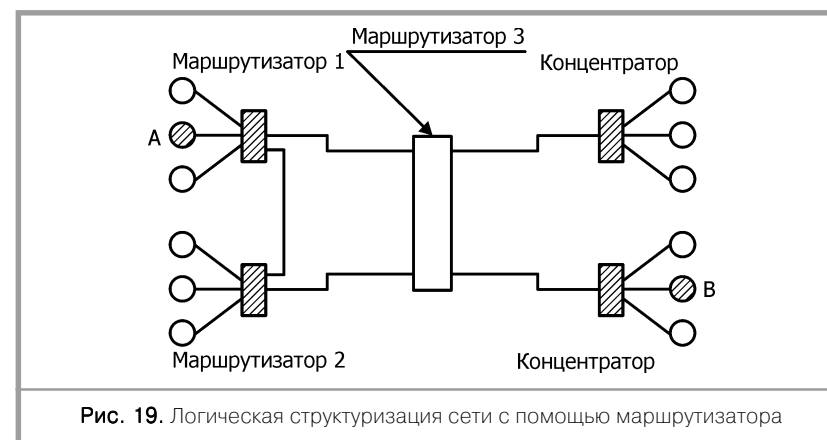
Маршрутизатор (роутер) (*router*) – ретрансляционная система, соединяющая две коммуникационные сети либо их части. Маршрутизаторы образуют логические сегменты посредством явной адресации, поскольку используют не плоские аппаратные, а составные числовые адреса. В этих адресах имеется поле номера сети, так что все компьютеры, у которых значение этого поля одинаково, принадлежат к одному сегменту, называемому в данном случае подсетью (*subnet*).

Маршрутизаторы более надежно и более эффективно, чем мосты, изолируют трафик отдельных частей сети друг от друга. Кроме локализации трафика, маршрутизаторы выполняют еще много других полезных функций. Они могут работать в сети с замкнутыми контурами, при этом осуществляя выбор наиболее рационального маршрута из нескольких возможных.

Сеть, представленная на рис. 19, отличается от сети с мостом (рис. 16) тем, что между подсетями отделов 1 и 2 проложена дополнительная связь, которая может использоваться как для повышения производительности сети, так и для повышения ее надежности.

Другой очень важной функцией маршрутизаторов является их способность связывать в единую сеть подсети, построенные с использованием разных сетевых технологий, например Ethernet и X.25.

Маршрутизаторы устанавливают соединения на 4-м (транспортном) уровне, при этом верхние уровни сети (5-й, 6-й и 7-й) должны быть одинаковы. Каждый маршрутизатор (рис. 20) реализует про-



токолы физического (1А, 1В), канального (2А, 2В) и сетевого (3А, 3В) уровней. Специальные сетевые процессы соединяют части коммутатора в единое целое. Физические, канальные и сетевые протоколы в разных сетях различны, поэтому соединение пар коммуникационных сетей осуществляется через маршрутизаторы, которые при необходимости преобразуют указанные протоколы.

Маршрутизаторы обеспечивают достаточно сложный уровень сервиса: выбор наилучшего маршрута для передачи сообщения, адресованного другой сети; управление сбалансированной нагрузкой в сети путем равномерного распределения потоков данных; защиту данных; буферизацию передаваемых данных; различные протокольные преобразования.

Использование маршрутизаторов при объединении ряда небольших локальных сетей в единую сеть дает следующие преимущества (по сравнению с большой ЛВС, имеющей такое же количество абонентских систем):

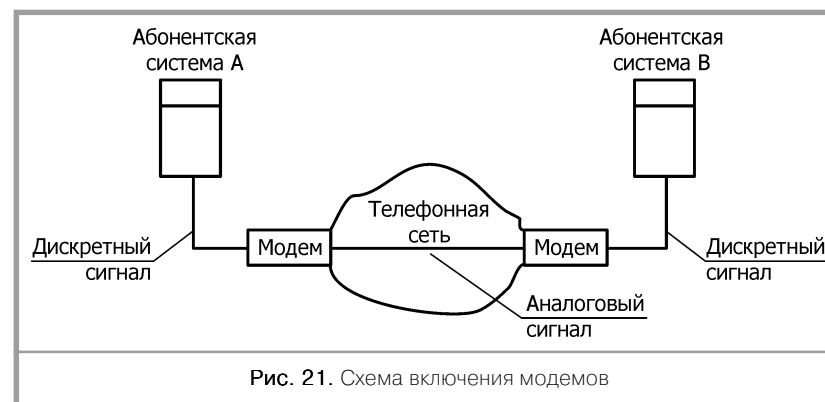
- обеспечивается большая безопасность информации, циркулирующей в сети. В большой ЛВС, работающей в широковещательном режиме, информация распространяется по всей кабельной системе, поэтому несанкционированный доступ к ней проще, чем к ЛВС, образованной из нескольких небольших локальных сетей. В этом случае с помощью маршрутизаторов осуществляется межсетевая коммутация, а обычные сетевые потоки данных остаются локальными, то есть работа в широковещательном режиме возможна только в пределах небольшой ЛВС;
- повышается надежность работы сети – выход из строя одной ЛВС не отражается на работе других взаимосвязанных сетей, так как маршрутизаторы, осуществляющие множественное взаимодействие, изолируют отказавшие сети;
- увеличивается производительность в пределах каждой индивидуальной сети, входящей в состав единой сети. В каждой небольшой ЛВС имеются свои средства управления сетью, повышающие степень ее самостоятельности. Кроме того, уменьшаются нагрузки, связанные с потоком данных, генерируемых рабочими станциями;
- увеличивается диапазон действия сети – выполняя функции усилителей сигнала, маршрутизаторы устраняют ограничение по допустимой протяженности длины кабеля.

3.3. Модем ЛВС

Модем (*modem*) – устройство, которое выполняет функции несущей синусоиды на передающей стороне и демодуляции на приемной стороне (модулятор-демодулятор) в аналоговых каналах связи. Модемы, обеспечивая согласование цифровых сигналов компьютера с аналоговыми сигналами телефонной линии, при передаче данных осуществляют модулирование аналоговых сигналов цифровой информацией, а при приеме – демодулирование. Главное отличие между ними – способ модуляции. Различают модемы с частотной, амплитудной и фазовой модуляцией.

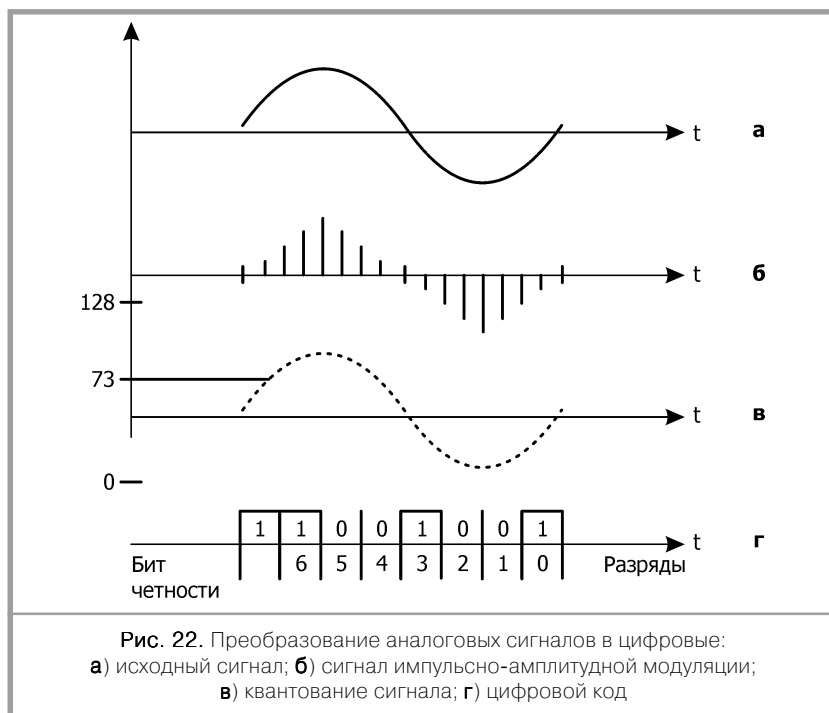
В настоящее время все еще широко используются каналы, по которым передаются аналоговые сигналы. Чаще всего – каналы

старой телефонной сети. Поэтому здесь между абонентскими системами, оперирующими дискретными сигналами, и аналоговыми каналами устанавливается модем (рис. 21). Скорость передачи данных через модемы составляет от 300 бит/с до нескольких десятков килобит в секунду.



Существуют различные методы модуляции. При импульсно-кодовой модуляции существуют три этапа (рис. 22). Сначала аналоговый сигнал «а» представляется в виде множества дискретных значений (отображений) – «б», каждое из которых называется сигналом в импульсно-амплитудной модуляции. Затем каждому сигналу в зависимости от требуемой точности преобразования присваивается определенное числовое значение в диапазоне от 1 до 128 или от 1 до 256. Этот процесс называется квантованием (оцифровыванием) – «в». Полученные числовые значения переводятся в двоичный код «г» для отображения значений разрядов ($2^7 = 128$), а в диапазоне от 1 до 256 – 8 двоичных разрядов ($2^8 = 256$).

Модемы бывают коммутируемые и некоммутируемые, по конструктивному исполнению различают внутренние и внешние модемы, по режиму обмена данными бывают модемы с симплексной, полудуплексной и дуплексной передачей данных. Для системы передачи данных используются факс-модемы, выполняющие и функцию ответов на телефонные звонки, которые позволяют принимать сигналы, осуществлять аналого-дискретное преобразование и сохранять аудиосообщение во внешней памяти компьютера.



3.4. Способы организации ЛВС

Как известно, компьютеры, входящие в состав ЛВС, могут быть расположены самым случайным образом на территории, где создается вычислительная сеть. Следует иметь в виду, что для способа обращения к передающей среде и методов управления сетью небезразлично, как расположены абоненты ЭВМ. Поэтому имеет смысл рассмотреть топологию ЛВС.

Под **топологией** (*topology*) сети понимается конфигурация графа, вершинам которого соответствуют конечные узлы сети (компьютеры), а ребрам – электрические и информационные связи между ними.

Существуют два основных класса топологии вычислительных (компьютерных) сетей:

- широковещательные;
- последовательные.

В широковещательных конфигурациях каждый абонент передает сигналы, которые могут быть восприняты остальными абонент-

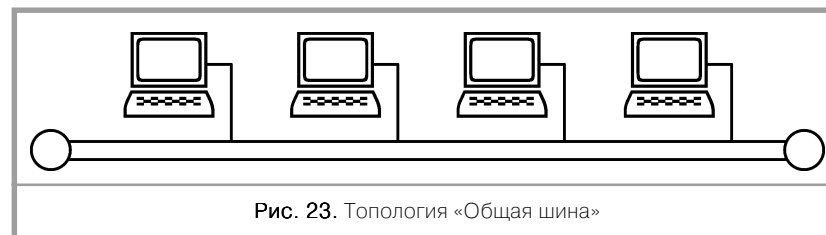
скими системами. К таким конфигурациям относятся общая шина, дерево, звезда с пассивным центром.

В последовательных конфигурациях каждый физический подуровень передает информацию только одной абонентской системе. Отсюда ясно, что широковещательные конфигурации – как правило, ЛВС с селекцией информации, а последовательные – ЛВС с маршрутизацией информации.

Любая топология сети может диктовать не только тип кабеля, но и способ его прокладки и определять метод доступа компьютера в сеть.

Топологии вычислительных (компьютерных) сетей могут быть самыми различными, но для ЛВС типичными являются три: «общая шина», «звезда» и «кольцо».

Топология «общая шина» предполагает использование одного сетевого кабеля, именуемого магистралью, или сегментом, вдоль которого подключены все рабочие сети (рис. 23).



Достоинствами ЛВС шинной топологии являются:

- простота расширения и используемых методов управления;
- экономный (минимальный) расход кабеля;
- недорогая и несложная в использовании среда передачи данных;
- простота системы, отсутствие необходимости в централизованном управлении.

В ЛВС с шинной топологией при передаче пакетов данных каждый компьютер адресует его конкретному компьютеру сети, передавая его по сетевому кабелю в виде электрических сигналов. Пакет в виде электрических сигналов передается по шине в обоих направлениях всем компьютерам сети. Однако информацию принимает только тот адрес, который соответствует адресу получателя, указанному в заголовке пакета. Данные в виде электрических сигналов распространяются по всей сети от одного конца кабеля к другому и, достигая его конца, поглощаются терминатором (заглушкой). Так как в каждый момент времени в сети может вести передачу только

одна рабочая станция, производительность ЛВС зависит от количества РС, подключенных к шине.

Шина – пассивная технология. Это означает, что компьютеры только «слушают» передаваемые по сети данные, но не перемещают их от отправителя к получателю, в связи с чем выход одного компьютера из строя не скажется на работе всей сети.

Однако шинная топология ЛВС имеет и определенные недостатки:

- низкая надежность – любой дефект в кабельной сети полностью ее парализует;
- низкая производительность, так как пропускная способность канала связи определяется всеми узлами сети;
- при значительном объеме трафика уменьшается пропускная способность сети.

При **топологии «звезда»** все компьютеры с помощью сегментов кабеля подключаются к центральному компоненту – концентратору (рис. 24). Пакеты данных от каждого компьютера направляются к центральному концентратору, который, в свою очередь, переправляет их к месту назначения.

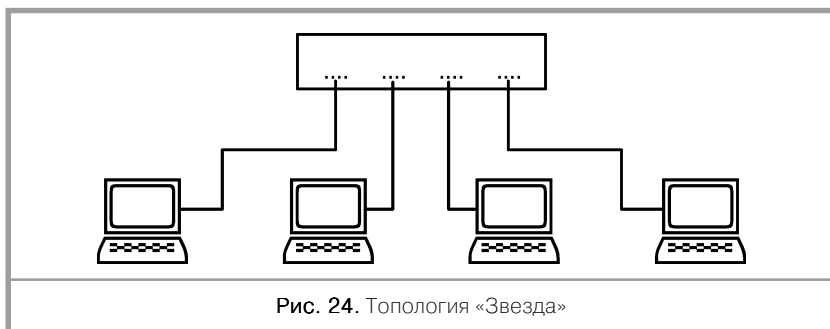


Рис. 24. Топология «Звезда»

В ЛВС с топологией типа «звезда» в центре находится пассивный соединитель или активный повторитель – достаточно простые и надежные устройства. В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети.

Звездообразные ЛВС обычно менее надежны, чем сети с топологией «шина», но они могут быть защищены от нарушений в кабеле с помощью центрального реле, которое отключает вышедшие из строя кабельные лучи. Основные преимущества топологии «звезда»:

- легко модулируются и наращиваются, добавляя новые РС;
- имеют централизованный контроль и управление;
- выход из строя РС не влияет на работу сети.

К недостаткам топологии «звезда» следует отнести:

- более высокую стоимость сетевого оборудования из-за наличия центрального устройства;
- наращивание количества узлов в сети, ограниченное количеством портов концентратора;
- большее количество кабеля для реализации сети, чем «шина» и «кольцо»;
- нарушение работы всей сети при выходе из строя концентратора.

При **топологии «кольцо»** сеть замкнута, образуя неразрывное кольцо (рис. 25), в связи с чем у кабеля канала связи не может быть свободного конца, к которому надо подключать терминатор (заглушку). Начав движение в какой-либо точке кольца, пакет данных в конце концов попадает в его начало. Из-за такой особенности данные в кольце движутся всегда в одном направлении. При перемещении кадра по кольцу каждая РС принимает кадр, анализирует его адресное поле, снимает копию кадра, если он адресован данной РС, ретранслирует кадр. Естественно, что все это замедляет передачу данных в кольце, причем длительность задержки определяется числом РС. Удаление кадра из кольца производится обычно станцией-отправителем. В этом случае кадр совершает по кольцу полный круг и возвращается к станции-отправителю, которая воспринимает его как квитанцию – подтверждение получения кадра адресатом. Удаление кадра из кольца может осуществляться и станцией-получателем, тогда кадр не совершает полного круга, а станция-отправитель не получает квитанции – подтверждения.

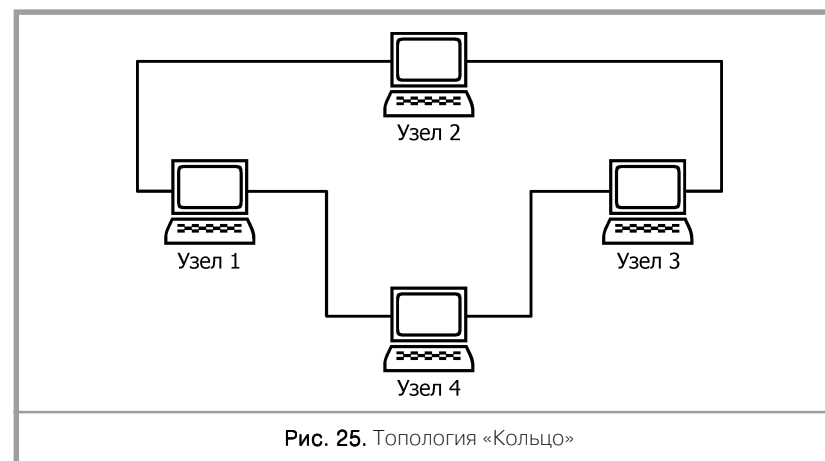


Рис. 25. Топология «Кольцо»

Кольцевая структура обеспечивает довольно широкие функциональные возможности ЛВС при высокой эффективности использования моноканала, низкой стоимости, простоте методов управления, возможности контроля работоспособности моноканала.

Основные достоинства топологии «кольцо» ЛВС:

- все РС имеют равный доступ;
- количество пользователей не сказывается на производительности;
- низкая стоимость;
- резервирование связи.

В отличие от пассивной топологии «шина», в «кольце» каждый компьютер выступает в роли репитера (повторителя), усиливая сигналы и передавая их следующему компьютеру.

К недостаткам кольцевой топологии ЛВС относятся:

- выход из строя одной РС выводит из строя всю сеть;
- изменение или модернизация конфигурации сети требует остановки всей сети;
- уязвимость в отношении отказов как последовательная конфигурация сети;
- слабая защищенность: данные проходят через каждый сетевой компьютер, давая возможность для перехвата информации.

Таким образом, для ЛВС можно выделить следующие характерные признаки:

- относительная простота конфигурации сети;
- использование высокоскоростных цифровых каналов передачи данных;
- высокий уровень взаимодействия пользователей сети;
- размещение сети на ограниченной территории, на которой замыкаются все основные информационные потоки;
- сравнительно невысокая стоимость сетевого оборудования, в том числе сетевых адаптеров.

3.5. Методы доступа в ЛВС

При использовании любой топологии, когда два компьютера одновременно начнут передавать данные, в сети происходит столкновение данных (коллизия), которое препятствует правильной передаче данных по сети. Вероятность возникновения коллизии зависит от интенсивности сетевого трафика.

Для решения этой проблемы служат **методы доступа** – набор правил, по которым РС узнает, когда линия связи свободна и можно передавать данные. Наибольшее распространение при проектировании и построении ЛВС получили два метода:

- 1) множественный метод с контролем несущей и обнаружением коллизий (CSMA/CD – Carrier-Sense Multiple Access and Collision Defection);
- 2) доступ с передачей маркера.

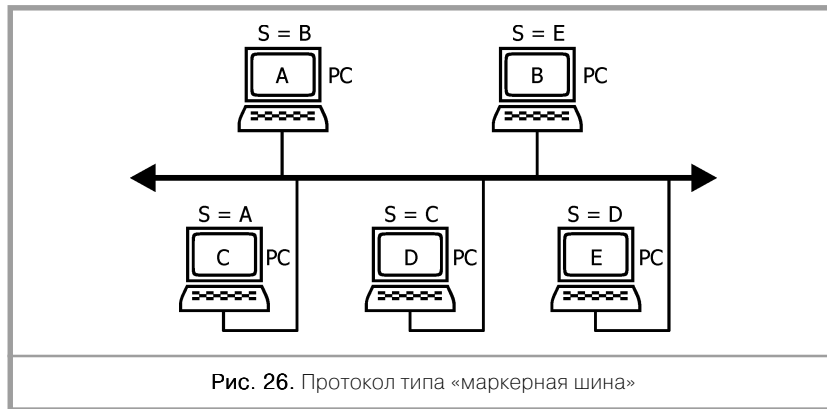
Первый метод CSMA/CD применяется исключительно в сетях с логической общей шиной (к которым относятся и радиосети, породившие этот метод). Все компьютеры такой сети имеют непосредственный доступ к общей среде, поэтому она может быть использована для передачи данных между любыми двумя узлами сети. Одновременно все компьютеры сети имеют возможность немедленно (с учетом задержки распространения сигнала по физической среде) получить данные, которые любой из компьютеров начал передавать в общую среду. Алгоритм рабочей станции, а точнее ее сетевого адаптера, при использовании метода доступа CSMA/CD заключается в следующем:

- а) рабочая станция прослушивает канал, стремясь обнаружить чью-либо передачу данных;
- б) если слышит чью-либо передачу, ожидает ее окончания;
- в) если канал свободен, начинает передачу пакета;
- г) при обнаружении коллизии вовремя передачу прекращает передачу;
- д) через случайный промежуток времени все повторяется, то есть осуществляет переход к прослушиванию канала.

Второй метод маркерного доступа заключается в том, что пакет особого типа (маркер) перемещается по замкнутому кругу, минуя по очереди все РС, пока его не получит тот, который хочет передать данные.

Метод передачи маркера широко используется в сетях магистральной (шинной), звездообразной и кольцевой топологий. Он относится к классу селективных методов: право на передачу данных станции получают в определенном порядке, задаваемом с помощью маркера, который представляет собой уникальную последовательность бит информации (уникальный кадр). Магистральные сети, использующие этот метод, называются сетями типа «**маркерная шина**», а кольцевые сети – сетями типа «**маркерное кольцо**».

В сетях типа «**маркерная шина**» (рис. 2б) доступ к каналу обеспечивается таким образом, как если бы канал был физическим кольцом, причем допускается использование канала кольцевого (шинного, звездообразного). Маркер (управляющий кадр) содержит адресное поле, где записывается адрес станции, которой предоставляется право доступа в канал. Станция, получив маркер со своим адресом, имеет исключительное право на передачу данных (кадра) по физическому каналу. После передачи кадра станция отправляет маркер

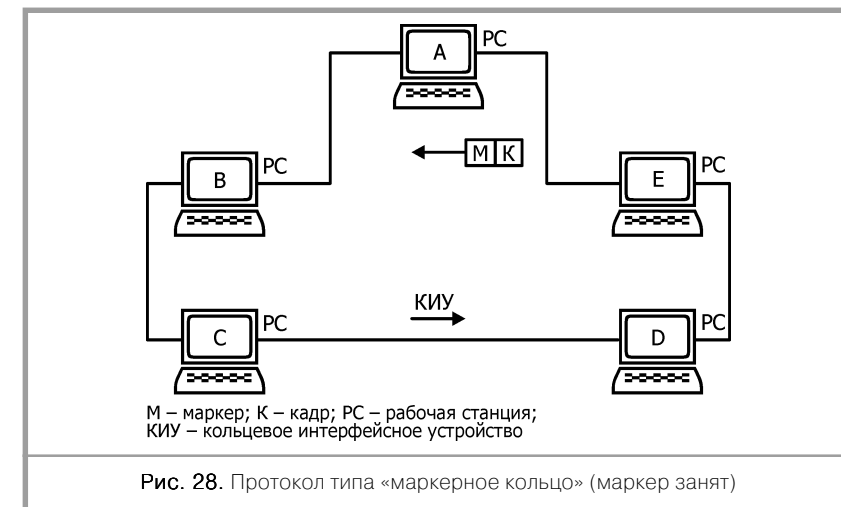
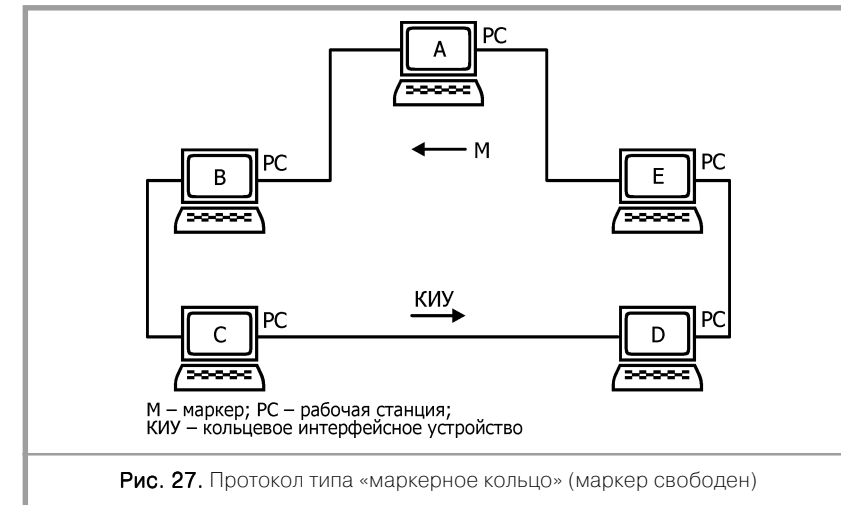


другой станции, которая является очередной по установленному порядку владения правом на передачу. Станции получают маркер в циклической последовательности, при этом в физической шине формируется так называемое логическое кольцо. Работая в режиме прослушивания работы канала, принять передаваемый кадр может только та станция, адрес которой указан в поле адреса получателя этого кадра.

Протокол типа «**маркерное кольцо**» применяется в сетях с кольцевой топологией, которые относятся к типу сетей с последовательной конфигурацией, где широкоэвещательный режим работы невозможен. В таких сетях сигналы распространяются через РС. В отличие от сетей с шинной структурой, где РС действуют только как передатчики или приемники и отказ РС не влияет на передачу данных к другим РС, в кольцевой структуре при передаче данных все РС играют активную роль, участвуя в ретрансляции, анализе и модификации проходящих сигналов.

В протоколе «маркерное кольцо» маркер не имеет адреса. Он снабжается полем занятости, в котором записывается. Один из кодов, обозначающий состояние маркера – свободное или занятое. Если ни один из РС сети не имеет данных для передачи, свободный маркер циркулирует по кольцу, совершая однонаправленное перемещение (рис. 27). В каждой РС маркер задерживается на время, необходимое для его приема, анализа (с целью установления занятости) и ретрансляции. В выполнении этих функций задействованы кольцевые интерфейсные устройства (КИУ).

Получив свободный маркер, РС, готовая к передаче кадра с данными, меняет состояние маркера на «занятый», передает его дальше по кольцу и добавляет к нему кадр (рис. 28). Занятый маркер



вместе с кадром совершает полный оборот по кольцу и возвращается к станции-отправителю. По пути станция-получатель, удостоверившись по адресной части кадра, что именно ей он адресован, снимает копию с кадра. Изменить состояние маркера снова на свободный может та РС, которая изменила его на занятый. По возвращении занятого маркера с кадром данных к РС-отправителю кадр

удаляется из кольца, а состояние маркера меняется на свободное. Маркер перемещается с большой скоростью. Например, в кольце диаметром 200 м маркер может циркулировать с частотой 10 000 оборотов в секунду.

3.6. Централизованные и одноранговые ЛВС

Как уже отмечалось выше, централизованные и одноранговые ЛВС обладают определенными характеристиками и областью применения.

Централизованная сеть с выделенным сервером, который выполняет функции хранения данных, предназначенных для использования всеми рабочими станциями, управления взаимодействием между РС и ряд функций, в полной мере не отвечает высоким требованиям к надежности процесса обработки данных в многопользовательском режиме.

Достоинства сети:

- надежная система защиты информации;
- высокое быстродействие;
- отсутствие ограничений на число рабочих станций без учета производительности трафика;
- простота управления по сравнению с одноранговыми сетями.

Недостатки сети:

- высокая стоимость из-за выделения одного компьютера под сервер;
- зависимость быстродействия и надежности сети от сервера;
- ограниченное число РС, влияющее на производительность трафика;
- меньшая гибкость по сравнению с одноранговой сетью.

Сети с выделенным сервером являются наиболее распространенными у пользователей с сетевыми операционными системами – LAN Server (IBM), Windows NTServer и NetWare (Novell).

Одноранговые сети обеспечивают информационно-вычислительные ресурсы, которые распределены по большому числу абонентских систем. В такой сети нет единого центра управления взаимодействием РС и нет единого устройства для хранения данных. Сетевая операционная система распределена по всем рабочим станциям. Каждая станция сети может выполнять функции как клиента, так и сервера. Она может обслуживать запросы от других рабочих станций и направлять свои запросы на обслуживание в сеть. Пользователю сети доступны все устройства, подключенные к другим станциям (диски, принтеры).

Достоинства одноранговых сетей:

- низкая стоимость;
- надежность.

Недостатки одноранговых сетей:

- зависимость эффективности работы сети от количества станций;
- сложность управления сетью;
- трудности в модификации и изменении программного обеспечения станций.

Наиболее популярными являются одноранговые сети на базе сетевых операционных систем LANtastic, NetWareLite.

Особое внимание следует уделять одному из типов серверов – файловому серверу (FileServer) – файл-сервер.

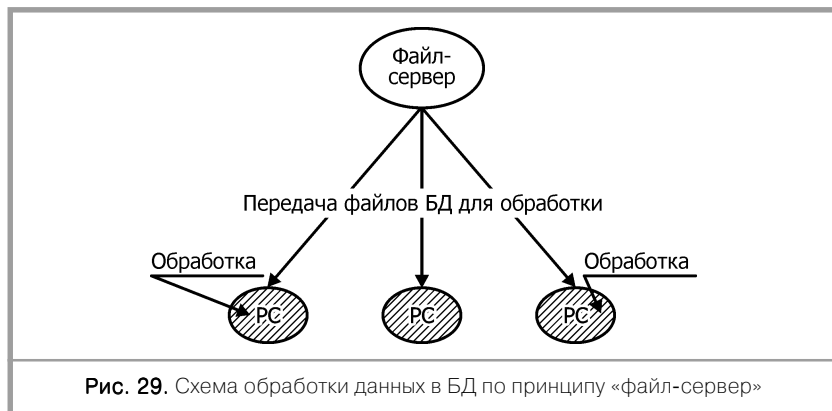
Файл-сервер хранит данные пользователей сети и обеспечивает им доступ к этим данным. Это компьютер с большой емкостью оперативной памяти, жесткими дисками большой емкости и дополнительными накопителями на магнитной ленте (стример). Он работает под управлением специальной операционной системы, которая обеспечивает одновременный доступ пользователей сети к расположенным на нем данным.

Файл-сервер выполняет следующие функции: хранение данных, архивирование данных, синхронизацию изменений данных различными пользователями, передачу данных.

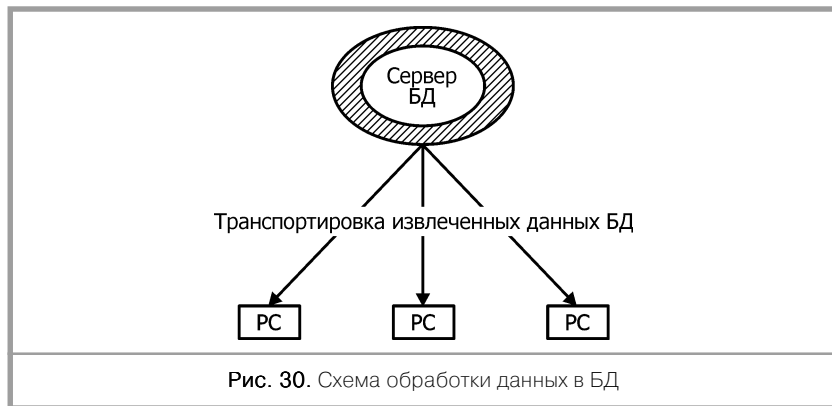
Архитектура систем баз данных (БД) с сетевым доступом предполагает выделение файл-сервера (сервер-файлов). На такой машине хранится совместно используемая централизованная БД. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном производится обработка. Концепция файл-сервера условно отображена на рис. 29.

Компьютерные сети реализуют распределенную обработку данных между двумя объектами: клиентом и сервером (рис. 8). В процессе обработки данных клиент может сформировать запрос на сервер для выполнения сложных процедур, чтения файла, поиска информации в базе данных и т. д. Сервер, определенный ранее, выполняет запрос, поступивший от клиента. Результаты выполнения запроса передаются клиенту. Сервер обеспечивает хранение данных общего пользования, организует доступ к этим данным и передает данные клиенту.

Клиент перерабатывает полученные данные и представляет результаты обработки в виде, удобном для пользователя. Для подобных систем приняты термины – системы «клиент-сервер» или архитектура «клиент-сервер». Спецификой архитектуры клиент-сервер



является использование языка запросов SQL. Концепция клиент-сервера условно отображена на рис. 30.



3.7. Построение ЛВС на базе ОС Windows

Операционная система **Windows** – однопользовательская операционная система корпорации Microsoft для ЛВС.

Windows является 32-разрядной многозадачной и многопоточной операционной системой, характеризующейся удобным графическим интерфейсом. Создана система для работы с персональными компьютерами ЛВС. Используя Windows, пользователь может эффективно работать сразу с несколькими прикладными процессами.

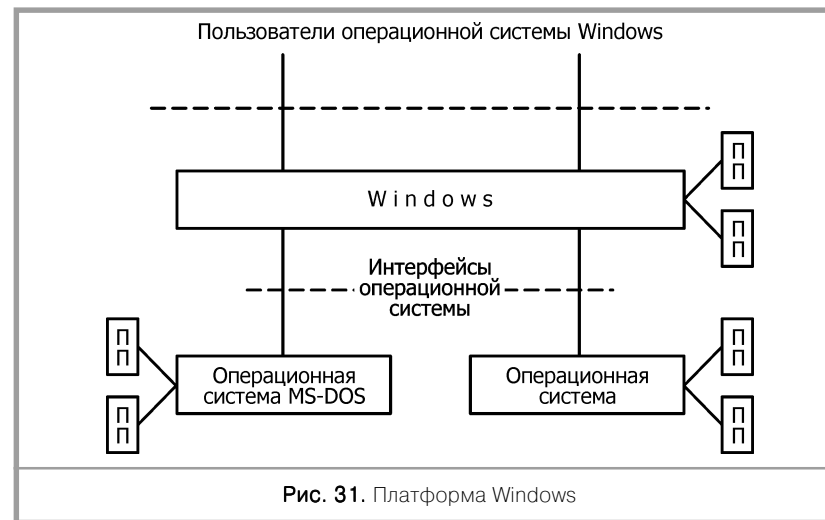
Windows выполняет более 600 функций полиэкранной технологии, снимая с программистов большую часть их работы. Операционная система содержит также обширные средства работы в сетях.

Windows характеризуется рядом важных особенностей:

- модульной расширяемостью архитектуры;
- способностью работать на многих аппаратных платформах;
- широким использованием средств аудио-, видеосистем.

Для операционной системы Windows фирма Microsoft предлагает ряд платформ и стандартных прикладных процессов. В их число входит комплекс программ Windows for Workgroups, который предназначен для коллективной работы абонентских систем в ЛВС.

Windows может также использоваться в качестве операционной платформы (рис. 31), располагаясь над другими операционными системами. В платформе выделяются два типа стандартных интерфейсов.



К ним относится интерфейс операционной системы. Прикладные процессы (ПП) работают с любой из указанных систем, используя интерфейс прикладной программы (API). Развитием Windows стала сетевая операционная система **Windows NT**, которая предназначена для архитектуры клиент-сервер ЛВС и обеспечивает независимо от конкретного типа сети возможность использования различных протоколов транспортного уровня.

Операционная система является многозадачной (выполняет сразу несколько прикладных программ, используя идеи операционной системы UNIX). Основными целями рассматриваемой системы являются блочность структуры, легкая расширяемость, мигрируемость (переход от одной аппаратной платформы к другой), предоставление удобных интерфейсов пользователя. Система организует несколько параллельно обрабатываемых потоков данных.

Система имеет 32-разрядную архитектуру и поддерживает локальные сети с широким кругом выполняемых функций. К ним в первую очередь относятся:

- возможность каждой абонентской системы в сети быть сервером либо клиентом;
- совместная работа группы пользователей;
- предоставление информации в удобной и понятной форме;
- секретность большого числа файлов;
- адресация большой памяти;
- запуск одновременно нескольких прикладных процессов;
- поддержка мультипроцессорных систем;
- управление сетью;
- обеспечение безопасности данных.

Операционная система погружается в прикладную среду и работает с большим набором типов микропроцессоров корпорации Intel, корпорации IBM, корпорации DEC.

Фирмой Novell была разработана операционная система NetWare. Она обеспечивает работу клиентов сети с одним либо группой серверов, позволяет создавать ассоциации локальных сетей.

Она может использоваться в любой распространенной в настоящее время физической структуре ЛВС: Token Ring, Ethernet или ARCnet. Поэтому сеть, поддерживаемая сетевой ОС NetWare, может иметь шинную, кольцевую и звездообразную топологии. ЛВС фирмы Novell представляет собой сеть шинной топологии, для реализации которой используется аппаратура Ethernet. Она является платформой, предоставляющей разнообразные сетевые службы. Структура сетевой операционной системы NetWare приведена на рис. 32. Ее можно рассматривать как распределенную операционную систему, модули которой располагаются на файл-сервере и на рабочих станциях.

Основным ресурсом ЛВС Novell является файл-сервер. На нем размещаются сетевая операционная система, базы данных и прикладные программы пользователей.

Сетевая операционная система NetWare представляет собой мультизадачную ОС реального времени. Она ориентирована на работу в локальной сети с централизованным управлением. В NetWare

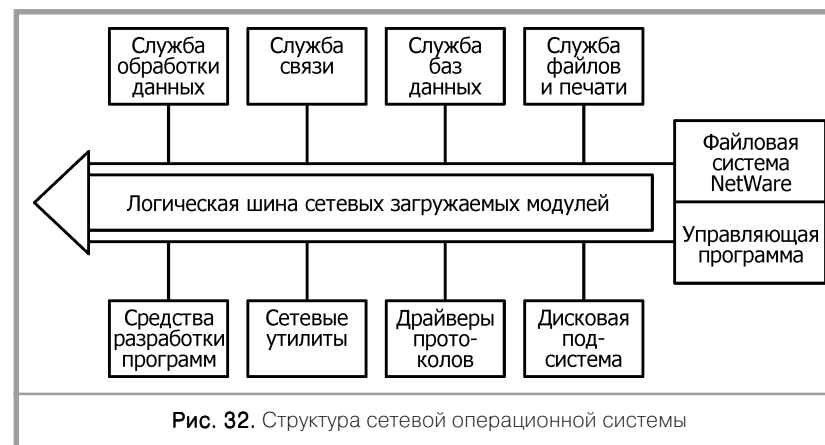


Рис. 32. Структура сетевой операционной системы

предусмотрена возможность работы в сети одного или нескольких файл-серверов.

На файл-сервере располагается ядро операционной системы, а остальные ее компоненты размещены на рабочих станциях.

Сетевые операции реализуются по технологии клиент-сервер. Это значит, что все разделяемые ресурсы управляются централизованно через файл-сервер. Ресурсы доступны рабочим станциям-клиентам только по запросу, трансформированному через сеть на сервер. Ресурсы рабочих станций доступны только пользователям самих станций и в сети не разделяются.

Система протоколов TCP/IP



4.1. Назначение и организация стека TCP/IP

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют две стороны, то есть в данном случае необходимо организовать согласованную работу двух «иерархий», работающих на разных компьютерах. Процедура взаимодействия двух узлов ВС может быть описана в виде набора правил взаимодействия каждой пары компьютеров соответствующих уровней обеих участвующих сторон.

Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются **протоколами**.

Модули, реализующие протоколы соседних уровней и находящиеся в одном узле, также взаимодействуют друг с другом в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений – эти правила называются **интерфейсом**.

В сущности, протокол и интерфейс выражают одно и то же понятие, но традиционно в сетях за ними закрепили разные области действия:

- протоколы определяют правила взаимодействия модулей одного уровня в разных узлах;
- интерфейсы определяют правила взаимодействия модулей соседних уровней в одном узле.

Средства каждого уровня должны обрабатывать:

- во-первых, свой собственный протокол;
- во-вторых, интерфейсы с соседними уровнями.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называются **стеком коммуникационных протоколов**.

В настоящее время в сетях используется большое количество стеков коммуникационных протоколов. Наиболее популярными в настоящее время являются:

- стек TCP/IP (протокол управления передачей/межсетевой протокол) – Transmission Control Protocol/Internet Protocol, пара взаимосвязанных протоколов сетевого и транспортного уровней;
- стек IPX/SPX (протоколы сетевого и сеансового уровней) – Internetwork Packet Exchange/Sequenced Packet Exchange, разработан для сетевой операционной системы NetWare фирмы Novell, популярность которой сейчас значительно уступает операционным системам Microsoft;
- стек NetBIOS/SMB – Network Basic Input/Output System, широко используется в продуктах компании IBM и Microsoft, на физическом и канальном уровнях которого задействованы все наиболее распространенные протоколы Ethernet, Token Ring, FDDI и др.

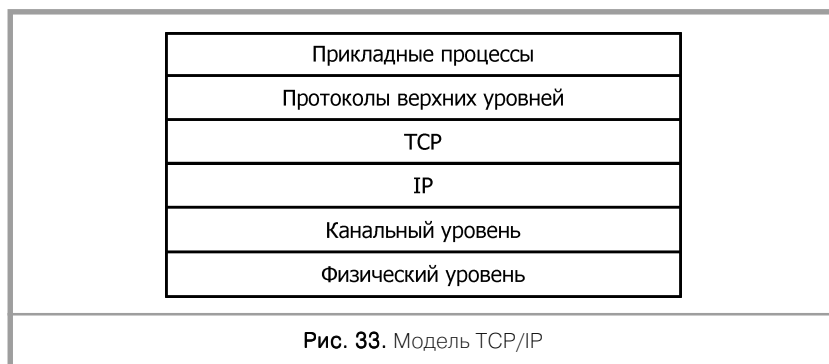
Стек TCP/IP был разработан по инициативе Министерства обороны США Агентством перспективных оборонных исследовательских проектов (DAPRA) для связи экспериментальной сети ARPAnet с другими сетями как набор общих протоколов для разнородной вычислительной среды. Большой вклад в развитие стека TCP/IP, который получил свое название по протоколам IP и TCP, внес университет Беркли, реализовав протоколы стека в своей версии ОС UNIX.

Стек TCP/IP на нижнем уровне поддерживает все популярные стандарты физического и канального уровней: для ЛВС – это Ethernet, Token Ring, FDDI, для ГС – протоколы работы на аналоговых коммутируемых и выделенных линиях SLIP, PPP, протоколы территориальных сетей X.25 и ISDN.

Протокол TCP/IP располагается между протоколами верхних уровней и канальным уровнем (рис. 33).

Протокол TCP организует создание виртуальных каналов, проходящих через коммуникационную сеть. В соответствии с этим TCP относят к транспортному уровню области взаимодействия открытых систем (OSI).

Протокол IP ориентирован на использование одиночных пакетов, именуемых датаграммами. Datagram – дейтаграмма – для обо-



значения блоков данных определенных уровней специального названия кадр (frame), пакет (packet), сегмент (segment). Его задачей являются обеспечение взаимодействия друг с другом и выполнение процессов, связанных с коммутацией и маршрутизацией. Для этого IP передает датаграммы из одной сети в другую. IP относят к сетевому уровню.

Задачей TCP является предоставление сервиса передачи датаграмм при гарантии упорядоченной доставки последовательностей блоков данных, несмотря на возможные их повреждения, потери, дублирования, нарушения последовательности. TCP имеет три фазы работы:

- установление соединения;
- передача по нему датаграмм;
- разъединение соединения.

Так как межсетевой протокол IP ненадежен, то TCP является сложным протоколом, обеспечивающим высокую степень надежности передачи данных.

Протокол IP осуществляет следующую реализацию коммуникационных аспектов:

- присвоение, контроль и преобразование имен объектов сети;
- сообщения о состояниях (недостижимость адресов, ошибки и запросы повторных вызовов);
- обеспечение обмена данными через шлюзы;
- управление передачей и сбор данных о работе сетей;
- изменение размеров передаваемых датаграмм (их фрагментация).

Успех продуктов TCP/IP связан с тем, что благодаря современному техническому развитию микропроцессоров стала возможной и эффективная реализация. Хотя протоколы TCP/IP неразрывно

связаны с Интернетом и каждый из многомиллионной армады компьютеров Интернета работает на основе этого стека, существует большое количество локальных, корпоративных и территориальных сетей, непосредственно не являющихся частями Интернета, в которых также используются протоколы TCP/IP. Очень полезным свойством, делающим возможным применение стека TCP/IP в больших сетях, является его возможность фрагментировать пакеты, что позволяет при переходе одной сети, имеющей большую максимальную длину, в сеть меньшей максимальной длиной проводить деление передаваемого кадра на несколько частей.

Другой особенностью технологии TCP/IP является гибкая система адресации, позволяющая проще, чем другие протоколы аналогичного назначения, включать в интернет сети разных технологий. В стеке TCP/IP очень экономно используются возможности широковещательных рассылок. Это свойство совершенно необходимо при работе на медленных каналах связи, характерных для территориальных сетей.

Однако к недостаткам технологии TCP/IP следует отнести высокие требования к ресурсам и сложность администрирования IP-сетей. Мощные функциональные возможности протоколов стека TCP/IP требуют для своей реализации больших вычислительных затрат. Гибкая система адресации и отказ от широковещательных рассылок приводят к наличию в IP-сети разнообразных централизованных служб типа DNS (Domain Name System – специальная распределенная служба доменных имен), DHCP (Dynamic Host Configuration Protocol – протокол динамической конфигурации «хоста» (конечный узел Интернета)) и т. п. Каждая из этих служб направлена на обеспечение администрирования сети, в том числе и на облегчение конфигурирования оборудования, но в то же время сама требует пристального внимания со стороны администраторов.

Можно приводить и другие доводы за и против стека TCP/IP, однако факт остается фактом – сегодня это самый популярный стек протоколов, широко используемый как в глобальных, так и в локальных сетях.

4.2. Многоуровневая структура стека TCP/IP

В стеке TCP/IP определены 4 уровня (рис. 34). Каждый из этих уровней несет на себе некоторую нагрузку по решению основной задачи – организации надежной и производительной работы составной сети, части которой построены на основе разных сетевых технологий.

1-й уровень. **Прикладной** (application) уровень стека TCP/IP соответствует трем верхним уровням модели OSI: прикладному, пред-

Уровни OSI	№ уровня стека	Уровни стека	
Прикладной (Application)	Уровень I	Прикладной (Application)	FTP, telnet, SMTP, HTTP
Представительский (Presentation)			
Сеансовый (Session)			
Транспортный (Transport)	Уровень II	Транспортный (Transport)	
Сетевой (Network)	Уровень III	Сетевой (Network)	
Канальный (Link)	Уровень IV	Уровень сетевых интерфейсов (Network Interface)	Протоколы инкапсуляции и преобразования адресов
Физический (Physical)			

Рис. 34. Многоуровневая структура стека TCP/IP

ставительскому и сеансовому. Он объединяет службы, предоставленные системой пользовательским приложениям. Протоколы прикладного уровня устанавливаются на «хостах», то есть конечном узле в Интернете. Прикладной уровень реализуется программными системами, построенными в архитектуре клиент-сервер. В отличие от протоколов других трех уровней, протоколы прикладного уровня занимаются деталями конкретного приложения и «не интересуются» способами передачи данных по сети, они обращаются к протоколам нижних уровней как к некоторому набору инструментов. Так, клиентская часть протокола прикладного уровня для обмена сообщениями со своей серверной частью, установленной на отдельном узле составной сети, должна обратиться с запросом к нижележащему транспортному уровню.

Этот уровень постоянно расширяется за счет присоединения к нему широко используемых протоколов сетевых служб типа Telnet – протокола эмуляции (преобразования характеристик реального терминала в характеристики, принятые в информационной сети) терминала, FTP (File Transfer Protocol) – протокола копирования файлов, SMTP (Simple Mail Transfer Protocol) – простого протокола передачи электронной почты, HTTP (Hypertext Transfer Protocol) – протокола передачи гипертекстовой информации и многих других.

2-й уровень. **Основной**, или **транспортный** (*transport*), уровень стека TCP/IP может предоставлять вышеизложенному уровню два типа сервиса:

- гарантированная доставка – протокол управления передачей (Transmission Control Protocol) – TCP;

- доставка по возможности («best effort») – протокол пользовательских дейтаграмм (датаграмм) (User Datagram Protocol) – UDP.

FTP – File Transfer Protocol – протокол передачи файлов.

SMTP – Simple Mail Transfer Protocol – простой протокол передачи почты.

SNMP – Simple Network Management Protocol – простой протокол управления сетью.

HTTP – HyperText Transfer Protocol – протокол передачи гипертекста.

Telnet – служба, которая осуществляет сервис виртуального терминала.

Протоколы инкапсуляции – способ передачи пакетов немаршрутизируемых протоколов в пакеты какого-либо сетевого протокола.

Архитектура ARPANET – информационная сеть Агентства перспективных исследований проектов (DARPA) МО США.

Уровни	Прикладные процессы
5–7	<ul style="list-style-type: none"> • Telnet • FTP • SMTP
4	TCP
3	IP
2	Канальный уровень
1	Физический уровень

Поскольку на сетевом уровне не устанавливаются соединения, то нет никаких гарантий, что все пакеты будут доставлены в место назначения целыми и невредимыми или придут в том же порядке, в котором они были отправлены. Эту задачу – надежной информационной связи между двумя отдельными узлами – решает основной (транспортный) уровень стека TCP/IP.

Первый протокол TCP – «гарантированная доставка» – обеспечивает надежную передачу сообщений между удаленными прикладными процессами за счет образования логических соединений, что позволяет ему нумеровать пакеты, подтверждать их прием квитанциями, в случае потери организовывать повторные передачи, распознавать и уничтожать дубликаты, доставлять прикладному уровню пакеты в том порядке, в котором они были отправлены.

Данный протокол позволяет равноправным объектам на компьютере-отправителе и компьютере-получателе поддерживать обмен данными в дуплексном режиме. TCP дает возможность без ошибок доставить сформулированный на одном из компьютеров поток байтов с любого другого компьютера, входящего в составную сеть. TCP

делит поток байтов на части – сегменты и передает их нижележащему уровню межсетевому взаимодействию. После того как эти сегменты будут доставлены средствами уровня межсетевому взаимодействию в пункт назначения, протокол TCP снова соберет их в непрерывный поток байтов.

Второй протокол этого уровня UDP – доставка «по возможности» – простейшим дейтаграммным протоколом, который используется в том случае, когда задача надежного обмена данными либо вообще не ставится, либо решается средствами более высокого уровня – системными прикладными службами или пользовательскими приложениями.

Таким образом, этот поток обеспечивает передачу прикладных пакетов дейтаграммным способом, как и главный протокол уровня межсетевому взаимодействию IP, и выполняет только функции связующего звена (мультиплексора) между сетевым протоколом и многочисленными службами прикладного уровня или пользовательскими процессами. От прикладного протокола транспортный уровень принимает задачу на передачу данных с тем или иным качеством, после выполнения рапортует об этом прикладному уровню. К нижележащему уровню межсетевому взаимодействию протоколы TCP и UDP обращаются как к своего рода инструменту, не очень надежному, но способному перемещать пакеты в свободном и рискованном путешествии по составной сети. Протоколы TCP и UDP, так же как и протокол прикладного уровня, устанавливаются на хостах (конечных узлах Интернета).

3-й уровень. **Уровень межсетевому взаимодействию** (*internet*), называемый также **сетевым** (*network*) уровнем, является стержнем всей архитектуры TCP/IP.

Именно этот уровень, функции которого соответствуют сетевому уровню модели OSI, обеспечивает перемещение пакетов в пределах всей составной сети. Протоколы уровня межсетевому взаимодействию поддерживают интерфейсы с вышележащим транспортным уровнем, получая от него запросы на передачу данных по составной сети. Он реализует концепцию передачи пакетов в режиме без установления соединений, то есть дейтаграммным способом, и обеспечивает возможность перемещения пакетов по сети, используя тот маршрут, который в данный момент является наиболее рациональным.

Основным протоколом межсетевому уровня (в терминах модели OSI) является межсетевому протокол IP (Internet Protocol). В его задачу входит продвижение пакета между подсетями – от одного программного маршрута до другого, до тех пор, пока пакет не попадет в сеть назначения. В отличие от протоколов «прикладного» и «основного» уровней, протокол IP устанавливается не только на хостах, но и на всех шлюзах (маршрутизаторах в Интернете).

Протокол IP – это дейтаграммный протокол, работающий без установления соединений по принципу «по возможности», в соответствии с которым он не берет на себя ответственность за доставку пакета до узла назначения. Если же по каким-то причинам пакет теряется (например, из-за переполнения буфера), протокол IP не пытается повторить его передачу. Максимум, на что он способен, – послать уведомление о потере пакета узлу-отправителю.

Протокол IP изначально проектировался как хорошо масштабируемое средство передачи пакетов в составных сетях, состоящих из большого количества сетей, объединенных как локальными, так и глобальными связями. Учитывая, что между двумя узлами сети может пролегать несколько возможных путей, задача перемещения данных в составной сети включает задачу прокладки и выбора маршрутов.

Протоколы, связанные с составлением и модификацией таблиц маршрутизации, такие как протоколы сбора маршрутной информации RIP (Routing Internet Protocol) и OSPF (Open Shortest Path First), также относятся к уровню межсетевому взаимодействию.

На этом же уровне работает и протокол межсетевых управляющих сообщений ICMP (Internet Control Message Protocol), предназначенный для обмена информацией об ошибках между маршрутизаторами сети и узлом – источником пакета. С помощью специальных пакетов ICMP сообщает о невозможности доставки пакета, о превышении времени жизни или продолжительности сборки пакета из фрагментов, об аномальных величинах параметров, об изменении маршрута пересылки и типа обслуживания, о состоянии системы и т. п.

Выбор последовательности шлюзов, через которые надо передать пакет, чтобы он дошел до места назначения, – это задача протоколов уровня межсетевому взаимодействию. А вот перемещение пакета между соседними шлюзами в пределах каждой из встречающихся на пути пакета подсетей – это уже задача локальной (то есть использующейся в каждой из подсетей) технологии. Каждый раз, когда требуется воспользоваться локальными средствами доставки пакета в пределах подсети, протокол IP обращается к нижележащему уровню межсетевых интерфейсов.

Протоколы маршрутизации (например, RIP или OSPF) следует отличать от собственно сетевых протоколов (например, IP или IPX). В то время как первые собирают и передают по сети чисто служебную информацию о возможных маршрутах, вторые предназначены для передачи пользовательских данных.

4-й уровень. **Уровень сетевых интерфейсов** (*network interface*) является интерпретацией функций самого нижнего уровня. Протоколы этого уровня должны обеспечивать интеграцию в составную

сеть других сетей, причем сеть TCP/IP должна иметь средства включения себя в любой другой сети, какую бы внутреннюю технологию передачи данных эта сеть ни использовала. Нижний уровень стека TCP/IP отвечает только за организацию интерфейса с частными технологиями подсетей. Упрощенная задача обеспечения интерфейса между двумя технологиями сводится, во-первых, к определению способа упаковки («инкапсуляции») пакета IP в единицу передаваемых данных промежуточной сети, а во-вторых, к определению способа преобразования сетевого адреса следующего шлюза в новый тип адреса, который принят для адресации узлов в технологии данной промежуточной сети.

4.3. Сетезависимые и сетезависимые уровни стека TCP/IP

Для каждой технологии, включаемой в составную сеть подсети, должны быть разработаны собственные интерфейсные средства. К таким интерфейсным средствам относятся протоколы инкапсуляции IP-пакетов уровня межсетевого взаимодействия в кадры локальных технологий. Отсюда следует, что этот уровень нельзя определить раз и навсегда.

Уровень сетевых интерфейсов в стеке TCP/IP не регламентируется, но он поддерживает все популярные технологии физического и канального уровней. Для локальных сетей это Ethernet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, для глобальных сетей – протоколы соединений «точка-точка» SLIP и PPP, протоколы территориальных сетей с коммутацией пакетов X.25, frame relay, ATM.

Рассматривая многоуровневую структуру TCP/IP, можно выделить в ней, подобно архитектуре OSI, уровни которой зависят от конкретной технической реализации сети, и уровни, функции которых ориентированы на работу с приложениями (рис. 35).

Протоколы прикладного уровня стека TCP/IP работают на компьютерах, выполняющих приложения пользователей. Даже полная смена сетевого оборудования в общем случае не должна влиять на работу приложений, если они получают доступ к сетевым возможностям через протоколы прикладного уровня.

Протоколы транспортного уровня зависят от сети уже в большей степени, так как они реализуют интерфейс к уровням, непосредственно организующим передачу данных по сети. Однако, подобно протоколам прикладного уровня, протоколы транспортного уровня устанавливаются только на конечных узлах. Протоколы двух нижних уровней являются сетезависимыми, программные модули протоколов межсетевого уровня и уровня сетевых интерфейсов устанавливаются на всех хостах и шлюзах.

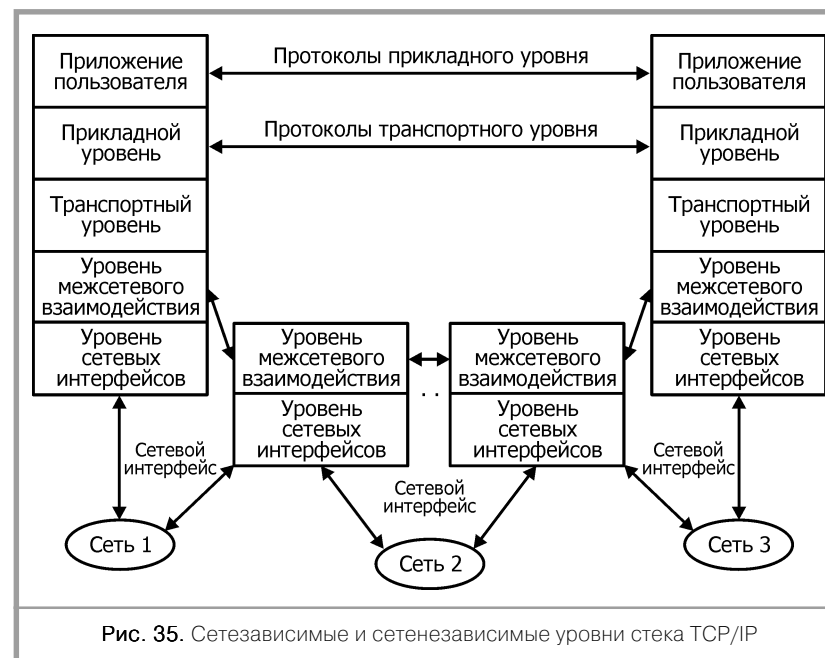


Рис. 35. Сетезависимые и сетезависимые уровни стека TCP/IP

Каждый коммуникационный протокол оперирует с некоторой единицей передаваемых данных. Названия этих единиц иногда закрепляются стандартом, а чаще просто определяются традицией. В стеке TCP/IP за многие годы его существования образовалась устоявшаяся терминология в этой области (рис. 36).

Протоколом (*stream*) называют данные, поступающие от приложений на вход протоколов транспортного уровня TCP и UDP. Протокол TCP «нарезают» из потока данных «сегменты» (*segment*).

Единицу данных протокола UDP часто называют дейтаграммой, или датаграммой (*datagram*).

Дейтаграмма – это общее название для единиц данных, которыми оперируют протоколы без установления соединений. К таким протоколам относится и протокол межсетевого взаимодействия IP, поэтому его единицу также называют дейтаграммой. Однако очень часто используется и другой термин – «пакет» (*packet*) IP.

В стеке TCP/IP принято называть «кадрами», или **фреймами** (*frame*), единицы данных любой технологии, в которые упаковываются пакеты IP для последующей переноски их через подсети составной сети. При этом не имеет значения, какое название исполь-

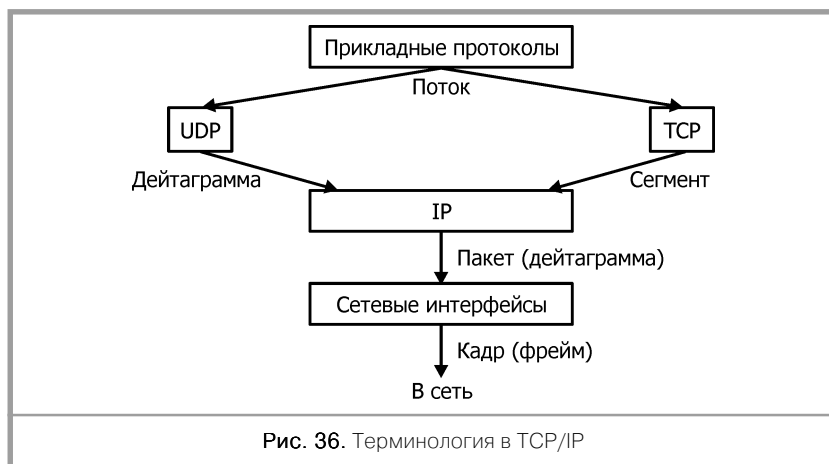


Рис. 36. Терминология в TCP/IP

зуется для этой единицы данных в локальной технологии подсети. Для TCP/IP фреймом являются и кадр Ethernet, и ячейка ATM, и пакет X.25, так как все они выступают в качестве контейнера, в котором пакет IP переносится через составную сеть.

Подводя итоги изложенному, можно сделать следующие выводы.

Наибольшее распространение при построении составных сетей в последнее время получил стек TCP/IP. Стек TCP/IP имеет 4 уровня: прикладной, транспортный, уровень межсетевого взаимодействия и уровень сетевых интерфейсов. Соответствие уровней стека TCP/IP уровням модели OSI достаточно условно.

Прикладной уровень объединяет все службы, предоставляемые системой пользовательским приложениям: традиционные сетевые службы типа telnet, FTP, TFTP, DNS, SNMP, а также сравнительно новые, такие, например, как протокол передачи гипертекстовой информации HTTP.

На традиционном уровне стека TCP/IP, называемом также основным, функционируют протоколы TCP и UDP. Протокол управления передачей TCP решает задачу обеспечения надежной информационной связи между двумя конечными узлами. Дейтаграммный протокол UDP используется как экономичное средство связи уровня межсетевого взаимодействия с прикладным уровнем.

Уровень межсетевого взаимодействия реализует концепцию коммуникации пакетов в режиме без установления соединений. Основными протоколами этого уровня являются дейтаграммный протокол IP и протоколы маршрутизации (RIP, OSPF, BGP и др.). Вспомогательную роль выполняют протокол межсетевых управляющих сообщений ICMP и протокол группового управления IGMP.

Протоколы уровня сетевых интерфейсов обеспечивают интеграцию в составную сеть других сетей. Этот уровень не регламентируется, но поддерживает все популярные стандарты физического и канального уровней: для локальных сетей – Ethernet, Token Ring, FDDI и т. д., для глобальных сетей – X.25, frame relay, PPP, ISDD и т. д. К этому уровню часто относят и протокол разрешения адресов ARP.

В стеке TCP/IP для наименования единиц передаваемых данных на разных уровнях используют следующие названия: поток, сегмент, дейтаграмма, пакет, кадр.

4.4. Адресация в IP-сетях. Типы адресов стека TCP/IP

В стеке TCP/IP используются три типа адресов:

- локальные, или аппаратные, адреса, применяемые для адресации узлов в пределах подсети;
- сетевые, или IP-адреса, используемые для однозначной идентификации узлов в пределах всей составной сети;
- доменные имена – символьные идентификаторы узлов, к которым часто обращаются пользователи.

В общем случае сетевой интерфейс может иметь одновременно один или несколько сетевых адресов, а также одно или несколько доменных имен.

В терминологии TCP/IP под определением «**локальный**» понимается действующий не во всей составной сети, а лишь в пределах подсети, то есть понятие «локальный адрес» означает адрес, который используется некоторой локальной технологией для адресации узлов в пределах подсети, являющейся элементом составной интeрcети.

Поскольку аппаратный (локальный) адрес идентифицирует узел в пределах подсети, а подсеть использует одну из базовых технологий LAN-Ethernet, FDDI, Token Ring, то для доставки данных любому узлу такой подсети достаточно указать MAC-адрес.

MAC-адрес (*Media Access Control* – управление доступом в физическую среду – стандарты IEEE) назначается сетевым адаптером и сетевым интерфейсом маршрутизаторов. Для всех существующих технологий локальных сетей MAC-адрес имеет формат 6 байт, например 11-A0-17-3D-BC-01.

В составную сеть TCP/IP могут входить подсети, построенные на основе более сложных технологий, к примеру технологии IPX/SPX. Эта сеть сама может быть разделена на подсети, и, так же как IP-сеть, она идентифицирует свои узлы аппаратными и сетевыми IPX-адресами. Но поскольку для составной сети TCP/IP составная сеть IPX/SPX является обычной подсетью, в качестве аппаратных адресов узлов

этой подсети выступают те адреса, которые однозначно определяют узлы в данной подсети, то есть адресами IPX-адреса. Аналогично если в составную сеть TCP/IP включена сеть X.25, то локальными адресами для протокола IP соответственно будут адреса X.25.

IP-адреса представляют собой основной тип адресов, на основании которых сетевой уровень передает пакеты между сетями. Эти адреса состоят из 4 байт, например 109.26.17.100. IP-адрес назначается администратором при конфигурировании компьютеров и маршрутизаторов. Он состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно либо назначен по рекомендации специального подразделения Интернета (Internet Network Information Center – InterNIC), если сеть должна работать как составная часть Интернета.

Номер узла в протоколе IP назначается независимо от локального адреса узла. Маршрутизатор по определению входит сразу в несколько сетей, поэтому каждый порт маршрутизатора имеет собственный IP-адрес. Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

Символьные имена в IP-сетях называются **доменными** и строятся по иерархическому признаку. Составляющие полного символического имени в IP-сетях разделяются точкой и перечисляются в следующем порядке: сначала простое имя хоста, затем имя группы узлов (например, имя организации), затем имя более крупной группы (поддомена) – и так до имени домена самого высокого уровня (например, домена, объединяющего организации по географическому признаку: RU – Россия, UK – Великобритания, SU – США). Примером доменного имени может служить имя base2.sales.zil.ru. Между доменным именем и IP-адресом узла нет никакой функциональной зависимости (нет алгоритмического соответствия), поэтому необходимо использовать другой способ установления соответствия в виде дополнительных таблиц или служб, чтобы узел сети однозначно определялся как по доменному имени, так и IP-адресу. В сетях TCP/IP используется распределенная служба доменных имен (Domain Name System – DNS), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называются DNS-именами.

4.5. Формы записи IP-адресов

Формы записи IP-адреса могут быть различны. IP-адрес имеет длину 4 байта (32 бита) и состоит из двух частей – номера сети и номера узла в сети. Наиболее употребляемой формой представления

IP-адреса является запись в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например: 128.10.2.30.

Этот же адрес может быть представлен в двоичной форме: 10000000.00001010.00000010.00011110.

А также в шестнадцатеричном формате: 80.0A.02.1D.

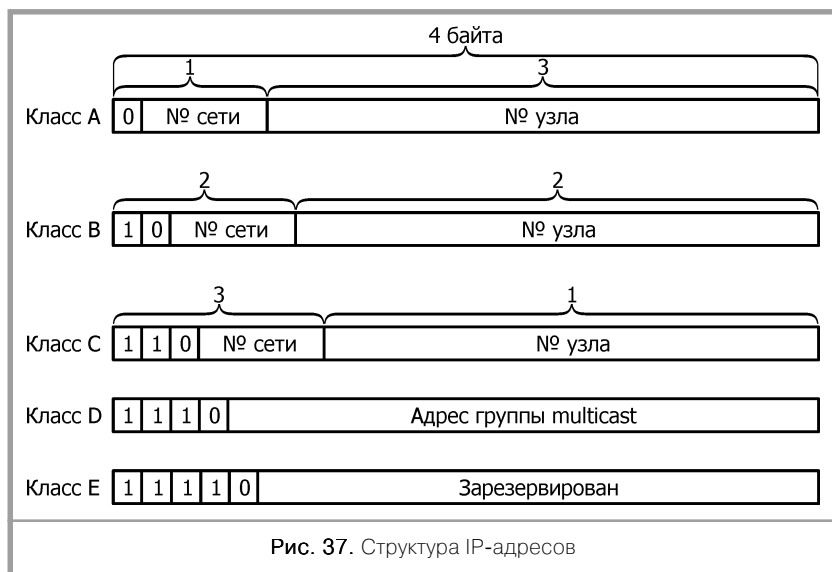
Заметим, что запись адреса не предусматривает специального разграничительного знака между номером сети и номером узла. Каким образом маршрутизаторы, на которые поступают пакеты, выделяют из адреса назначения номер сети, чтобы по нему определить дальнейший маршрут? Какая часть из 32 бит, отведенных под IP-адрес, относится к номеру сети, а какая – к номеру узла? Можно предположить несколько вариантов решения этой задачи.

Простейший из них состоит в том, что все 32-битовое поле адреса заранее делится на две части необязательно равной, но фиксированной длины, в одной из которых всегда будет размещаться номер сети, а в другой – номер узла. Но такой подход не позволяет дифференцированно подходить к потребностям отдельных предприятий и организаций и не нашел широкого применения.

Другой подход основан на использовании **«маски»**, которая позволяет максимально гибко устанавливать границу между номером сети и номером узла. В данном случае «маска» – это число, которое используется в паре с IP-адресом; двоичная запись маски содержит последовательность единиц в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Граница между последовательностью единиц и последовательностью нулей в маске соответствует границе между номером сети и номером узла в IP-адресе.

Традиционный способ решения данной проблемы заключается в использовании «классов». Этот способ представляет собой компромисс по отношению к двум вышеописанным: размеры сетей хотя и не являются произвольными, как при использовании масок, но и не являются одинаковыми, как при установлении фиксированных границ. Вводится несколько классов сетей, и для каждого класса определены свои размеры. Схема деления IP-адреса на номер сети и номер узла и основана на понятии класса, который определяется значениями нескольких первых бит адреса. На рис. 37 показана структура IP-адреса различных классов.

Если адрес начинается с 0, то этот адрес относится к классу «А», в котором под номер сети отводится один байт, а остальные три байта интерпретируются как номер узла в сети. Сети, имеющие номера в диапазоне от 1 (00000001) до 126 (01111110), называются сетями класса «А». Номер 0 не используется, а номер 127 зарезервирован для специальных целей. Сетей класса «А» немного, зато количество узлов в них может достигать 2^{24} , то есть 16 777 216 узлов.



Если первые два бита адреса равны 10, то адрес относится к классу «В». В адресах класса «В» под номер сети и номер узла отводится по два байта (по 16 бит). Сети, имеющие номера в диапазоне от 128.0 (10^{16}) до 191.255 (1011111111111111), называются сетями класса «В». Таким образом, сетей класса «В» больше, чем сетей класса «А», но размеры их меньше, максимальное количество узлов в них составляет 2^{16} (65 536).

Если адрес начинается с последовательности битов 110, то это адрес класса «С». В этом случае под номер сети отводятся 24 бита, а под номер узлов – 8 бит. Сети класса «С» наиболее распространены, но число узлов в них ограничено значением 2^8 (256) узлов.

Если адрес начинается с последовательности битов 1110, то это адрес класса «D», который обозначает особый, групповой адрес – multicast. Групповой адрес идентифицирует группу узлов (сетевых интерфейсов), которые в общем случае могут принадлежать разным сетям. Если при отправке пакета в качестве адреса назначения указан адрес класса «D», то такой пакет должен быть доставлен всем узлам, которые входят в группу.

Если адрес начинается с последовательности битов 11110, то это значит, что данный адрес относится к классу «Е». Адреса этого класса зарезервированы для будущих применений. На рис. 38 приведены диапазоны номеров сетей и максимальное число узлов, соответствующих каждому классу сетей.

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	0	1.0.0.0	126.0.0.0	2^{24}
B	10	128.0.0.0	191.255.0.0	2^{16}
C	110	192.0.0.0	223.255.255.0	2^8
D	1110	224.0.0.0	239.255.255.255	Multicast
E	11110	240.0.0.0	247.255.255.255	Зарезервирован

Рис. 38. Характеристики адресов разного класса

Большие сети получают адреса класса «А», средние – класса «В», а небольшие – класса «С».

Особые IP-адреса означают следующую интерпретацию IP-адресов:

- если весь IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет (этот режим используется только в некоторых сообщениях ICMP);
- если в поле номера сети стоят только нули, то считается, что узел назначения принадлежит той же сети, что и узел, который отправил пакет;
- если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться по всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется ограниченным широковещательным сообщением (limited broadcast);
- если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номером сети. Например, пакет с адресом 192.190.21.255 доставляется всем узлам сети 192.190.21.0. Такая рассылка называется широковещательным сообщением (broadcast).

Использование масок при IP-адресации позволяет более гибко устанавливать границу между номером сети и номером узла и отказаться от понятий классов адресов.

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

же должны представлять непрерывную последовательность. Например, если адрес 185.23.44.206 ассоциировать с маской 255.255.255.0, то номер сети будет 185.23.44.0, а не 185.23.0.0, как это определено системой классов.

В масках количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8, чтобы повторять деление адреса на байты. Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде IP-адрес 129.64.134.5 выглядит так: 10000001, 01000000, 1000110, 00000101.

А маска 255.255.128.0 – так: 11111111, 11111111, 10000000, 00000000.

Если игнорировать маску, то в соответствии с системой классов адрес 129.64.134.5 относится к классу «В», а значит, номером сети являются первые два байта – 129.64.0.0, а номером узла – 0.0.134.5.

Если же использовать для определения границы номера сети маску, то 17 последовательных двоичных единиц в маске 255.255.128.0, «наложенные» на IP-адрес, делят его на следующие две части.

	№ сети	№ узла
IP-адрес 129.64.134.5	10000001.01000000.1	0000110.00000101
Маска 255.255.128.0	11111111.11111111.1	0000000.00000000

В десятичной форме записи номер сети – 129.64.128.0, а номер узла – 0.0.6.5.

Для стандартных классов сетей маски имеют следующие значения:

- 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).

Механизм масок широко распространен в IP-маршрутизации, причем маски могут использоваться для самых разных целей. С их помощью администратор может разбивать свою сеть на несколько других, не требуя от поставщика услуг дополнительных номеров сетей (операция subnetting). На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения так называемых «префиксов» с целью уменьшения объема таблиц маршрутизации и повышения за счет этого производительности маршрутизаторов – такая операция называется subnetting.

Поскольку IP-адреса уникально идентифицируют узел в пределах составной сети, они назначаются централизованно. Если сеть

небольшая и автономная, то уникальность IP-адресов в пределах этой сети может быть обеспечена администратором сети. При этом он может выбирать для нумерации сетей и узлов любые синтаксически правильные IP-адреса. Если сеть очень велика, как, например, Интернет, то процесс назначения IP-адресов становится сложным и разбивается на два этапа. Первый – распределение номеров сетей – регулируется специальным административным органом (в Интернете это ICAN), обеспечивающим однозначность нумерации сетей. После того как сеть получила номер, наступает второй этап – назначение номеров узлам сети. Назначение IP-адресов узлами сети может происходить вручную – администратор сети сам ведет списки свободных и занятых адресов и конфигурирует сетевой интерфейс, либо автоматически – с использованием протокола DHCP через сервер, который автоматически выделяет адреса узлам в ответ на поступающие запросы.

Одной из главных задач, которая ставилась при создании протокола IP, являлось обеспечение совместной согласованной работы в сети, состоящей из подсетей, в общем случае использующих разные сетевые технологии. Очевидно, что для того, чтобы частная технология подсети смогла доставить пакет на следующий маршрутизатор, необходимо:

- во-первых, упаковать пакет в кадр соответствующего для данной подсети формата (например, Ethernet);
- во-вторых, снабдить кадр адресом, формат которого был бы понятен локальной технологии подсети (преобразовать, например, IP-адрес в MAC-адрес (управление доступом в физическую среду, стандарт IEEE, Media Access Control)). Для определения локального адреса по IP-адресу используется протокол разрешения адресов (Address Resolution Protocol – ARP). Он позволяет определять адреса сетевых адаптеров узлов, расположенных в одной физической сети.

Протокол разрешения адресов реализуется различным образом в зависимости от того, какой протокол канального уровня работает в данной сети – протокол локальной сети (Ethernet, Token Ring, FDDI) с возможностью широковещательного доступа одновременно ко всем узлам сети или же какой-либо из протоколов глобальной сети (X.25, frame relay), которые, как правило, не поддерживают широковещательного доступа.

Рассмотрим работу протокола ARP в сетях с широковещанием. Протокол ARP всегда сначала ищет адреса IP и сетевого адаптера в кэш-памяти (памяти блокнотного типа) перед формированием широковещательного ARP-запроса.

4.6. Разрешение локального IP-адреса

Перед соединением двух узлов IP-адрес каждого из них должен быть преобразован в адрес сетевого адаптера. Этот процесс состоит из выполнения ARP-запроса и получения ARP-ответа.

1. ARP-запрос формируется каждый раз при попытке одного узла связаться с другим. Если протокол IP определяет, что IP-адрес принадлежит локальной сети, узел-отправитель ищет адрес узла-получателя в своем собственном ARP-кэше.
2. Если он не найден, протокол ARP формирует запрос типа «Чей это IP-адрес и каков Ваш адрес сетевого адаптера?», в который также включаются адреса IP и сетевого адаптера узла-отправителя. ARP-запрос посылается в широковещательном режиме, чтобы все узлы в локальной сети могли принять и обработать его.
3. Каждый узел в локальной сети получает этот широковещательный запрос и сравнивает указанный в нем IP-адрес со своим собственным. Если они не совпадают, запрос игнорируется.
4. Узел-получатель определяет, что IP-адрес в запросе совпадает с его собственным, и посылает на узел-отправитель ARP-ответ, в котором указывает свой адрес сетевого адаптера. Затем он обновляет свой ARP-кэш, занося в него соответствие IP-адреса узла-отправителя адресу его сетевого адаптера. После того как узел-отправитель получает ARP-ответ, соединение может быть установлено.

Разрешение удаленного IP-адреса. Протокол ARP также позволяет связываться двум узлам из различных сетей. В этом случае широковещательный ARP-запрос обеспечивает возможность выяснить адрес используемого отправителем шлюза по умолчанию, а не узла-получателя. Если получатель находится в удаленной сети, то широковещательный ARP-запрос используется для поиска маршрутизатора, который может пересылать пакеты в эту сеть.

1. При соединении определяется, что IP-адрес узла-получателя принадлежит удаленной сети. Узел-отправитель ищет в локальной таблице маршрутизации путь к узлу-получателю или его сети. Если путь не найден в таблице, узел-отправитель определяет IP-адрес шлюза по умолчанию. Затем он ищет в кэше протокола ARP соответствующий ему адрес сетевого адаптера.
2. Если этот адрес в кэше отсутствует, то широковещательный ARP-запрос используется для получения адреса шлюза, а не узла-получателя. Маршрутизатор (шлюз) в ответ на ARP-запрос узла-отправителя посылает адрес своего сетевого адапте-

- ра. Затем узел-отправитель адресует пакет на маршрутизатор для доставки его в сеть получателя и далее – узлу-получателю.
3. На маршрутизаторе выясняется, является ли IP-адрес получателя локальным или удаленным. Если он локальный, то маршрутизатор использует протокол ARP (кэш или широковещание) для получения его адреса сетевого адаптера. Если же удаленный, маршрутизатор ищет в своей таблице маршрутизации необходимый шлюз, а затем использует протокол ARP (кэш или широковещание) для получения адреса его сетевого адаптера. Далее пакет отправляется непосредственно следующему получателю в этой цепочке.
 4. Когда пакет достигнет получателя и будет обработан им, все исходящие пакеты таким же образом будут доставлены обратно. Например, при выполнении команды P!pd в ответ на эхо-запрос узел-получатель пакета формирует эхо-ответ протокола ICMP (ICMP echo-reply). Поскольку узел-отправитель находится в удаленной сети, в локальной таблице маршрутизации ищется адрес шлюза к сети узла-отправителя. Если поиск завершается успехом, адрес сетевого адаптера шлюза выясняется с помощью ARP.
 5. Если адреса сетевого адаптера указанного шлюза нет в кэше протокола ARP, то для его определения используется широковещательный ARP-запрос. Как только адрес получен, эхо-ответ протокола ICMP посылается на маршрутизатор, который перенаправляет его на исходный узел-отправитель.

Совсем другой способ разрешения адресов используется в глобальных сетях, в которых не поддерживаются широковещательные сообщения. Здесь администратору сети чаще всего приходится вручную формировать и помещать на какой-либо сервер ARP-таблицы, в которых он задает, например, соответствие IP-адресов адресам X.25, которые имеют для протокола IP смысл локальных адресов.

В то же время сегодня наметилась тенденция автоматизации протокола ARP и в глобальных сетях. Для этой цели среди всех маршрутизаторов, подключенных к какой-либо глобальной сети, выделяется специальный маршрутизатор, который ведет ARP-таблицу для всех остальных узлов и маршрутизаторов этой сети.

При таком централизованном подходе для всех узлов и маршрутизаторов вручную нужно задать только IP-адрес и локальный адрес выделенного маршрутизатора. Затем каждый узел и маршрутизатор регистрирует свои адреса в выделенном маршрутизаторе, а при необходимости установления соответствия между IP-адресом и локальным адресом протокольный модуль обращается к выделен-

ному с запросом и автоматически получает ответ без участия администратора. Работающий таким образом маршрутизатор называется ARP-сервером.

В некоторых случаях возникает обратная задача – нахождение IP-адреса по известному локальному адресу. Тогда в действие вступает «реверсивный протокол ARP» (Reverse Address Resolution Protocol – RARP). Этот протокол используется, например, при старте бездисковых станций, не знающих в начальный период своего IP-адреса, но знающих MAC-адрес своего сетевого адаптера.

4.7. Служба доменных имен

Широковещательный способ установления соответствия между символьными именами и локальными адресами хорошо работает только в небольшой локальной сети, не разделенной на подсети. В стеке TCP/IP применяется доменная (выделенное множество объектов) система имен, которая имеет иерархическую древовидную структуру, допускающую использование в имени произвольного качества составных частей (рис. 39).

Совокупность имен, у которых несколько старших составных частей совпадает, образует домен имен.

Соответствие между доменными именами и IP-адресами может устанавливаться как средствами локального хоста, так и средствами централизованной службы. По мере роста Интернета файлы hosts также росли, и создание масштабируемого разрешения имен нашло свое решение в создании социальной службы – системы доменных имен (Domain Name System – DNS). DNS – это централизованная служба, основанная на распределенной базе отображений «доменное имя – IP-адрес». Служба DNS использует в своей работе протокол типа «клиент-сервер». В нем определены DNS-серверы и DNS-клиенты. DNS-серверы поддерживают распределенную базу отображений, а DNS-клиенты обращаются к серверам с запросами о разрешении доменного имени в IP-адрес.

В доменной системе адресации DNS каждый корреспондент получает сетевой адрес, включающий две составляющие: идентификатор пользователя (userid) и идентификатор узла (nodeid). Идентификатор «userid» является уникальным для узла сети. Идентификатор «nodeid» представляет собой текстовую строку, состоящую из доменов, разделенных точками.

В системе DNS ключевым является понятие «полностью определенное имя домена» – это имя домена, которое включает все домены более высокого уровня и образует полное, целое имя. Структуру DNS (Domain Name System – служба доменных имен) можно представить в виде дерева, каждый узел которого имеет свое название (метку) (рис. 39).

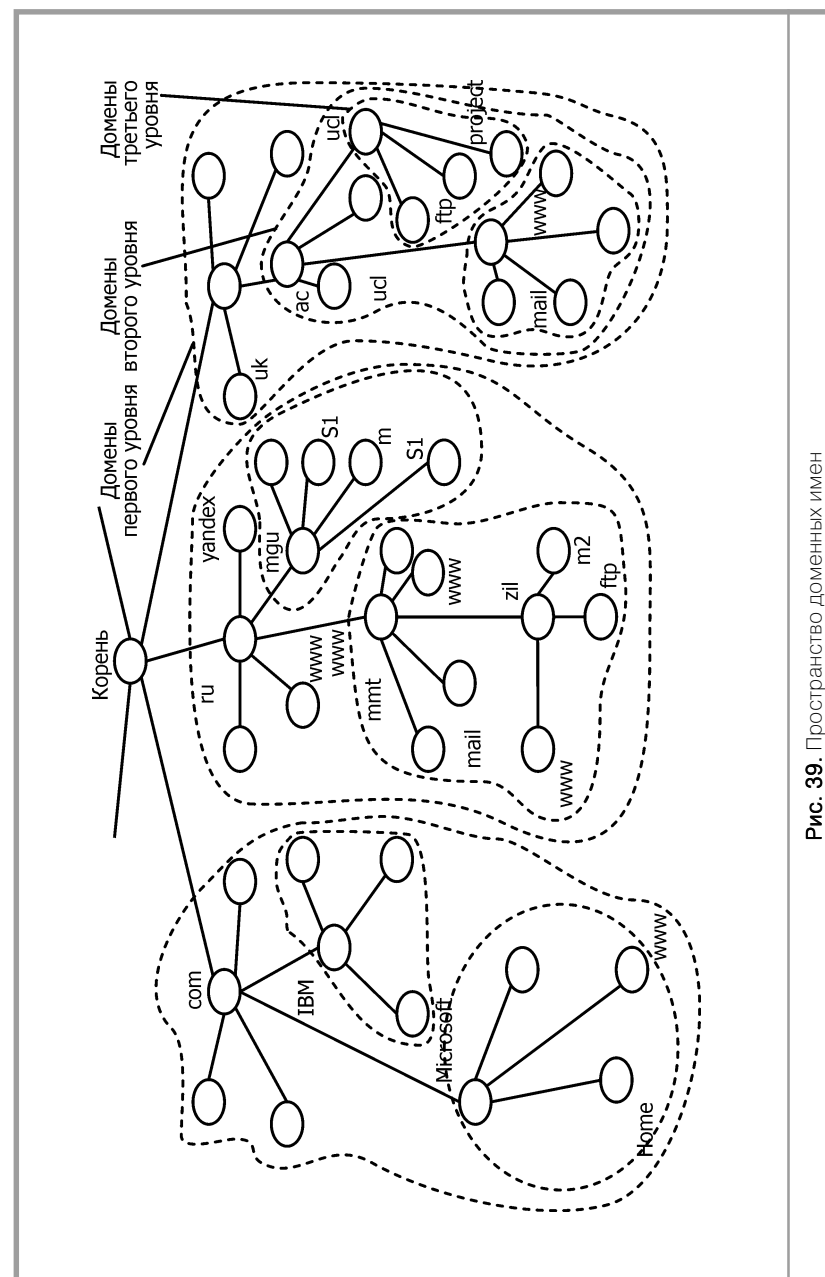


Рис. 39. Пространство доменных имен

Для каждого конкретного узла «полностью определенное имя домена» будет состоять из его имени и имени всех узлов, связывающих его с корневым деревом, причем корневой домен всегда нулевой.

Существуют две основные схемы разрешения DNS-имен. В первом варианте работу по поиску IP-адреса координирует DNS-клиент.

1. DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени.
2. DNS-сервер отвечает, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в старшей части запрошенного имени.
3. DNS-клиент делает запрос следующего DNS-сервера, который отправляет его к DNS-серверу нужного поддомена, и т. д., пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.

Такая схема взаимодействия называется нерекурсивной, когда клиент сам итеративно выполняет последовательность запросов к разным серверам имен. Но эта схема загружает клиента достаточно сложной работой, и применяется она редко.

Во втором варианте реализуется рекурсивная процедура.

1. DNS-клиент запрашивает локальный DNS-сервер, то есть тот сервер, обслуживающий поддомен, к которому принадлежит имя клиента.
2. Если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту; это может соответствовать случаю, когда запрошенное имя входит в тот же поддомен, что и имя клиента, а также случаю, когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше.
3. Если локальный сервер не знает ответа, то он выполняет итеративные запросы к корневому серверу и т. д. точно так же, как это делает клиент в первом варианте. Получив ответ, он передает его клиенту, который все это время просто ожидает его от своего локального DNS-сервера.

Итерация – процесс повторения последовательности действий.

В этой схеме клиент перепоручает работу своему серверу, поэтому схема называется косвенной, или рекурсивной. Практически все DNS-клиенты используют рекурсивную процедуру.

4.8. Протокол межсетевое взаимодействие IP

Протокол IP не ориентирован на соединение, поскольку он не устанавливает сеанс связи, перед тем как начать обмен данными. Протокол IP ненадежный – он не гарантирует доставку, хотя делает все

возможное для доставки пакета. По пути пакет может быть потерян, доставлен в неправильной последовательности, продублирован или задержан.

Протокол IP не требует подтверждения при приеме данных. Отправитель или получатель не информируется при потере пакета или доставке его в неправильной последовательности. Ответственность за подтверждение получения пакетов несут высокоуровневые транспортные протоколы, например TCP.

Поля IP-датаграммы в приведенной ниже таблице добавляются в заголовок пакета при его получении с транспортного уровня.

Поле	Описание
Source IP-address (IP-адрес отправителя)	Идентифицирует отправителя датаграммы при помощи IP-адреса
Destination IP-address (IP-адрес получателя)	Идентифицирует получателя датаграммы при помощи IP-адреса
Protocol (Протокол)	Информирует протокол IP узла-получателя о том, какому протоколу верхнего уровня – TCP и UDP – его следует передать
Checksum (Контрольная сумма)	Легко вычисляемое значение для проверки целостности пришедшего пакета
Time to live, или TTL (время существования)	Определяет, сколько секунд находится датаграмма в сети, перед тем как она будет отвергнута (discarded). Предотвращает бесконечное блуждание пакета по сети. Маршрутизаторы должны уменьшать TTL на количество секунд, проведенных датаграммой в маршрутизаторе. TTL уменьшается по меньшей мере на одну секунду каждый раз, когда датаграмма проходит через маршрутизатор

Если протокол IP идентифицирует адрес получателя как локальный, то он передаст пакет на этот узел напрямую. Если же IP-адрес получателя идентифицирован как удаленный, то IP начнет искать маршрут к удаленному узлу в локальной таблице маршрутизации. Если он не найдет подходящего маршрута, то отправит пакет на шлюз по умолчанию, заданный в конфигурации узла-отправителя.

Реализация IP на маршрутизаторе. Маршрутизатор обрабатывает полученные им IP-пакеты следующим образом:

1. Уменьшает значение ТТТ на 1 с или больше, если пакет надолго задерживается на маршрутизаторе. Если значение ТТТ достигает нуля, пакет отвергается.
2. Пакет может быть фрагментирован, если его размер слишком велики для сети дальнейшего следования.
3. Если пакет фрагментирован, то IP создает для каждого нового пакета (фрагмента) отдельный заголовок, устанавливая:
 - Flag (Флаг), указывающий, что существуют и другие фрагменты, которые будут отправлены вслед;
 - Fragment ID (Идентификатор фрагмента), идентифицирующий все фрагменты, составляющие один пакет;
 - Fragment Offset (Смещение фрагмента), обеспечивающее правильную сборку пакета на узле-получателе.
4. Вычисляет новую контрольную сумму.
5. Определяет адрес сетевого адаптера следующего маршрутизатора.
6. Направляет пакет дальше в сеть.

4.9. Протокол доставки сообщений ТСП

Протокол ТСП предоставляет надежную, ориентированную на соединение службу доставки.

Данные протокола ТСП передаются сегментами, и соединение должно быть установлено до того, как узлы начнут обмениваться данными. ТСП использует потоки, в которых данные представлены в виде последовательности байт.

ТСП обеспечивает надежность, присваивая номера последовательности (sequence number) каждому передаваемому сегменту. Если сегмент разбивается на мелкие пакеты, то узел-получатель сможет узнать, все ли части получены. Для этого используются подтверждения. Для каждого отправленного сегмента узел-получатель должен вернуть отправителю подтверждение (acknowledgement, АСК) в течение определенного времени (рис. 40).

Если отправитель не получил АСК, то данные передаются повторно. Если сегмент поврежден, то узел-получатель отвергает его. Поскольку АСК в этом случае не посылается, отправитель передает сегмент еще раз.

Порты. Приложения, использующие сокет, идентифицируют себя на компьютере посредством номера порта протокола (protocol port number). Например, FTP-сервер использует определенный ТСП-порт, поэтому другие приложения могут связаться с ним.

Порты могут иметь любой номер от 0 до 65 536. Номера портов для приложений клиентов динамически назначаются операционной системой при обработке запроса и обслуживания.

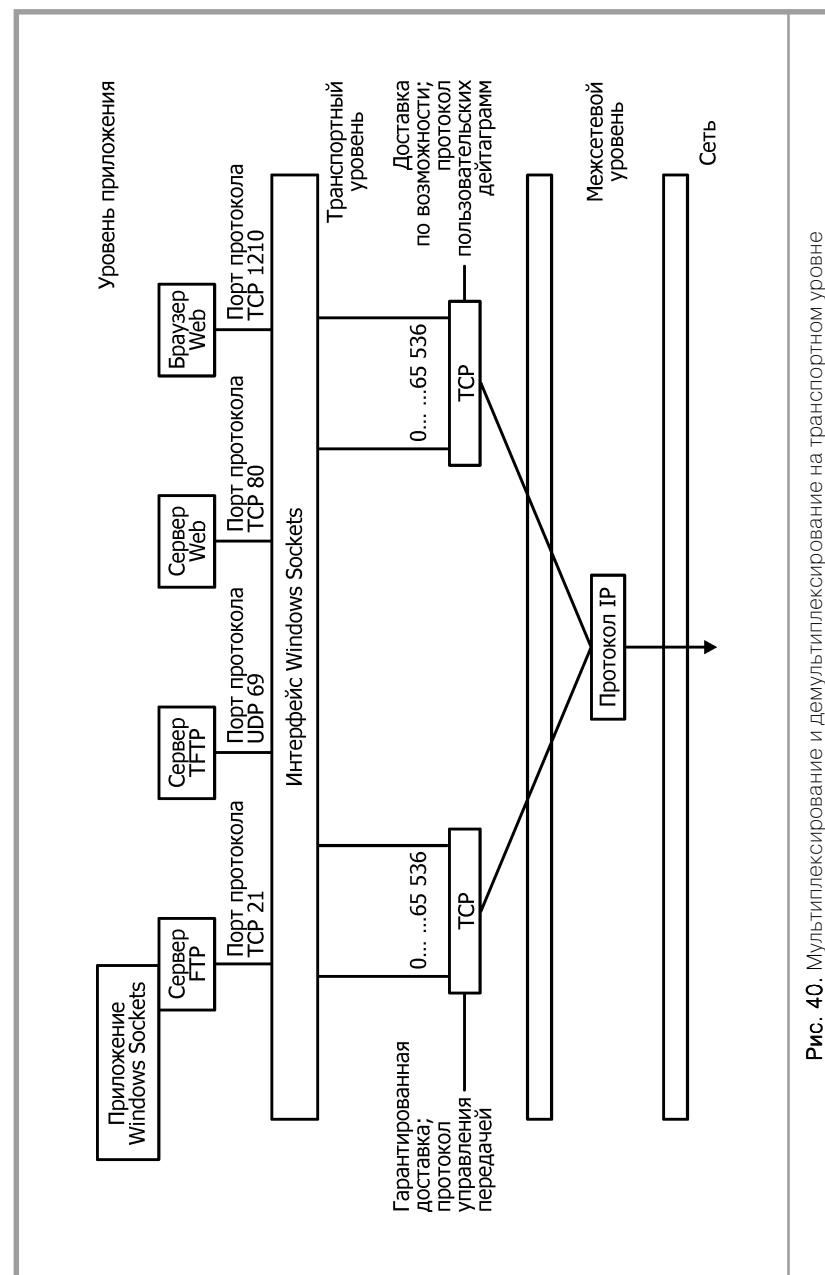


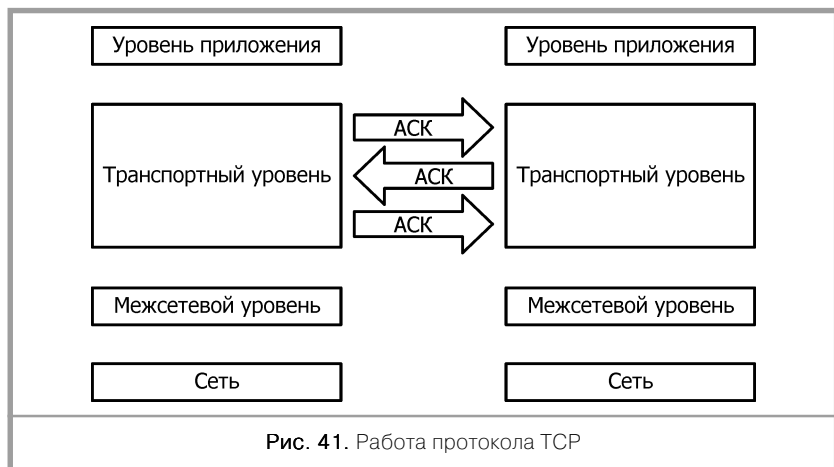
Рис. 40. Мультиплексирование и демultipлексирование на транспортном уровне

Сокеты. *Socket (socket)* обеспечивает конечную точку сетевого соединения. Приложение, создавая сокет, указывает три параметра: IP-адрес узла, тип обслуживания (протокол TCP для ориентированного на соединение обслуживания и UDP для неориентированного) и порт, используемый приложением.

Порты протокола TCP. Порт протокола TCP указывает место доставки сообщений. Номера портов, меньшие 256, определены как широко используемые.

Номер порта	Описание
21	FTP
23	Telnet
53	Доменная система имен (DNS)
139	Сервер NetBIOS

Установка связи по протоколу TCP. Инициализация TCP-соединения происходит в три этапа. Делается это для синхронизации отправки и получения сегментов, извещения другого узла о количестве данных, которые можно послать за один раз, и установки виртуального соединения. Вот из каких операций состоит этот процесс (рис. 41).



1. Узел-отправитель запрашивает соединение, посылая сегмент с установленным флагом синхронизации (SYN).
2. Узел-адресат подтверждает получение запроса, отправляя обратно сегмент с:
 - установленным флагом синхронизации;
 - порядковым номером начального байта сегмента, который может послать, или номером последовательности (sequence number);
 - подтверждением, включающим порядковый номер следующего сегмента, который он ожидает получить.
3. Запрашивающий узел посылает обратно сегмент с подтверждением номера последовательности и номером своего подтверждения (acknowledgement number).

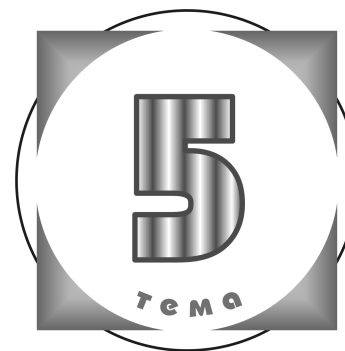
Для завершения соединения TCP действует аналогично. Это гарантирует, что оба узла закончат передачу и примут все данные.

Скользящие окна протокола TCP. Протокол TCP буферизует данные для передачи между двумя узлами, используя скользящие окна (sliding windows). Каждый TCP/IP-узел поддерживает два скользящих окна: одно – для приема данных, а другое – для отправки. Размер окна определяет объем данных, которые могут быть буферизованы на компьютере.

Структура TCP-пакета. Все пакеты протокола TCP имеют две части – данные и заголовок. В таблице приведены поля заголовка TCP-пакета.

Поле	Описание
Source Port (Порт отправителя)	TCP-порт узла-отправителя
Destination Port (Порт получателя)	TCP-порт узла-получателя. Определяет конечную точку соединения
Sequence Number (Порядковый номер)	Номер последовательности пакета. Используется для проверки получения всех байт соединения
Acknowledgement Number (Номер подтверждения)	Порядковый номер байта, который локальный узел планирует получить следующим
Data Length (Длина данных)	Длина TCP-пакета
Reserved (Зарезервировано)	Зарезервировано для будущего использования

Поле	Описание
Flags (Флаги) Windows (Окно)	Это поле описывает содержимое сегмента. Показывает, сколько места доступно в настоящий момент в окне протокола TCP.
Checksum (Контрольная сумма)	Проверяет, поврежден ли заголовок.
Urgent Pointer (Указатель срочности)	Когда отправляются срочные данные (указано в поле Flags), в этом поле задается конечная граница области срочных данных в пакете.



Организация и администрирование ЛВС

5.1. Задачи и методы администрирования ЛВС

Управление ЛВС (local – area network management) – модель администрирования с целью обеспечения эффективной работы локальной сети.

Международная организация по стандартизации (ISO) согласно рекомендациям ITU-TX.700 и стандарту ISO-7498-4 определила следующие пять категорий управления, которые должна включать система управления ЛВС:

1. Управление конфигурацией сети и именованием. В рамках этой категории проводятся установление и управление параметрами, определяющими состояние ЛВС.
2. Обработка отказов, сбоев, ошибок в работе сети. Здесь осуществляются обнаружение, изоляция и исправление указанных неполадок в сети.
3. Учет ресурсов и работы сети. Основные функции управления учетом – запись и выдача информации об использовании ресурсов ЛВС.
4. Анализ и контроль производительности и надежности компонентов сети. Здесь производится анализ и управление скоростью, с которой сеть обрабатывает данные.
5. Обеспечение безопасности данных. Основные функции – контроль доступа к ресурсам ЛВС и защита информации от несанкционированного доступа.

1. Задача управления *конфигурацией сети и именованием* заключается в конфигурировании параметров как элементов сети, так и сети в целом.

Для элементов сети, таких как маршрутизаторы, мультиплексоры и т. п., с помощью этой группы задач определяются сетевые адреса, идентификаторы (имена), географическое положение и прочее.

Для сети в целом управление конфигурацией обычно начинается с построения карты сети, то есть отображения реальных связей между элементами сети – образование новых физических или логических каналов, изменение таблиц коммутации и маршрутизации.

Управление конфигурацией, как и другие задачи управления, могут выполняться в автоматическом, ручном или пользовательском режимах. Карта сети может составляться автоматически, на основании зондирования реальной сети пакетами-исследователями, а может быть построена оператором системы управления вручную. Чаще всего применяются автоматические методы, когда автоматически полученную карту оператор подправляет вручную. Методы автоматического построения топологической карты, как правило, являются фирменными разработками.

Более сложной задачей является настройка коммутаторов и маршрутизаторов на поддержку маршрутов и виртуальных путей между пользователями сети. Согласованная ручная настройка таблиц маршрутизации при полном или частичном отказе от использования протокола маршрутизации (а в некоторых глобальных сетях, например X.25, такого протокола просто не существует) представляет собой сложную задачу. Многие системы управления сетью общего назначения ее выполняют, но существуют специализированные системы конкретных производителей, например система NetSgs компании Cisco Systems, которые решают ее для маршрутизаторов этой же компании.

2. Группа задач *обработки ошибок* включает выявление, определение и устранение последствий *сбоев* и *отказов* в работе сети. На этом уровне выполняется не только регистрация сообщений об ошибках, но и их фильтрация, маршрутизация и анализ на основе некоторой корреляционной модели. Фильтрация позволяет выделить из весьма интенсивного потока сообщений об ошибках, который обычно наблюдается в большой сети, только важные сообщения, маршрутизация обеспечивает их доставку нужному элементу системы управления, а корреляционный анализ дает возможность найти причину, породившую поток взаимосвязанных сообщений (например, обрыв кабеля может быть причиной большого количества сообщений о недоступности сетей и серверов).

Устранение ошибок может быть как автоматическим, так и полуавтоматическим.

В первом случае система непосредственно управляет оборудованием или программным комплексом и обходит отказавший элемент

за счет резервных каналов и т. п. В полуавтоматическом режиме основные решения и действия по устранению неисправности выполняются людьми, а система управления только помогает в организации этого процесса – оформляет квитанции на выполнение работ и отслеживает их поэтапное выполнение (подобно системам групповой работы). В этой группе задач иногда выделяют подгруппу задач управления проблемами, подразумевая под проблемой сложную ситуацию, требующую для решения обязательного привлечения специалистов по обслуживанию сети.

3. К задачам *учета работы и ресурсов сети* относится регистрация времени использования различных ресурсов сети – устройств, каналов и транспортных служб за определенный отрезок времени. Подобные задачи имеют дело с такими понятиями, как время использования службы и плата за ресурсы – *billing*. Ввиду специфического характера оплаты услуг у различных поставщиков и различными фирмами соглашения об уровне услуг эта группа функций обычно не включается в коммерческие системы и платформы управления типа HP Open View, а реализуется в заказных системах, разрабатываемых для конкретного заказчика.

4. Задачи *анализа производительности и надежности* связаны с оценкой на основе накопленной синтетической информации таких параметров, как время реакции системы, пропускная способность реального или виртуального канала связи между двумя конечными абонентами сети, интенсивность трафика в отдельных каналах и сегментах сети, вероятность искажения данных при их передаче через сеть, а также коэффициент готовности сети или ее определенной транспортной службы. Функции анализа производительности и надежности сети нужны как для оперативного управления сетью, так и для планирования развития сети. Результаты анализа производительности и надежности позволяют контролировать соглашение об уровне обслуживания (*Service Level Agreement – SLA*), заключенное между пользователем сети и ее администраторами (или компанией, продающей услуги). Обычно в *SLA* оговариваются такие параметры надежности, как коэффициент готовности службы в течение года или месяца, максимальное время устранения отказа, а также параметры производительности, например средняя и максимальная пропускная способность при соединении двух точек подключения пользовательского оборудования, время реакции сети (если информационная служба, для которой определяется время реакции, поддерживается внутри сети), максимальная задержка пакетов при передаче через сеть (если сеть используется только как транзитный транспорт). Без средств анализа производительности и надежности поставщик услуг публичной сети или отдел информационных технологий предприятия не сможет ни проконтролировать

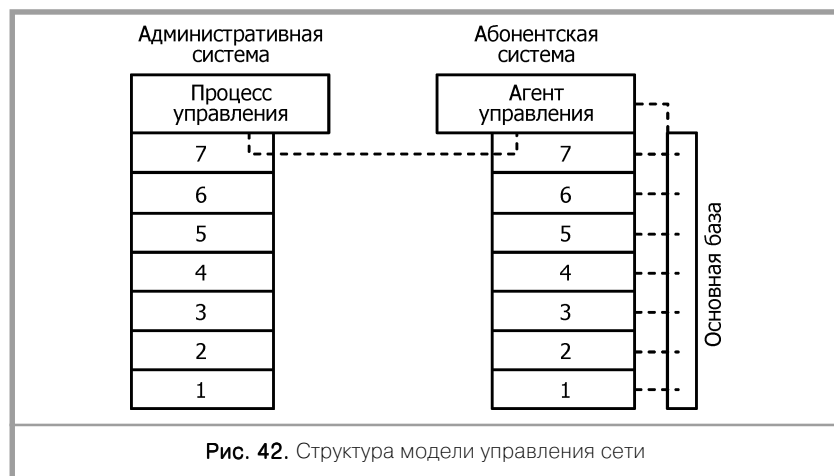
вать, ни тем более обеспечить нужный уровень обслуживания для конечных пользователей сети.

5. Задачи управления *безопасностью* (Security Management) подразумевают контроль доступа к ресурсам сети (данным и оборудованию) и сохранение целостности данных при их хранении и передаче через сеть. Базовыми элементами управления безопасностью являются процедуры аутентификации пользователей, назначение и проверка прав доступа к ресурсам сети, распределение и поддержка ключей шифрования, управления полномочиями и т. п. Части функции этой группы не включаются в системы управления сетями, а реализуются в виде специальных продуктов (например, систем аутентификации и авторизации Kerberos, различных защитных экранов, систем шифрования данных), либо входят в состав операционных систем и системных приложений.

5.2. Основные цели управления ЛВС

Основные цели управления ЛВС заключаются в следующем (рис. 42):

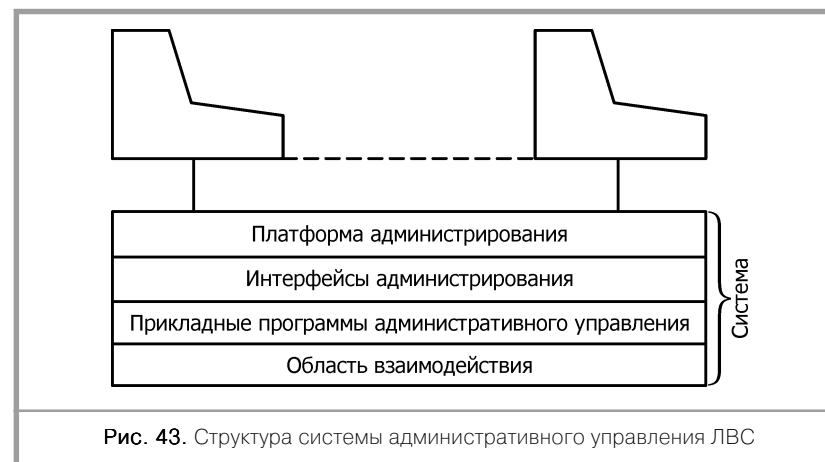
- уменьшить число сетевых неполадок за счет правильной организации процесса функционирования сети;
- изолировать возникающие неполадки в работе сети и уменьшить сопутствующие им потери.



Все более широкое признание получают так называемые открытые «платформы», обеспечивающие управление устройствами и программами различных производителей. Они используют стан-

дартный для всех систем протокол сетевого администрирования, например простой протокол управления сетью (SNMP – Simple Network Management Protocol), опрашивая программы-агенты, разрешенные на всех объектах сети.

Схема разграничения функций между платформой администрирования и прикладными программами показана на рис. 43. Платформа взаимодействует с терминалами, обеспечивающими работу персонала административного управления. Она содержит модули опроса, фильтрации, событий, учета хранения данных. С помощью интерфейса прикладной программы (API – Application Program Interface) платформа связывается с прикладными программами, собирающими в сети необходимые сведения и дающими команды программам-агентам. Прикладные программы связаны с областью взаимодействия.



При управлении ЛВС особо выделяются три основных типа компонентов сети: серверы, клиенты и средства связи. Управление локальной сетью осуществляется на сервере либо специально для этого используемом клиентом компьютере. Он именуется административной консолью.

Важно, чтобы штат администраторов, занимающихся управлением, был минимальным. Обычно считается, что на 50 систем достаточно иметь одного сотрудника.

В модели ISO каждая система (рис. 44) имеет так называемую основную базу управляющей информации, содержащую на всех семи уровнях области взаимодействия открытых систем (OSI) объекты.



Эти объекты собирают сведения о работе уровней в системе. Рассматриваемая база подчинена действиям агента управления, который является прикладным процессом абонентской системы. За работой всех агентов управления в соответствии с протоколом управления следит процесс управления, находящийся в административной системе (NMS – Network Management System – система управления сетью).

Управление сетью обеспечивается сетевой службой NMS. Разработаны и фирменные стандарты управления сетью – Простой протокол управления сетью (SNMP).

В централизованной структуре все функции управления сетью поручаются одной административной системе. Вместе с тем в последнее время приобретает большую популярность распределенная структура. В ней функции управления сетью распределяются между группой систем. Эта структура дороже, но более надежна. В выполнении функций управления помогают анализаторы сетей.

Целью управления данными является освобождение пользователей от большинства операций по манипулированию файлами и обращению к наборам данных. Благодаря этому пользователи могут сосредоточиться на логических характеристиках данных.

В архитектуре терминал – главный компьютер управления данными (УД), функции манипуляции данными (Ф) и образы предос-

тавления данных (ПД) вначале размещались в главном компьютере. С компьютером через коммуникационную сеть взаимодействовали терминалы (Т), представляющие простой интерфейс пользователя. Затем начался процесс децентрализации управления данными.

5.3. Основные принципы управления ЛВС

Представление данных стало распределенным, а затем удаленным. После этого и функции манипуляции стали распределенными, а затем – удаленными. И наконец, начался этап внедрения распределенного управления данными. По мере развития процесса децентрализации возрастают возможности и скорость обработки данных. Абонент из простого терминала превращается в многоцелевого клиента. Возникает новая технология, определяемая архитектурой клиент-сервер.

Основные принципы управления ЛВС определяют главные решения по реализации функций в рамках указанных выше категорий управления.

К ним относятся следующие.

1. Управление сетью осуществляется с использованием ее плана, который изменяется вместе с изменениями, происходящими в сети. В плане сети должна содержаться информация о ее топологии, кабельных трассах и схемах соединения кабелей, протяженности сети, стандарте протоколов и оборудования, сетевых топологиях, росте числа рабочих станций, появлении новых средств и инструментов для управления сетью.
2. Для управления современной ЛВС (особенно большой ЛВС) необходима автоматизированная система управления (АСУ ЛВС), которая должна учитывать многие технические аспекты по сбоям и неполадкам в сети.

Система управления ЛВС должна:

- обеспечивать возможность проведения перекрестного контроля для надежного обнаружения сбоев и отказов, особенно в тех случаях, когда отказы в одном из компонентов сети могут воздействовать на другие компоненты;
- обнаруживать и сообщать о таких аппаратных или программных сбоях, которые могут привести ЛВС в состояние полной остановки или в режим резкого увеличения трафика сети, на который она не рассчитывает (например, сетевые адаптеры, обнаружив ошибку, переходят в режим передачи сообщений об этом событии, что и увеличивает трафик сети);
- обладать устойчивостью в работе, адекватной реакции на ошибочные или лишние сообщения о работоспособности

сети. Система управления должна правильно реагировать на дублированные сообщения или сообщения от незарегистрированных или отключенных рабочих станций. Она должна продолжать работу, игнорируя такие сообщения, или уведомлять оператора об ошибках и посылать сигнал сброса на дефектный узел сети;

- иметь средства для периодического тестирования сети, включать встроенные средства для испытания сетевых интерфейсов, средства проведения учета и проверки систем ЛВС и средства для протоколирования активности компонентов сети;
- обладать способностью адаптироваться при развитии ЛВС, связанном с добавлением новых узлов, введением новых технологий, присоединением к другим сетям.

3. Система управления ЛВС в дополнение к техническим аспектам должна решать вопросы и административного характера, а именно:

- иметь возможность контроля и управления процессом распределения программ в сети с целью предотвращения использования нелегального программного обеспечения и борьбы с компьютерными вирусами. Одни из возможных и широко применяемых способов такого контроля – распространение всего программного обеспечения через некоторый центр (вначале программы копируются на файловый центр из единого центра распределения, а затем переносятся на локальные накопители рабочих станций);
- передавать отчеты о работе сети и контроль ее характеристик на рабочую станцию администратора ЛВС (в малых ЛВС, где работа администратора выполняется одним из пользователей) или на центральную машину (в больших ЛВС, где имеется штатный администратор) для их дальнейшего анализа и обзора;
- предоставлять средства для контроля активности файлового сервера, серверов печати, межсетевых шлюзов и иметь возможность для оперативной индикации сбоев и неполадок в этих узлах на дисплее администратора;
- вести учет событий, таких как время суток, когда в сети имеют место пиковые нагрузки, появление новых адресов, ошибочных ситуаций. Результаты учета таких событий используются администратором для накопления статистики и последующего анализа;
- предоставлять администратору сети информацию о статусе устройств, присоединенных ЛВС, таких как рабочие

станции, мосты, межсетевые шлюзы, а также информацию о тестировании состояния трассы ЛВС между рабочими станциями;

- иметь возможность управления конфигурацией ЛВС. Для этого требуются знания о том, какое программное обеспечение установлено на каждой рабочей станции сети и как эта станция сконфигурирована. При централизованном разделении программного обеспечения в сети не возникает проблем с получением необходимой информации. Наилучшим решением (особенно в ЛВС с сотнями и тысячами рабочих станций) является такое, когда имеется стандартная, заранее оговоренная конфигурация для каждой из используемых программ.
4. Управление ЛВС должно включать функции контроля доступа к ресурсам сети и защиты данных. В малых ЛВС эти функции выполняет сетевая операционная система (СОС), в больших сетях они выполняются средствами управления ЛВС. Программное обеспечение системы управления сетью поддерживает функции администратора как руководителя службы контроля и даже может регулировать доступ к прикладным программам.

5.4. Многоуровневое представление задач управления

Кроме описанного выше разделения задач управления на несколько функциональных групп, полезно разделять задачи управления на уровни в соответствии с иерархической организацией корпоративной сети. Корпоративные сети строятся иерархически, отражая иерархический характер самого предприятия и его задач. Нижний уровень сети составляют элементы сети – отдельные компьютеры, коммуникационные устройства, каналы передачи данных. На следующем уровне иерархии эти элементы образуют сети разного масштаба – сеть рабочей группы, сеть отдела, сеть отделения и, наконец, сеть предприятия в целом.

Для построения интегрированной системы управления различными элементами сети естественно применить многоуровневый иерархический подход. Это в принципе стандартный подход для построения большой системы управления любого типа и назначения. Применительно к системам управления сетями наиболее проработанным и эффективным для создания многоуровневой иерархической системы является стандарт Telecommunication Management Network (TMN), разработанный совместно ISO, ANSI и ETSI. Хотя этот стандарт и предназначался изначально для теле-

коммуникационных сетей, но ориентация на использование общих принципов делают его полезным для построения любой крупной интегрированной системы управления сетями. Модель TMN упрощенно можно представить в виде двухмерной диаграммы (рис. 45).

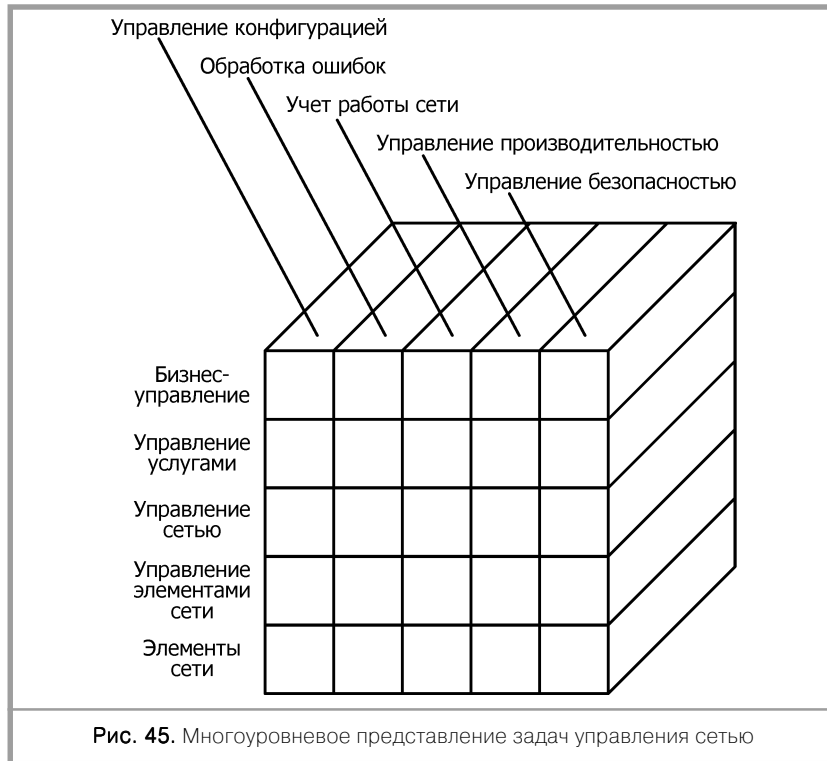


Рис. 45. Многоуровневое представление задач управления сетью

На каждом уровне иерархии модели TMN решаются задачи одних и тех же пяти функциональных групп, рассмотренных выше (то есть управления конфигурацией, ошибками, учетом, производительностью и безопасностью), однако на каждом уровне эти задачи имеют свою специфику. Чем выше уровень управления, тем более общий и агрегированный характер приобретает собираемая о сети информация, а сугубо технический характер собираемых данных начинает по мере повышения уровня меняться на производственный, финансовый и коммерческий.

Структура сети модели TMN (рис. 45):

- уровень элементов сети (Network Element layer, NE);
- уровень управления элементами сети (Network Element management layer);
- уровень управления сетью (Network management layer);
- уровень управления услугами (Service management layer);
- уровень бизнес-управления (Business management layer).

Элементы сети. Уровень элементов сети состоит из отдельных устройств сети: каналов, усилителей, оконечной аппаратуры, мультиплексоров, коммутаторов и т. п. Элементы могут содержать встроенные средства для поддержки управления – датчики, интерфейсы управления, а могут и представлять вещь в себе, требующую для связи с системой управления разработки специального оборудования – «устройства связи с объектом» (УСО). Современные технологии обычно имеют встроенные функции управления, которые позволяют выполнять хотя бы минимальные операции по контролю за состоянием устройства и за передаваемым устройством трафиком. Устройства, которые работают по протоколам, не имеющим встроенных функций контроля и управления, снабжаются отдельными блоками управления, которые поддерживают один из двух распространенных протоколов – SNMP или CMIP. Эти протоколы относятся к прикладному уровню модели OSI.

Управление элементами сети. Этот уровень представляет собой элементарные системы управления. Элементарные системы управления автономно управляют отдельными элементами сети – контролируют канал связи SDH (Synchronous Digital Hierarchy – синхронная дискретная иерархия), управляют коммутатором или мультиплексором. Уровень управления элементами изолирует верхние слои системы управления от деталей и особенностей управления конкретным оборудованием. Этот уровень ответствен за моделирование поведения оборудования и функциональных ресурсов нижележащей сети. Атрибуты этих моделей позволяют управлять различными аспектами поведения управляемых ресурсов. Обычно элементарные системы управления разрабатываются и управляются производителями оборудования.

Управление сетью – этот уровень координирует работу элементарных систем управления, позволяя контролировать конфигурацию составных каналов, согласовывать работу транспортных подсетей разных технологий и т. п. С помощью этого уровня сеть начинает работать как единое целое, передавая данные между своими абонентами.

Управление услугами занимается контролем и управлением за транспортными и информационными услугами, которые предос-

тавляются конечным пользователям сети. В задачу этого уровня входят подготовка сети к представлению определенных услуг, ее активизация, обработка вызовов клиентов. В функции этого уровня входит также выдача уровню управления сетью задания на конфигурирование виртуального или физического канала связи для поддержания услуги. После формирования услуги данный уровень занимается контролем за качеством ее реализации, то есть за соблюдением сетью всех принятых на себя обязательств в отношении производительности и надежности транспортных услуг. Результаты контроля качества обслуживания нужны, в частности, для подсчета оплаты за пользование услугами клиентами сети.

Уровень бизнес-управления занимается вопросами долговременного планирования сети с учетом финансовых аспектов деятельности организации, владеющей сетью. На этом уровне ежемесячно и поквартально подсчитываются доходы от эксплуатации сети и ее отдельных составляющих, учитываются расходы на эксплуатацию и модернизацию сети, принимаются решения о развитии сети с учетом финансовых возможностей. Этот уровень является частным случаем уровня автоматизированной системы управления предприятием (АСУП), в то время как все нижележащие уровни соответствуют уровням автоматизированной системы управления технологическими процессами (АСУТП). Если для такого специфического типа предприятия, как телекоммуникационная или корпоративная сеть, чаще всего является основой телекоммуникационной компании, то корпоративную сеть и обслуживающий ее персонал трудно назвать предприятием. Тем не менее в западных фирмах корпоративная сеть выделена со своим бюджетом и со своими финансовыми договорами на обслуживание, которое данное подразделение заключает с основными производственными подразделениями предприятия.

В основе всех систем управления сетями лежит схема «менеджер-агент», где агент является посредником между управляемым ресурсом и основной управляющей программой-менеджером (рис. 46). Эта схема использует абстрактную модель управляемого ресурса, называемую базой управляющей информацией (Management Information Base – MIB). Агент взаимодействует с управляемым ресурсом по нестандартному интерфейсу, а с менеджером – по стандартному протоколу через сеть. В больших системах управления используют несколько менеджеров, которые взаимодействуют друг с другом по одной из двух схем – одноранговой или иерархической.

Иерархическая схема взаимодействия менеджеров соответствует стандартам TMN (Telecommunication Management Network – управляющая коммуникационная сеть) и является более перспективной.

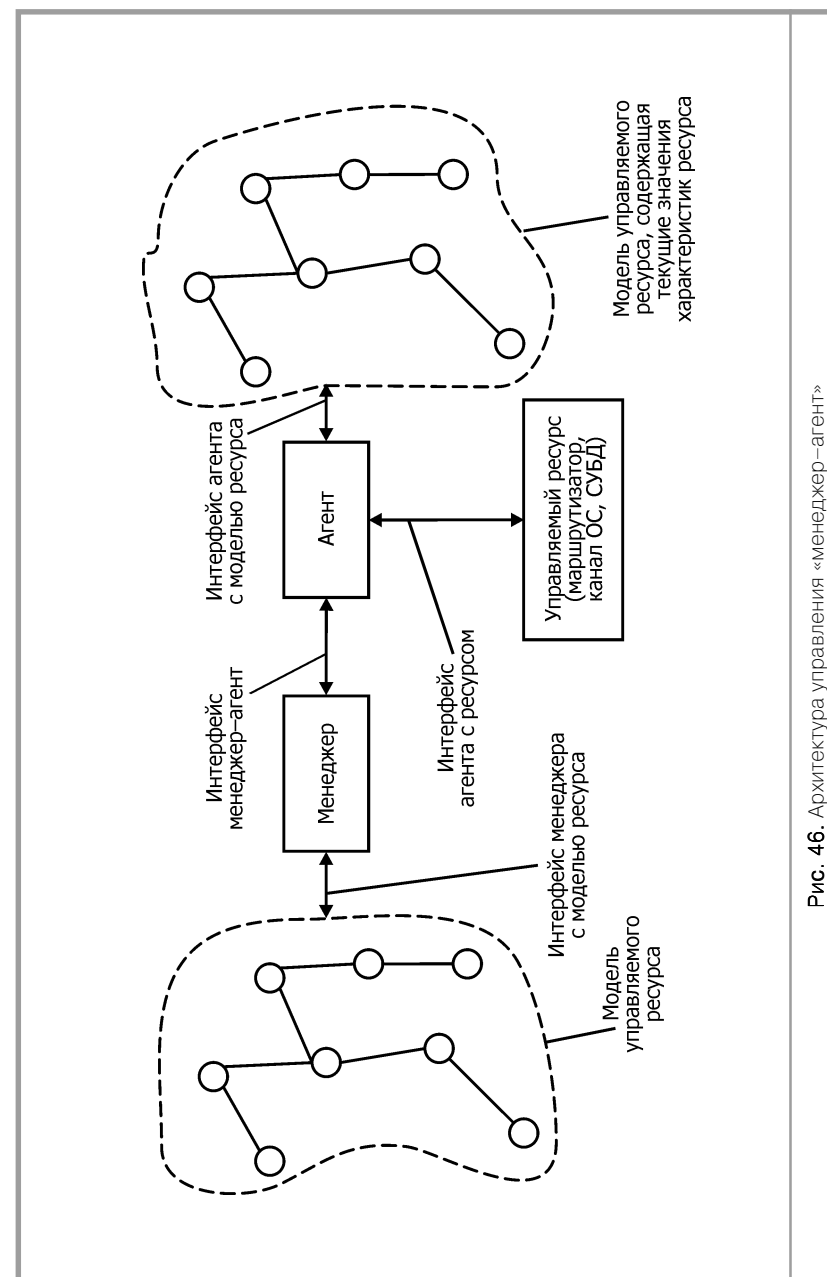


Рис. 46. Архитектура управления «менеджер-агент»

При построении систем управления активно используется платформенный подход (рис. 43). Платформа системы управления выполняют для менеджеров роль операционной системы для обычных приложений, так как обеспечивают разработчика менеджеров набором полезных системных вызовов, общим для любой системы управления назначения.

5.5. Средства и протоколы управления ЛВС

Средства управления ЛВС предназначены для реализации функций в рамках пяти категорий управления, определенных международной организацией по стандартизации. Эти средства входят в состав системы управления ЛВС и включают четыре типа средств: контрольно-измерительные приборы, сетевые мониторы, сетевые анализаторы и интегрированные системы управления сетями.

Из **контрольно-измерительных приборов** наиболее распространенными являются рефлектометры, осциллографы, детекторы разрывов, измерителя мощности.

Сетевой монитор представляет собой компьютер, подключенный к ЛВС для контроля трафика всей сети или выделенной ее части. Являясь автономной функциональной частью сети, они работают непрерывно, набирая информацию об использовании сети, типах пакетов сообщений каждым узлом ЛВС.

Сетевые анализаторы являются сложными и дорогостоящими устройствами, обладающими более широкими возможностями, чем кабельные тестеры, в которые входят рефлектометры. Они применяются не только для обнаружения неполадок в сети, но и для выяснения их причин и устранения.

Интегрированные системы управления (ИСУ) ЛВС реализуют функции по всем пяти категориям управления вычислительной сетью, определенным ISO. При использовании ИСУ контроль всей сети осуществляется из единого центра с помощью терминала с графическим пользовательским интерфейсом, интегрированным со станцией управления сетью.

Протоколы управления ЛВС специально разработаны и используются для диагностики работоспособности различных локальных сетей.

SNMP (Simple Network Management Protocol) – простой протокол для управления вычислительной сетью, предназначен для решения коммуникационных проблем в сетях TCP/IP (в настоящее время область его применения расширена: его возможности позволяют контролировать сетевой трафик и выявлять аппаратные неисправности и узкие места в широком диапазоне не только TCP/IP сетевых устройств).

CMIP (Common Management Information Protocol) – протокол общего управления информацией, предназначен для решения коммуникационных проблем в сетях модели ISO и являются частью этой стандартной модели. Это стандарт управления для сетей, соответствующих модели ISO.

Стандарты систем управления, основанных на протоколе SNMP, формализуют минимум аспектов системы управления, а стандарты ISO – максимум аспектов, как и большинство стандартов, разработанных ITV-T.

Традиционно в локальных и корпоративных сетях применяются в основном системы управления на основе SNMP, а стандарты ISO/ITU-T и протокол CMIP находят применение в телекоммуникационных сетях.

Основное сходство протоколов SNMP и CMIP (кроме общей цели, состоящей в облегчении задач управления и диагностики при работе в ЛВС) заключается в использовании одной и той же концепции MIB и ее расширении (Management Information Base – База управления информацией). Концепция состоит из набора переменных, тестовых точек и контрольных параметров, которые поддерживаются всеми устройствами сети и могут контролироваться администратором ЛВС.

Сравнение протоколов SNMP и CMIP:

- протокол SNMP ориентирован на связь без соединения с целью сокращения накладных расходов и обеспечения управления на пользовательском уровне. Для передачи запросов или ответов при управлении ЛВС в SNMP используются простые дейтаграммы. В этом случае связывающиеся стороны должны предусматривать возможность неполучения данных адресатом и, следовательно, быть готовыми к повторным передачам. Для маршрутизации сообщений в SNMP могут использоваться простые коммуникационные протоколы (IPX или IP и UDP). Протокол CMIP ориентирован на связь с соединением, обеспечивающим прозрачную обработку параметров. Использование в этом протоколе сеансового обмена информацией делает его более удобным при необходимости получения большого количества данных. Однако это может затруднить управление сетью при возникновении неполадок;
- протокол CMIP содержит гораздо более надежный набор средств сетевого управления, чем SNMP. Он обеспечивает шесть типов услуг: управление конфигурацией, управление защитой, контроль неисправностей, учет, управление качеством функционирования и службу каталогов. Серьезным недостатком SNMP является отсутствие средств защиты;

- применение протокола SNMP позволяет строить как простые, так и сложные системы управления, а применение протокола CMIP определяет некоторый, достаточно высокий начальный уровень сложности системы управления, так как для его работы необходимо реализовать ряд вспомогательных служб, объектов и баз данных объектов;
- в протоколе SNMP не различаются объект и его атрибуты (объект может быть устройством, а атрибут – характеристикой или параметром этого устройства). Это означает, что в среде SNMP приходится формировать новые определения для каждого из устройств, которые создаются для SNMP-сети. При работе в среде CMIP для новых устройств используются уже созданные определения, включаются только дополнительные атрибуты, чтобы было можно отличить новые устройства;
- протокол SNMP работает через периодические опросы устройств сети для определения их статуса. В протоколе CMIP используются отчеты устройств, в которых они информируют центральную управляющую станцию об изменениях в своем статусе;
- система управления сетью на базе протокола SNMP отличается большей компактностью, большим быстродействием и меньшей стоимостью. Изделия на базе SNMP получили большее распространение. Протокол CMIP еще не получил широкого применения, потому что пока мало сетей, работающих по протоколам модели OSI.

5.6. Службы безопасности ЛВС

Защита информации в компьютерных сетях становится одной из самых острых проблем в современной информатике. Сформулированы три базовых принципа информационной безопасности, которая должна обеспечить:

- целостность данных (защиту от сбоев, ведущих к потере информации, а также неавторизованного создания или уничтожения данных);
- конфиденциальность информации;
- доступность информации для всех авторизованных пользователей.

Обеспечение безопасности в компьютерных сетях требует защиты следующих информационных ресурсов:

- общесистемное программное обеспечение (ОС, СУБД, офисные приложения и т. п.);
- прикладное программное обеспечение;
- информационное обеспечение (БД, файлы).

Нарушителями нормального функционирования ЛВС могут быть:

- администраторы ЛВС;
- пользователи, допущенные к обработке информации ЛВС;
- технический персонал, обслуживающий вычислительную технику, но не допущенный к обработке информации в ЛВС в повседневной деятельности;
- удаленные пользователи, не допущенные к обработке информации в ЛВС;
- разработчики программного обеспечения и вычислительной техники, используемых в ЛВС.

Возникновение угроз информационным ресурсам ЛВС может происходить вследствие следующих причин:

- 1) несанкционированного доступа (НСД) злоумышленников (взломщиков сети – хакеров):
 - копирование, уничтожение или подделка информации;
 - ознакомление с конфиденциальной информацией посторонних лиц;
- 2) ошибок обслуживающего персонала и пользователей:
 - случайное уничтожение или изменение данных;
 - некорректное использование программного обеспечения, ведущее к уничтожению или изменению данных;
- 3) некорректной работы программного обеспечения:
 - при ошибках в программном обеспечении или внесении в него недеklarированных возможностей;
 - при заражении системы компьютерными вирусами;
- 4) неправильного хранения архивных данных;
- 5) сбоев оборудования:
 - кабельной системы;
 - электропитания;
 - дисковых систем;
 - систем архивации данных;
 - серверов, рабочих станций, сетевых карт и т. п.;
- 6) пути несанкционированного доступа к защищаемой информации:
 - перехват побочных электромагнитных излучений;
 - перехват в радиосетях;
 - принудительное магнитное облучение (подсветка) линий связи с целью паразитной модуляции несущей частоты;
 - применение подслушивающих устройств (закладок);
 - перехват акустических излучений и восстановление текста принтера;
 - хищение носителей информации;
 - копирование информации с ее носителей с преодолением средств защиты;

- маскировка под зарегистрированного пользователя;
 - маскировка под запросы системы;
 - использование программных ловушек;
 - незаконное подключение к аппаратуре и линиям связи;
 - злоумышленный вывод из строя средств защиты;
 - внедрение и использование компьютерных вирусов;
- 7) «эффекты», вызываемые вирусами, принято делить на следующие целевые группы:
- искажение информации в файлах либо таблицы разрешения файлов, которое может привести к разрушению файловой системы в целом;
 - имитация сбоя программных средств;
 - создание визуальных и звуковых эффектов, включая отображения сообщений, вводящих операторов в заблуждение или затрудняющих их работу;
 - инициирование ошибок в программах пользователей или операционных систем.

Концепция обеспечения безопасности информационных ресурсов ЛВС должна базироваться на технологии многоуровневой защиты (MLS – Multilevel Security), в которую входят:

- физическая защита кабельной системы, электропитания, средств архивации, дисковых массивов и т. д.;
- организация мер защиты вычислительной техники, кадровый подбор и расстановка персонала, контроль доступа в помещения и парольная защита компьютеров, планы действия в чрезвычайных ситуациях и т. д.;
- программно-технические средства защиты информации, вычислительной техники и систем передачи данных, а именно – средства защиты от НСД (средства поиска и обнаружения закладных устройств НСД к информации); межсетевые экраны; средства шифрования (криптографии) информации, передаваемой по каналам связи; электромагнитное управление техники или помещений, в которых расположена техника; активная радиотехническая маскировка с использованием широкополосных генераторов шума; электронно-сетевая подпись; антивирусные программы; системы разграничения доступа; системы мониторинга и анализа уязвимости ЛВС.

Нейтрализация перечисленных и других угроз безопасности осуществляется службами сети и механизмами реализации функций этих служб. Документами Международной организации стандартизации (МОС) определены следующие службы безопасности:

- аутентификация (подтверждение подлинности);
- обеспечение целостности передаваемых данных;

- контроль доступа;
- защита от отказов.

Первые три службы характеризуются различиями для виртуальных и дейтаграммных сетей, а последние две службы инвариантны по отношению к этим сетям.

В виртуальных сетях используются протоколы информационного обмена типа виртуального соединения. Передача информации между абонентами организуется по виртуальному каналу и происходит в три этапа: создание (установление) канала, собственно передача и уничтожение (разъединение) канала. При этом сообщения разбиваются на одинаковые части (пакеты). Пакеты передаются по виртуальному каналу в порядке их следования в сообщении.

В дейтаграммных сетях реализуются дейтаграммные протоколы информационного обмена. Пакеты, принадлежащие одному и тому же сообщению, передаются от отправителя к получателю в составе дейтаграмм независимо друг от друга и в общем случае по различным маршрутам, то есть в сети они являются самостоятельными единицами информации. На приемном пункте из пакетов, поступивших по различным маршрутам и в разное время, составляется первоначальное сообщение.

Службы и механизмы безопасности используются на определенных уровнях эталонной модели ВОС.

На рис. 47 представлено распределение служб безопасности (СБ) по уровням эталонной модели ВОС, а на рис. 48 – механизмы реализации служб безопасности.

Служба аутентификации одноуровневого объекта обеспечивает подтверждение (опровержение) того, что объект, предлагающий себя в качестве отправителя сообщения по виртуальному каналу, является именно таковым как на этапе установления связи между абонентами, так и на этапе передачи сообщения.

В дейтаграммных сетях эта служба называется службой аутентификации источника данных, передаваемых в виде дейтаграмм.

Службы целостности обеспечивают выявление искажений в передаваемых данных. Они разделяются по виду сетей, в которых они применяются (СБ в виртуальных и дейтаграммных сетях), по действиям, выполняемым при обнаружении аномальных ситуаций (с восстановлением данных или без восстановления), по степени охвата передаваемых данных (сообщение или дейтаграмма в целом либо их части, называемые выборочными полями).

Службы засекречивания обеспечивают секретность передаваемых данных: в виртуальных сетях – всего пересылаемого сообщения или только его выборочных полей, а дейтаграммных – каждой дейтаграммы или только отдельных ее элементов.

В рамках построения защиты ЛВС возможен выбор одной из двух концепций:

- создание системы безопасности, построенной на базе каналов связи и средств коммутации территориальных сетей связи общего использования, в которых применяются открытые протоколы Интернета;
- отказ от средств Интернета, создание локальной сети на базе специализированной или выделенной сети связи с использованием конкретной сетевой технологии, в частности ATM, FR, ISDN.

5.7. Организация одноранговой сети в среде «Microsoft Windows»

Сетевая операционная система «Microsoft Windows» – популярная графическая операционная оболочка, которая запускается на выполнение как обычная программа MS DOS, работает на базе MS DOS и в совокупности с ней образует полноценную операционную систему.

Windows, именуемая также **Chicago**, является 32-разрядной многозадачной и многопоточной системой с приоритетами, которая имеет широко используемый интерфейс прикладной программы (API).

Операционная система предоставляет разнообразные средства для распределенной обработки данных. Она позволяет рабочим группам использовать общие файлы и прикладные программы.

Windows создает среду для объектно-ориентированной архитектуры и выполняет разнообразные функции, связанные с определением и изменением конфигурации внешних устройств и программного обеспечения, работающих в сети. В соответствии с этим она ведет учет, используя технологии Plug-and-Play и технологию связи и компоновки объектов (OLE). Передача данных осуществляется в соответствии с протоколом IPX/SPX. При этом выполняются также функции телефонии, факсимильной связи и обеспечивается взаимодействие с телефонной сетью. Обеспечиваются защита от отказов и безопасность данных.

Windows работает с любыми типами данных: текстами, звуками и изображениями. Система позволяет управлять громкостью звуков и яркостью изображения. Используется удобный упрощенный интерфейс пользователя, позволяющий также работать с трехмерной графикой. Обеспечивается соответствие цветов экрана и распечаток на принтере. Пиктограммы дают возможность обращаться к внешним устройствам и сетевым службам. Осуществляется также обработка сигналов. Поддерживается работа с электронными перьями.

Windows имеет модуль, являющийся универсальным почтовым ящиком, предназначенным для хранения сообщений электронной почты, речевой почты и факсимильной связи. Ящик взаимодействует с групповым программным обеспечением в соответствии с телефонным программным интерфейсом прикладной программы (TAPI).

Сжатие аудио- и видеoinформации, выполняемое системой, позволяет эффективнее использовать память. Управление энерго-снабжением дает возможность экономить электрическую энергию, выполнять ряд требований, связанных с концепцией «зеленого компьютера».

Инсталляция внешних устройств и программного обеспечения может осуществляться централизованно с сервера.

Ключевыми новинками Windows являются:

- простота в работе, достигаемая использованием нового пользовательского интерфейса, технологии поддержки самонастраиваемой аппаратуры Plug-and-Play, допустимостью применения данных имен файлов, встроенной сетевой поддержкой, усовершенствованной обучающей и справочной системой;
- повышенная производительность, обновленная такими свойствами, как истинная вытесняющая многозадачность и масштабируемость рабочих характеристик для 32-битовых приложений, повышенная устойчивость к сбоям, ускоренная печать, наличие высокоэффективных мультимедийных компонентов, увеличенный объем памяти для MS DOS;
- совместимость с существующими приложениями MS DOS и Windows, поддержка любого существующего оборудования и драйверов устройств, сетевая совместимость с существующими клиентами реального времени и процедурами регистрации.

Windows – мощная 32-битовая многопоточная система, имеющая ряд преимуществ:

- интегрированная операционная система, ядро которой, загружаемое в момент включения компьютера, активирует графический интерфейс пользователя и обеспечивает полную совместимость с операционной системой MS DOS;
- вытесняющая многозадачность, то есть свойство операционной системы самостоятельно в зависимости от внутренней ситуации передавать или забирать управление у того или иного приложения;
- многопоточность – свойство операционной системы выполнять операции одновременно под потоками нескольких 32-битовых приложений, называемых процессами.

В Windows широко используется технология Plug-and-Play, обеспечивающая новые возможности интеграции программных продуктов и аппаратных средств. Она ориентирована на поддержку любого типа устройств, включая мониторы, видеоплаты, принтеры, звуковые карты, модемы, приводы CD-ROM, различные контроллеры жестких дисков.

Топология Plug-and-Play (PuP) – способ создания или реконструкции абонентской системы быстрой установкой либо заменой ее компонентов – основана на использовании объектно-ориентированной архитектуры, а ее абонентами являются внешние устройства сети и программы. Она позволяет решать следующие задачи:

- подключать к системе разнообразные внешние устройства без выполнения процедур настройки на их использование;
- вынимать программы из одного и передавать их в другой набор, при этом эти программы должны иметь один и тот же интерфейс;
- помощь при распознавании устройств для их установки и настройки;
- уведомление программных продуктов и приложений и динамическое изменение состояния системы.

Windows отвечает потребностям администраторов сетей и обладает интегрированной, высокопроизводительной и легкоуправляемой 32-битовой сетевой архитектурой. В то же время Windows отвечает и потребностям пользователей, представляя средства доступа к сети и ее управлению. А также упрощая поиск, просмотр и печать файлов по сети благодаря совершенному пользовательскому интерфейсу.

Windows – высокоэффективная платформа для мультимедиа.

Основными достоинствами Windows в мире мультимедиа являются следующие:

- простота подключения средств мультимедиа;
- значительная мощность платформы Windows для разработчиков мультимедиа-программ;
- новые возможности для изготовителей компьютерного оборудования.

Сетевая операционная система Windows NT предназначена для архитектуры клиент-сервер и обеспечивает независимо от конкретного типа сети возможность использования различных протоколов транспортного уровня. Она является многозадачной – выполняет сразу несколько прикладных программ, используя идеи операционной системы UNIX. Основными целями рассматриваемой системы являются: блочность структуры, легкая расширяемость, мигрируемость (переход от одной аппаратной платформы к дру-

гой), предоставление удобных интерфейсов пользователя. Система организует несколько параллельно обрабатываемых потоков данных.

Windows NT имеет 32-разрядную архитектуру и поддерживает локальные сети с широким кругом выполняемых функций. К ним относятся:

- возможность каждой абонентской системы в сети быть сервером либо компонентом;
- совместная работа группы пользователей;
- предоставление информации в удобной и понятной форме;
- секретность большого числа файлов;
- работа электронной почты;
- адресация большой памяти;
- запуск одновременно несколько прикладных процессов;
- поддержка мультипроцессорных систем;
- управление сетью;
- обеспечение безопасности данных;
- инсталляция системы с компактных дисков.

Благодаря эффективному процессу сжатия данных система обеспечивает увеличение логической памяти магнитного диска на 200÷400%.

Сетевая организационная система Windows NT погружается в прикладную среду и работает с большим набором типов микропроцессоров корпораций Intel, IBM, DEC.

Сетевая операционная система Windows NTAS, основой которой является сетевая операционная система Windows NT, предназначена для архитектуры клиент-сервер и одноранговой архитектуры. Система осуществляет симметричную мультипроцессорную обработку и управляет компьютерами, каждый из которых имеет до четырех процессоров. Важнейшими особенностями рассматриваемой сетевой операционной системы являются:

- наличие разнообразных сетевых средств;
- работа с большой оперативной памятью и внешней памятью;
- поддержка различных адаптеров и принтеров;
- выполнение Протокола управления передачей/Межсетевым протоколом (TCP/IP), протокола IPX/SPX или сетевой базовой системы ввода-вывода (NetBIOS);
- простота процедуры установки с помощью мыши;
- разнообразие средств безопасности данных, назначение имени пользователя (до 20 символов), пароля (до 14 символов), допустимых периодов входа в систему, дата окончания срока подключений.

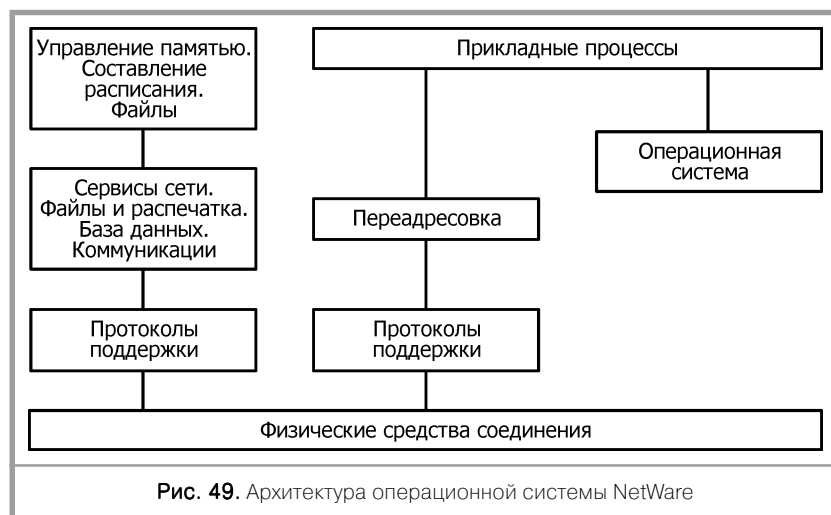
В системе NTAS используются четыре стратегии безопасности данных:

1. Стратегия бюджетов контролирует время доступа пользователей к ресурсам сети.
2. Стратегия прав контролирует разрешения пользователям на выполнение различных функций (восстановление файлов, останова системы, резервное копирование документов и т. д.).
3. Стратегия отслеживания описывает события в сети, которые должны регистрироваться в журнале защиты.
4. Стратегия доменов контролирует взаимосвязи между последними. На этой стратегии основаны службы каталогов.

NTAS позволяет создавать общие и местные рабочие группы пользователей. Члены общих групп могут также входить в состав местных групп. Общие группы состоят из пользователей одного домена.

5.8. Организация работы в сети Novel NetWare

Сетевая операционная система NetWare была создана формой Novel для локальных сетей и является самой распространенной сетевой операционной системой. NetWare использует архитектуру клиент-сервер либо одноранговую архитектуру и применяется в смешанных сетях (рис. 49). Система может также опираться на асинхронный способ передачи (ATM). Для передачи данных опера-



ционная система в основном использует Протокол IPX/SPX. При необходимости загружаемый модуль NetWare (NLM) позволяет серверам и клиентам использовать Протокол управления передачей/Межсетевой протокол (TCP/IP). По имени операционной системы появились сети NetWare.

Система предоставляет разнообразные виды сервиса, в том числе:

- справочную службу, информирующую обо всех ресурсах сети;
- выполнение общих прикладных процессов на серверах;
- создание и ликвидацию логических каналов и соединений;
- работу с файлами, размещенными на разных серверах;
- передачу данных между локальными сетями;
- работу с базами данных серверов;
- использование магнитных дисков, компактных дисков и принтеров серверов;
- синхронизацию группы взаимосвязанных прикладных программ, работающих в различных системах;
- электронную почту;
- электронную доску объявлений;
- использование аудио- и видеосистем NetWare;
- поддержку массивов дисков RAID;
- обеспечение симметричной мультипроцессорной обработки;
- электронное распространение программного обеспечения;
- каталог объектов сети;
- обеспечение безопасности данных;
- организацию функционирования рабочих групп.

NetWare определяет архитектуру сети, ее функциональный профиль, протоколы, сетевые службы, программное обеспечение и устройства, реализующие поставленные цели. Система состоит из микроядра и набора функциональных блоков, обеспечивающих использование различных протоколов передачи данных и опору на различные микропроцессоры.

Рассматриваемая сетевая операционная система обеспечивает создание не только одной, но и группы сетей NetWare. Более того, доступ к NetWare удаленных клиентов и серверов осуществляется через различные типы территориальных сетей (рис. 50).

Сетевую операционную систему NetWare можно рассматривать как распределенную операционную систему, модули которой располагаются на файл-сервере и на рабочих станциях. Структура сетевой оболочки операционной системы представлена на рис. 51.

Рабочие станции сети – персональные компьютеры. Пользователи, работающие на них, имеют свой набор файлов и собственную операционную систему: MS DOS, OS/2 и др. Так как операционная система NetWare отличается от операционных систем рабочих

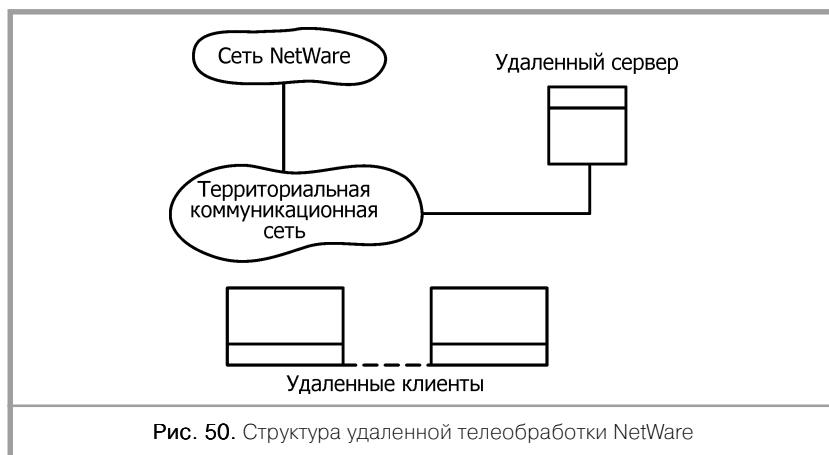


Рис. 50. Структура удаленной телеобработки NetWare

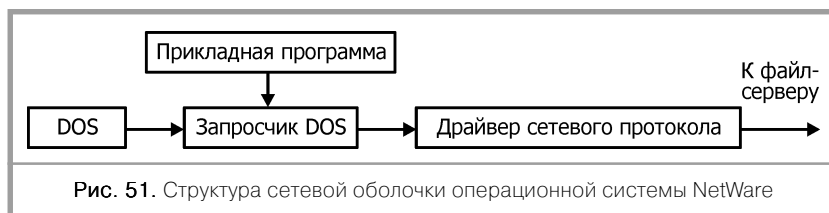


Рис. 51. Структура сетевой оболочки операционной системы NetWare

станций, то рабочие станции взаимодействуют с файл-сервером через специальную сетевую оболочку (рис. 51). В состав оболочки входят четыре резидентные программы, основной из которых является запросчик DOS (DOS Requester). Главная задача его – перенаправление запросов от прикладных программ на доступ к файлам либо на локальный диск, либо на сетевой диск. Если файлы находятся на сетевом диске, то запросчик возвращает запрос в DOS. Если же файлы находятся на сетевом диске, то производится обслуживание запроса: определяется, с какого сервера, с какого тома или каталога необходимо прочитать файл.

Распределенная служба каталогов NDS – NetWare Directory Service – хранит информацию обо всех пользователях сети, о серверах и ресурсах сети в специальной базе данных. Доступ к этой базе данных имеют как администраторы, так и пользователи сети.

Стратегия защиты сетевых ресурсов от несанкционированного доступа сети Novell NetWare включает в себя ограничения, накладываемые на пользователя, и ограничения, накладываемые на каталоги и файлы.

В состав ограничений, накладываемых на пользователя, входят:

- защита паролем;
- ограничение числа конкурирующих соединений;
- ограничение попыток ввода пароля;
- ограничение времени входа в сеть.

Для введения ограничения доступа к файлам и каталогам в системе NetWare устанавливаются восемь типов прав доступа, то есть возможность выполнять в сети определенные операции с данными, предоставляемая пользователю сетевой операционной системы. Кроме того, для файлов и каталогов установлены специальные признаки-атрибуты, которые определяют возможность работы с файлами и каталогами для того или иного типа пользователя. Так как ограничения, накладываемые на пользователя, зависят от их типа или категории, то операционная система NetWare определяет категории пользователей сети: администратор (супервизор), пользователь рабочей станции, оператор, аудитор.

Администратор отвечает за правильную и бесперебойную работу данной сети и управляет работой всей системы. Он может вводить и удалять пользователя, назначать права доступа, обновлять и конфигурировать всю сеть.

Пользователи рабочих станций – это конечные пользователи, регулярно работающие с сетью. Они получают в свое распоряжение файловое пространство, доступ к разделяемым ресурсам базы данных, возможность выхода в другие сети и доступ к сетевой печати.

Операторы сети – это те же пользователи, имеющие некоторые дополнительные возможности по управлению сетью (управление сетевой печатью, работой сервера печати и т. д.).

Аудитор – пользователь, который может собирать различные статистические данные о сети и о событиях, происходящих в ней, без контроля администратора сети.

В сети могут существовать несколько групп пользователей, каждой из которых назначается менеджер. Он может создавать и удалять группы пользователей, включать и удалять отдельных пользователей группы, назначать опекунские права и присваивать статус менеджера отдельным пользователям.

Всех остальных пользователей назначает администратор. Он присваивает им имена и пароли. Каждый новый пользователь по умолчанию получает право пользоваться сетевой печатью и доступ к утилитам, находящимся в каталоге PUBLIC.

Для того чтобы работать в сети, пользователю необходимо знать или уметь выяснять свое сетевое имя и расписание работы. Эту задачу помогает решить единственный доступный пользователю (после загрузки сетевой оболочки) каталог на файл-сервере – LOGIN.

С помощью утилиты NLIST пользователь определяет доступные для него файл-серверы данной сети.

С помощью утилиты LOGIN пользователь регистрируется в сети на выбранном файл-сервере.

Подключение в процессе работы к другому файл-серверу обеспечивает утилита ATTACH.

Для подключения одного из файл-серверов или для выхода из сети используется утилита LOGOUT.

Система NetWare предоставляет пользователю четыре уровня защиты:

- защита именем регистрации и паролем;
- защита правилами опекуна;
- защита каталогов фильтров наследуемых прав;
- защита каталогов и файлов при помощи атрибутов.

Защита именем регистрации и паролем устанавливается администратором сети с помощью утилиты NETAMIN в режиме «возможности супервизора». С ее же помощью устанавливается ограничение числа конкурирующих соединений и числа допустимых вводов неправильного пароля. Эти ограничения вводятся для всех пользователей.

Опекунские права (Trustec Rights) предоставляются для работы с файлами определенного каталога и на отдельные файлы как пользователям, так и группам. NetWare предусматривает назначение восьми видов опекунских прав: S (Supervisor) – супервизорное право; R (Read) – чтение из файла; W (Write) – запись в файл; A (Access Control) – контроль доступа; C (Create) – создание новых файлов и каталогов; E (Erase) – удаление файлов и каталогов; F (File Scan) – поиск файлов и каталогов; M (Modify) – модификация атрибутов файлов, переименование файлов и каталогов.

Фильтр наследуемых прав (Inherited Rights Filter) максимально включает все опекунские права и сокращенно записывается в виде SRWCEMFA. Его применяют для каталогов, которые совместно используются многими пользователями сети. Он позволяет фильтровать права только сверху вниз. Изменение фильтра можно выполнить с помощью команды MAP или утилиты FILTER.

В операционной системе NetWare пользователь может просмотреть свои права в разных каталогах с помощью утилиты NETADMIN. Установка маски прав, назначение опекунов каталогам и изменение опекунских прав производятся с помощью утилиты FILTER.

Кроме утилит меню, используются и утилиты командной строки RIGHTS.

Защита файлов и каталогов атрибутами необходима при работе в вычислительной сети, имеющей большие информационные фай-

лы, к которым имеет доступ большое количество пользователей. При разрушении или искажении информации файла восстановить его практически невозможно. Применение атрибутов дает возможность пользователям регулировать процесс совместного использования его различных файлов и каталогов и модификации их. В сети NetWare для защиты файлов имеются 13 атрибутов, а для каталогов – 9 атрибутов.

Любой созданный файл по умолчанию получает атрибуты Read-Write (Rw) – файл для чтения и записи, и он является файлом неразделимым.

Корпоративные информационные приложения



6.1. Корпоративно-функциональные компоненты информационной системы

Процессы, обеспечивающие работу информационной системы (ИС), можно представить в виде блочной схемы (рис. 52):

- ввод информации из внешних или внутренних источников;
- обработка входной информации и представление ее в удобном виде;
- вывод информации для представления потребителям или передаче в другую систему;
- обратная связь – это информация, переработанная людьми данной организации для коррекции входной информации.

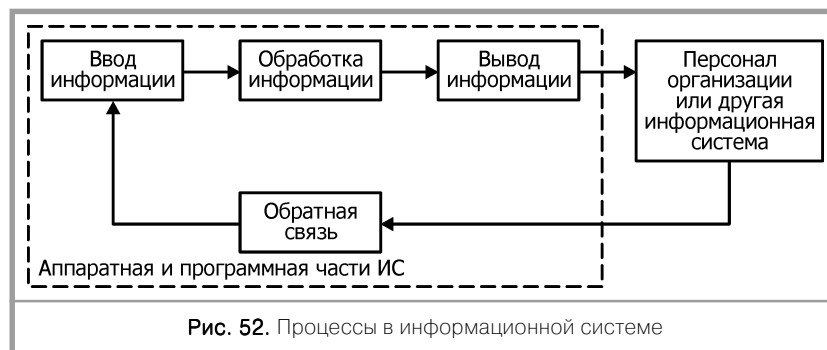


Рис. 52. Процессы в информационной системе

В настоящее время любая фирма имеет несколько локальных информационных систем различного назначения, которые взаимодействуют между собой и поддерживают управленческие решения на всех уровнях. На рис. 53 показан один из таких вариантов. Между локальными ИС организуются связи различного характера и назначения. Одни локальные информационные системы могут быть связаны с большим количеством работающих в фирме систем и иметь выход во внешнюю среду, другие связаны только с одной и несколькими родственными ЛВС. Современный подход к организации связи основан на применении локальных внутрифирменных компьютерных сетей с выходом на аналогичную информационную систему другой фирмы или подразделения корпорации. При этом пользуются ресурсами региональных и глобальных сетей.

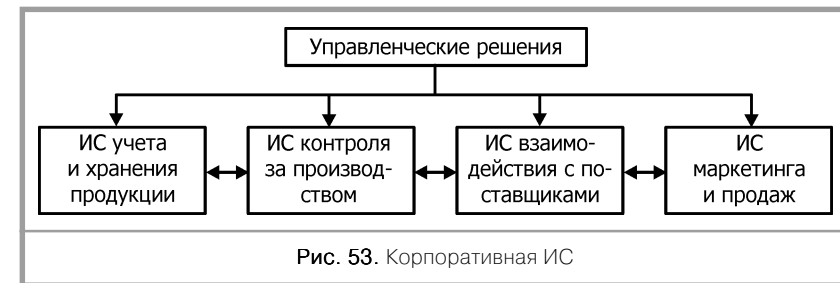


Рис. 53. Корпоративная ИС

На основе интеграции информационных систем разного назначения с помощью компьютерных сетей в фирме создаются корпоративные информационные системы или корпоративные вычислительные сети (КВС). Подобные КВС предоставляют пользователю возможность работать как с общефирменной базой данных, так и с локальными базами данных.

Наблюдаемый в настоящее время громадный рост корпоративных сетей (в 2002 г. используются до 4 млн серверов КВС) объясняется их преимуществами по совместному использованию информации, сотрудничеству, быстрому доступу к данным и наличию большого числа пользователей, уже знакомых с необходимым программным обеспечением по работе в Интернете.

Корпоративная сеть, объединяющая локальные сети отделений и предприятий корпорации (организации, компании), является материально-технической базой для решения задач планирования, организации и осуществления ее производственно-хозяйственной деятельности. Она обеспечивает функционирование автоматизи-

рованной системы управления и системы информационного обслуживания корпорации.

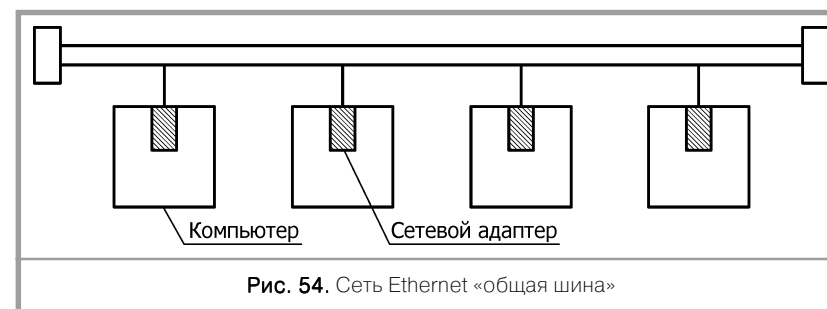
Эффективность использования КВС зависит от успешного решения как технологических, так и организационных вопросов, причем по мере эксплуатации сети, когда технологические вопросы получили должное разрешение, все большее значение приобретают организационные вопросы. Ключевыми факторами успешного и эффективного функционирования КВС являются рациональное распределение информации, необходимой для планирования, организации и осуществления производственно-хозяйственной деятельности корпорации; обеспечение сотрудников корпорации системами управления документооборотом и предоставление доступа к различным корпоративным базам данных, воспитание культуры совместного использования информации. Основное внимание должно быть направлено на потребности пользователей, а не на расширение технологических возможностей сети.

Общие и основные подходы к решению наиболее важных проблем построения КВС воплощены в наиболее популярной сетевой технологии – Ethernet.

Сетевая технология – это согласованный набор стандартных протоколов и реализующих их программно-аппаратных средств (например, сетевых адаптеров, драйверов, кабелей и разъемов), достаточный для построения вычислительной сети. Эпитет «достаточный» подчеркивает то обстоятельство, что этот набор представляет собой минимальный набор средств, с помощью которых можно построить работоспособную сеть. Такую сеть можно улучшить выделением в ней подсетей, что сразу потребует, кроме протоколов стандарта Ethernet, применения протокола IP, а также специальных коммутационных устройств – маршрутизаторов.

Протоколы, на основе которых строится КВС, специально разрабатывались для совместной работы сетей, поэтому от разработчика сети не требуется дополнительных усилий по организации их взаимодействия. Иногда сетевые технологии называют «базовыми технологиями», имея в виду то, что на их основе строится базис любой сети. Примерами базовых сетевых технологий КВС могут служить наряду с Ethernet такие известные технологии локальных сетей, как Token Ring и FDDI, или же технологии территориальных сетей X.25 и Frame Relay. Для получения работоспособной сети в этом случае достаточно приобрести функциональные компоненты, то есть программные и аппаратные средства, относящиеся к одной базовой технологии (например, КВС) – сетевые адаптеры с драйверами, концепторы, коммутаторы, кабельную систему, модемы и т. п.) – и соединить их в соответствии с требованиями стандарта на данную технологию.

Основной принцип, положенный в основу Ethernet, – случайный метод доступа к разделяемой среде передачи данных. В качестве такой среды могут использоваться толстый или тонкий коаксиальный кабель, витая пара проводов, оптоволокно или радиоканал. В стандарте Ethernet строго зафиксирована топология электрических связей. Компьютеры подключаются к разделяемой среде в соответствии с типовой структурой «общая шина» (рис. 54). С помощью разделяемой во времени шины любые два компьютера могут обмениваться данными. Управление доступом к линии связи осуществляется специальными контроллерами – **сетевыми адаптерами** Ethernet. Каждый компьютер, а более точно, каждый сетевой адаптер, имеет уникальный адрес. Передача данных происходит со скоростью 10 Мбит/с. Эта величина является пропускной способностью сети Ethernet.



Главным достоинством сетей Ethernet, благодаря которому они стали такими популярными, является их экономичность. Для построения сети достаточно иметь по одному сетевому адаптеру для каждого компьютера плюс один физический сегмент коаксиального кабеля нужной длины.

Другие базовые технологии, например Token Ring, для создания даже небольшой среды требуют наличия дополнительного устройства – концентратора.

В сетях Ethernet реализованы достаточно простые алгоритмы доступа к среде, адресации и передачи данных. Простая логика работы сети ведет к упрощению и, соответственно, удешевлению сетевых адаптеров и их драйверов, а значит, и к высокой их надежности в работе. Кроме того, сети Ethernet обладают хорошей расширяемостью, то есть легко подключаются к новым узлам.

6.2. Классификация корпоративных сетей

Корпоративная сеть – это сеть, главным назначением которой является поддержание работы конкретного предприятия, владеющего данной сетью. Пользователями корпоративной сети являются сотрудники данного предприятия. В зависимости от масштаба предприятия, а также от сложности и многообразия решаемых задач различают сети отдела, сети кампуса и корпоративные сети (то сеть большого предприятия).

Сети отдела – это сети, которые используются сравнительно небольшой группой сотрудников, работающих в одном отделе предприятия.

Главной целью сети отдела является разделение локальных ресурсов, таких как приложения, данные, лазерные принтеры и модемы. Обычно сети отделов имеют один и два файловых сервера, не более тридцати пользователей и не разделяются на подсети (рис. 55). В этих сетях локализуется большая часть трафика предприятия. Сети отделов обычно создаются на основе какой-либо одной сетевой технологии – Ethernet, Token Ring. Для такой сети характерен один или максимум два типа операционных систем. Небольшое количество пользователей дает возможность использования в сети отделов одноранговых сетевых операционных систем, таких как Windows Microsoft.

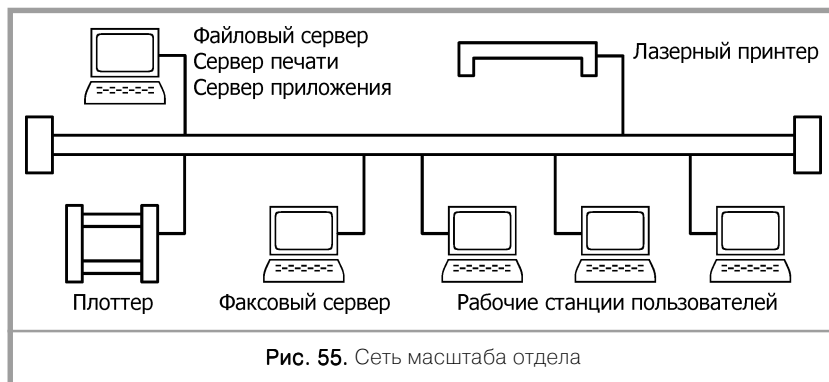


Рис. 55. Сеть масштаба отдела

Существует и другой тип сетей, близкий к сетям отделов, – **сети рабочих групп**. К таким сетям относят совсем небольшие сети, включающие до 10–20 компьютеров. Характеристики сетей рабочих групп практически не отличаются от характеристик сетей от-

делов. Такие свойства, как простота сети и однородность, здесь проявляются в наибольшей степени, в то время как сети отделов могут приближаться в некоторых случаях к следующему по масштабу типу сетей – сетям кампусов.

Сети кампусов получили свое название от английского слова *campus* – студенческий городок. Именно на территории университетских городков часто возникала необходимость объединения нескольких мелких сетей в одну большую сеть. Сейчас это название не связывают со студенческими городками, а используют для обозначения сетей любых предприятий и организаций.

Главными особенностями сетей кампусов является то, что они объединяют множество сетей различных отделов одного предприятия в пределах отдельного здания или в пределах одной территории, покрывающей площадь в несколько квадратных километров (рис. 56). При этом глобальные соединения в сетях кампуса не используются. Службы такой сети включают взаимодействие между сетями отделов, доступ к общим базам данных предприятия, доступ к общим факс-серверам, высокоскоростным модемам и высокоскоростным принтерам. В результате сотрудники каждого отдела предприятия получают доступ к некоторым файлам и ресурсам сетей других отделов. Важной службой, предоставляемой сетями кампусов, стал доступ к корпоративным базам данных независимо от того, на каких типах компьютеров они располагаются.

Именно на уровне сети кампусов возникают проблемы интеграции неоднородного аппаратного и программного обеспечения. Типы компьютеров, сетевых операционных систем, сетевого аппаратного обеспечения могут отличаться в каждом отделе. Отсюда вытекают сложности управления сетями кампусов. Администраторы должны быть в этом случае более квалифицированными, а средства оперативного управления сетью – более совершенными.

Корпоративные сети называют также сетями масштаба предприятия, что соответствует дословному переводу термина *enterprise – wide network*. Сети масштаба предприятия (корпоративные сети) объединяют большое количество компьютеров на всех территориях отдельного предприятия. Они могут быть сложно связаны и покрывать город, регион или даже континент. Число пользователей и компьютеров может измеряться тысячами, а число серверов – сотнями, расстояния между сетями отдельных территорий могут оказаться такими, что использование глобальных связей становится необходимым (рис. 57). Для соединения удаленных локальных сетей и отдельных компьютеров в корпоративной сети применяются разнообразные телекоммуникационные средства, в том числе телефонные каналы, радиолокаторы, спутниковая связь. Корпора-

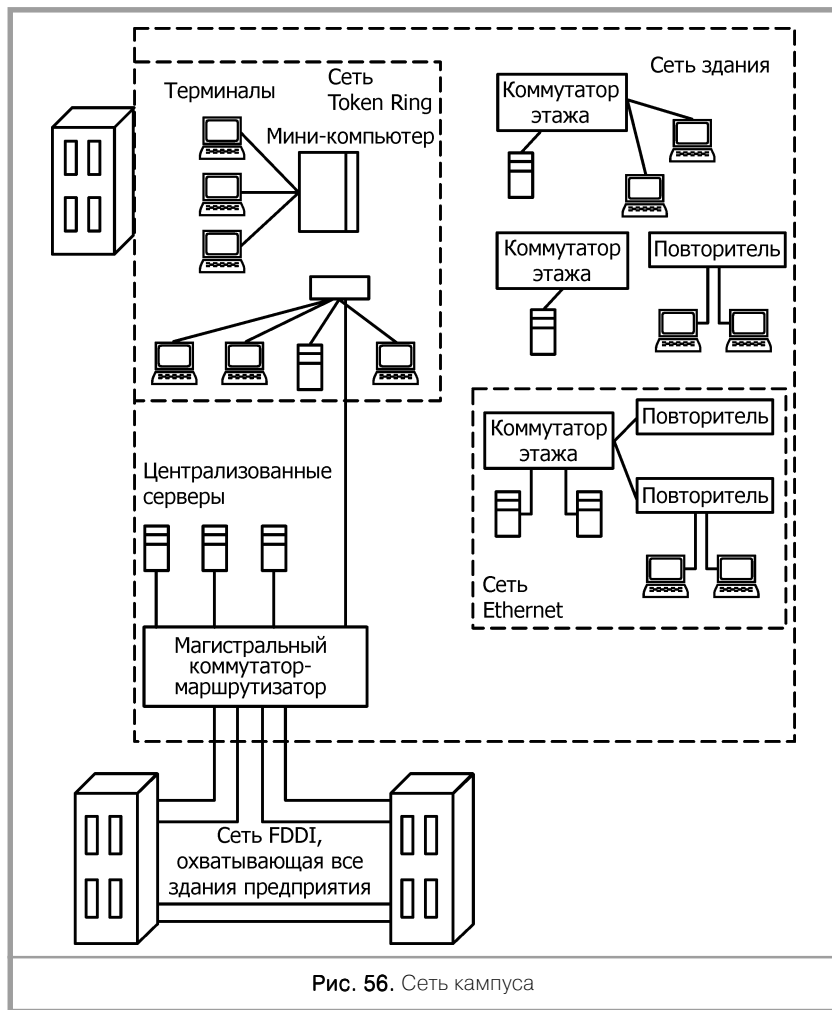


Рис. 56. Сеть кампуса

тивную сеть можно представить в виде «островков» локальных сетей, «плавающих» в телекоммуникационной среде. Непременным атрибутом такой сложной и крупномасштабной сети является высокая степень неоднородности (интерогенности) – нельзя удовлетворить потребности тысяч пользователей с помощью однотипных и аппаратных средств. В корпоративной сети обязательно используются различные типы компьютеров – от мэйнфреймов до персона-

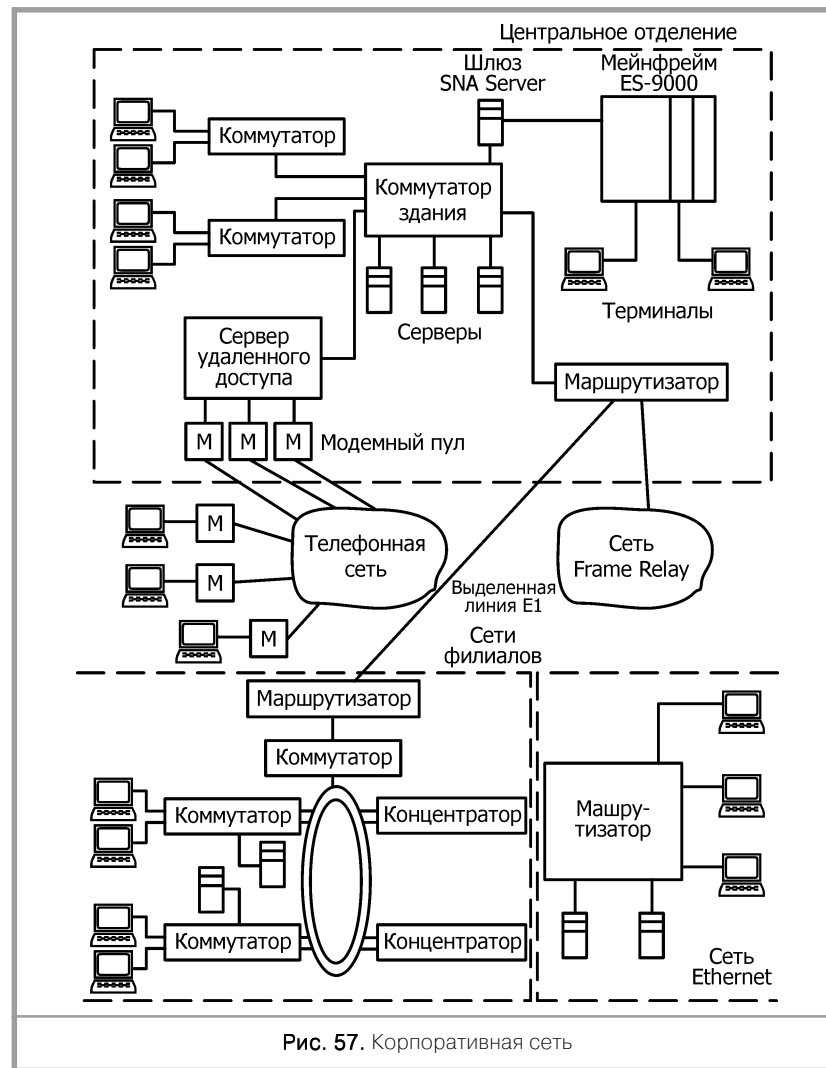


Рис. 57. Корпоративная сеть

лок, несколько типов операционных систем и множество различных приложений. Неоднородные части корпоративной сети должны работать как единое целое, предоставляя пользователям по возможности удобный и простой доступ ко всем необходимым ресурсам.

Появление корпоративной сети – это хорошая иллюстрация известного философского постулата о переходе количества в качество. При объединении отдельных сетей крупного предприятия, имеющего филиалы в разных городах и даже странах, в единую сеть многие количественные характеристики объединенной сети превосходят некоторый критический порог, за которым начинается новое качество. В этих условиях существующие методы и подходы к решению традиционных задач сетей меньших масштабов для корпоративных сетей оказались непригодными. На первый план вышли такие задачи и проблемы, которые в распределенных сетях рабочих групп, отделов и даже кампусов либо имели второстепенное значение, либо вообще не проявлялись.

В распределенных локальных сетях, состоящих из 1–20 компьютеров и примерно такого же количества пользователей, необходимые информационные данные перемещены в локальную базу данных каждого компьютера, к ресурсам которых пользователи должны иметь доступ, то есть данные извлекаются из локальной учетной базы и на их основе доступ предоставляется или не предоставляется.

Но если в сети насчитывается несколько тысяч пользователей, каждому из которых нужен доступ к нескольким десяткам серверов, то, очевидно, это решение становится крайне неэффективным, так как администратор должен повторить несколько десятков раз (по числу серверов) операцию занесения учетных данных каждого пользователя. Сам пользователь также вынужден повторять процедуру логического входа каждый раз, когда ему нужен доступ к ресурсам нового сервера. Решение этой проблемы для крупной сети – использование централизованной справочной службы, в базе данных которой хранится необходимая информация. Администратор один раз выполняет операцию занесения данных пользователя в эту базу, а пользователь один раз выполняет процедуру логического входа, причем не в отдельный сервер, а в сеть целиком. По мере увеличения масштабов сети повышаются требования к ее надежности, производительности и функциональным возможностям. По сети циркулируют все ее возрастающие объемы данных, и сеть должна обеспечить их безопасность и защищенность наряду с доступностью. Все это приводит к тому, что корпоративные сети строятся на основе наиболее мощного и разнообразного оборудования и программного обеспечения.

Конечно, корпоративным вычислительным сетям присущи и свои проблемы. Эти проблемы в основном связаны с организацией эффективного взаимодействия отдельных частей распределенной системы.

Во-первых, это сложности, связанные с программным обеспечением – операционными системами и приложениями. Программирование для распределенных систем принципиально отличается от программирования для централизованных систем. Так, сетевая операционная система, выполняя все функции по управлению локальными ресурсами компьютера, решает ее многочисленные задачи по предоставлению сетевых серверов. Разработка сетевых приложений осложняется из-за необходимости организовать совместную работу их частей, выполняющихся на разных машинах. Много забот доставляет обеспечение совместимости программного обеспечения, устанавливаемого в узлах сети.

Во-вторых, много проблем связано с транспортировкой сообщений по каналам связи между компьютерами. Основные задачи здесь – обеспечение надежности (чтобы предоставляемые данные не терялись и не искажались) и производительности (чтобы обмен данными происходил с приемлемыми задержками). В структуре общих затрат на вычислительную сеть расходы на решение «транспортных вопросов» составляют существенную часть, в то время как в централизованных системах эти проблемы полностью отсутствуют.

В-третьих, это вопросы, связанные с обеспечением безопасности, которые гораздо сложнее решаются в вычислительной сети, чем в автономно работающем компьютере. В некоторых случаях, когда безопасность особенно важна, от использования сети лучше вообще отказаться.

Однако в целом использование локальных (корпоративных сетей) дает предприятию следующие возможности:

- разделение дорогостоящих ресурсов;
- совершенствование коммутаций;
- улучшение доступа к информации;
- быстрое и качественное принятие решений;
- свобода в территориальном размещении компьютеров.

Для корпоративной сети (сети предприятий) характерны:

- масштабность – тысячи пользовательских компьютеров, сотни серверов, огромные объемы хранимых и передаваемых по линиям связи данных, множество разнообразных приложений;
- высокая степень гетерогенности (неоднородности) – типы компьютеров, коммуникационного оборудования, операционных систем и приложений различны;
- использование глобальных связей – сети филиалов соединяются с помощью телекоммуникационных средств, в том числе телефонных каналов, радиоканалов, спутниковой связи.

6.3. Программное обеспечение корпоративных сетей

Возможности компьютера как технической основы системы обработки данных в локальных сетях связаны с использованием программного обеспечения. Структура и функции программного обеспечения корпоративных сетей обусловлены тем, что эти сети основаны на технологии Интернета, сформировавшейся прежде всего вокруг протокола TCP/IP. Корпоративная сеть состоит из определенного числа взаимосвязанных компьютеров или ЛВС, использующих одну или более сетевых технологий, таких как Ethernet или Token Ring. Для управления работой сети необходима сетевая операционная система (СОС), реализующая принцип сетевой модели «клиент-сервер». Наиболее популярным СОС в этом случае являются Windows NT компании Microsoft и Net Ware компании Novell.

Программное обеспечение (*software*) – совокупность программ обработки данных и необходимых для их эксплуатации документов.

Программа (*problem, routine*) – упорядоченная последовательность команд (инструкций) компьютера для решения задач.

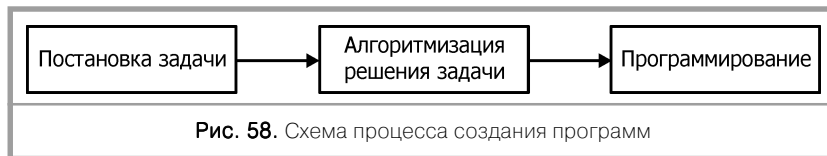
Задача (*problem, task*) – проблема, подлежащая решению.

Приложение (*application*) – программная реализация на компьютере решения задачи.

Таким образом, задачи означают проблему, подлежащую реализации с использованием средств информационных технологий, а приложение – реализованное на компьютере решение по задаче.

Приложение, являясь синонимом слова «программа», считается более удачным термином и широко используется в информатике.

Процесс создания программ можно представить как последовательность действий, представленных на рис. 58.



Постановка задачи (*problem definition*) – это такая формулировка задач на компьютере с описанием входной и выходной информации.

Постановка задачи – обобщенный термин, который означает определенность содержательной стороны обработки данных. Постановка задачи связана с конкретизацией основных параметров ее

реализации, определением источников и структуры входной и выходной информации, востребуемой пользователем.

Алгоритм – система точно сформулированных правил, определяющая процесс преобразования допустимых исходных данных (входной информации) в желаемый результат (выходную информацию) за конечное число шагов.

Алгоритм решения задачи имеет ряд обязательных свойств:

- дискретность – разбиение процесса обработки информации на более простые этапы (шаги выполнения), выполнение которых компьютером или человеком не вызывает затруднений;
- определенность – однозначность выполнения каждого отдельного шага преобразования информации;
- выполнимость – конечность действий алгоритма решения задач, позволяющая получить желаемый результат при допустимых исходных данных за конечное число шагов;
- массовость – пригодность алгоритма для решения определенного класса задач.

Программирование (*programming*) – теоретическая и практическая деятельность, связанная с созданием программ.

Программирование является собирательным понятием и может рассматриваться и как наука, и как искусство, на этом основан научно-практический подход к разработке программ.

В настоящее время бурно развивается направление, связанное с технологией создания программных продуктов.

Инструментарий технологии программирования – программные продукты поддержки (обеспечения) технологии программирования.

В рамках этих направлений сформировались следующие группы программных продуктов (рис. 59).

Средства для создания приложений, включающие:

- локальные средства, обеспечивающие выполнение отдельных работ по созданию программ;
- интегрированные среды разработчиков программ, обеспечивающие выполнение комплекса взаимосвязанных работ по созданию программ;
- CASE-технология (Computer – Aided Engineering), представляющая методы анализа, проектирования и создания программных систем и предназначенная для автоматизации процессов разработки и реализации информационных систем.

Средства для создания приложений классифицируются следующим образом:

1. Локальные средства разработки программ на рынке программных продуктов наиболее представительны и включают

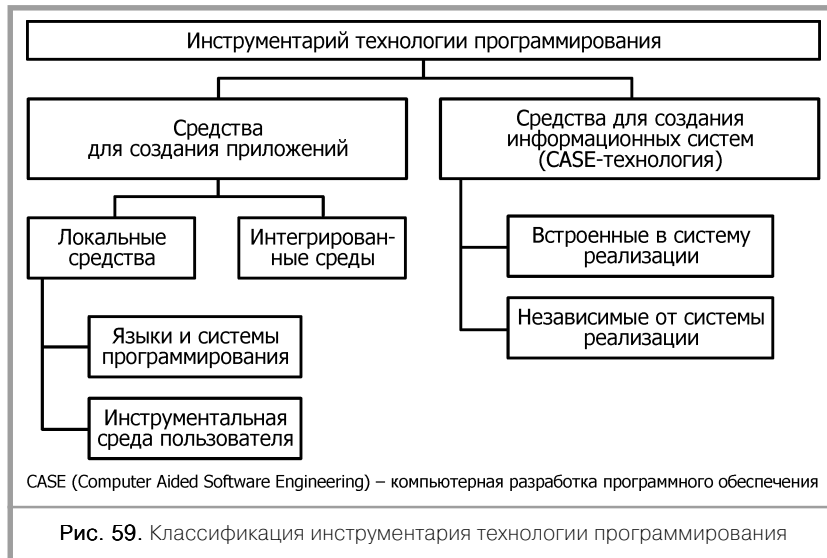


Рис. 59. Классификация инструментария технологии программирования

языки и системы программирования, а также инструментальную среду пользователя.

Язык программирования – формализованный язык для описания алгоритма решения задачи на компьютере.

2. Языки программирования подразделяются на :
- классы, согласно синтаксису образования его конструкции;
 - ориентированные на реализацию основ структурного программирования;
 - объектно-ориентированные, поддерживающие понятие объектов и их свойств и методов обработки.

Программа, подготовленная на языке программирования, проходит этап «трансляции», когда происходит преобразование «исходного кода» программы (source code) в объектный код (object code), который далее пригоден к обработке «редактором связей». Редактор связей – специальная программа, обеспечивающая построение «загрузочного модуля» (load module), пригодного к выполнению (рис. 60). Трансляция может выполняться с использованием средств «компиляторов» (compiler) или «интерпретаторов» (interpreter). Компиляторы транслируют всю программу, но без ее выполнения. Интерпретаторы, в отличие от компиляторов, обеспечивают кооперативную обработку и выполнение программы.



Рис. 60. Схема процесса создания загрузочного модуля программы

3. Инструментальная среда пользователя представлена специальными средствами, встроенными в пакеты прикладных программ (например, библиотека функций, процедур; макроккоманды; генераторы приложений и т. п.).
4. Интегрированные среды разработки программ предназначены для дальнейшего развития локальных средств разработки программ, которые объединяют набор средств для комплексного их применения на всех технологических этапах создания программ.
5. CASE-технология – программный комплекс, автоматизирующий весь технологический процесс анализа, проектирования, разработки и сопровождения сложных программных систем.

Средства CASE-технологий делятся на две группы:

- встроенные в систему реализации – все решения по проектированию и реализации привязаны к выбранной системе управления базами данных (СУБД);
- независимые от системы реализации – все решения по проектированию ориентированы на унификацию начальных этапов жизненного цикла и средств их документирования, обеспечивают большую гибкость в выборе средств реализации.

6.4. Корпоративные информационные приложения

Рассмотрим примеры некоторых корпоративных информационных приложений.

1. Приложение FLOWMARK – Flowmark application – программное обеспечение автоматизации процесса делопроизводства, предложенное корпорацией IBM. Приложение предназначено для электронных офисов и обеспечивает интеграцию значительного числа программ, вклю-

чая групповое программное обеспечение, электронную почту, обработку текстов, обработку изображений. Flowmark опирается на несколько операционных систем и построено в соответствии с архитектурой клиент-сервер. Они являются ключевой платформой в стратегии IBM по развитию группового программного обеспечения, включающего:

- средства передачи сообщений;
- среду коллективной работы;
- календарное планирование;
- управление документами.

Пользователи приложения могут взаимодействовать не только с серверами, но и с базовыми компьютерами.

2. Приложение LOTUS NOTES – Lotus Notes application – программное обеспечение распределенной обработки данных в локальной сети, предложенной обработкой данных в локальной сети, предложенное корпорацией Lotus Development для автоматизации делопроизводства.

Lotus Notes является открытой программной платформой, предназначенной для работы пользователя с распределенной информацией и разработки прикладных программ. Notes обеспечивает:

- создание и распределение любых типов данных;
- проведение телеконференций;
- управление обработкой документов и организацию документооборота;
- планирование и координацию работы сотрудников;
- контроль за проектами и договорами;
- ведение справочных баз данных;
- функционирование электронной почты.

Lotus Notes использует архитектуру клиент-сервер и погружается в значительное число сетевых операционных систем. В сети может работать до 150 серверов. Число пользователей Lotus Notes не ограничивается. Специальные программы обеспечивают взаимодействие Lotus Notes со службой глобального соединения (WWW) сети Интернет. С другой стороны, обеспечен доступ из Интернета к базам данных Lotus Notes.

3. Приложение MICROSOFT OFFICE – Microsoft Office application – комплекс прикладных программ для офиса, предложенный корпорацией Microsoft. Этот комплекс определил многосвязную стратегию электронного офиса, разработанную фирмой для нужд пользователей. Microsoft Office включает текстовый редактор, систему управления базами данных, электронную почту. База оперирует как с текстами, так и с изображениями. Выбор нужных полей в файлах происходит в соответствии

с технологией Plug-and-Play, что особенно удобно для пользователей.

Это позволяет автоматизировать создание конфигурации абонентских систем при подключении внешних устройств.

Приложение использует технологию связи и компоновки объектов (OLE), что позволяет выполнять одновременно несколько прикладных программ, передавать между ними данные и использовать общий метаязык.

Сети интранет



7.1. Назначение сети интранет

Основное назначение сети интранет (Intranet) – предоставление информационных и вычислительных ресурсов подключенным к ней пользователям. Процесс переноса служб и технологий из глобальной сети интернет (Internet) в локальные приобрел такой массовый характер, что появился даже специальный термин интранет-технологии (*intra* – внутренний).

Несмотря на то что различия между глобальными, региональными (городскими), корпоративными и локальными сетями, безусловно, существенны, все эти сети на достаточно высоком уровне абстракции имеют тождественные (подобные) структуры.

В общем виде сеть интранет состоит из следующих компонентов (рис. 61):

- сети доступа (access network);
- магистральной сети, или магистрали (core network, или backbone);
- информационных центров, или центров управления сервисами (data centers, или services control point).

Как сеть доступа, так и магистральная сеть строятся на основе коммутаторов. Каждый коммутатор оснащен некоторым количеством портов, которые соединяются с портами других коммутаторов каналами связи. Сеть доступа составляет нижний уровень иерархии вычислительной сети. К этой сети подключаются конечные (терминальные) узлы – оборудование, установленное у пользователей (абонентов, клиентов) сети. Для компьютерной сети интранет конечными узлами являются компьютеры.

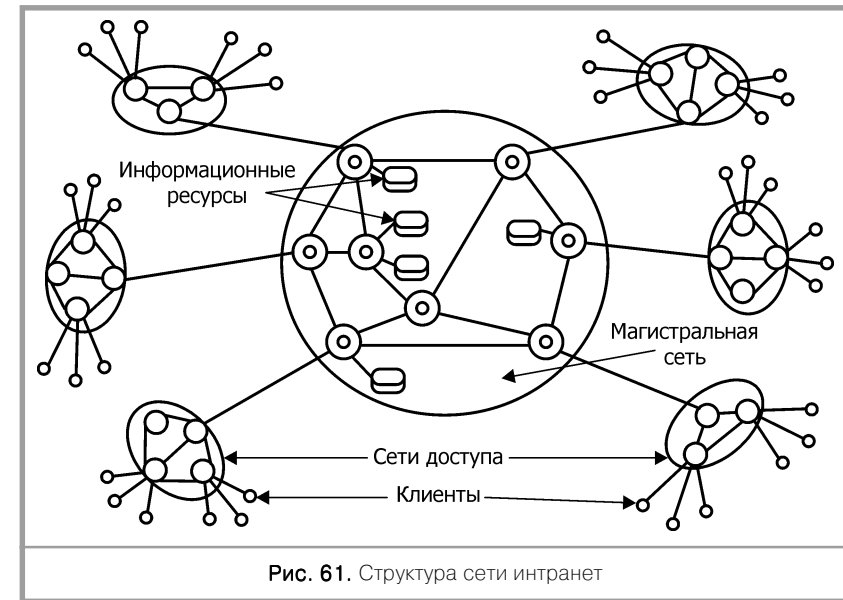


Рис. 61. Структура сети интранет

Основное назначение сети доступа – концентрация информационных потоков, поступающих по многочисленным каналам связи от оборудования пользователей в сравнительно небольшом количестве узлов магистральной сети. Сеть доступа, как и компьютерная сеть в целом, может состоять из нескольких уровней (на рис. 61 их показано два). Коммутаторы, установленные в узлах нижнего уровня, мультиплексируют информацию, поступающую по многочисленным абонентским каналам, называемым часто абонентскими окончаниями (local loop), и передают ее коммутаторам верхнего уровня, чтобы те, в свою очередь, передали ее коммутаторам магистрали.

Количество уровней сети доступа зависит от ее размеров, небольшая сеть доступа может состоять из одного уровня, а крупная – из двух-трех. Следующие уровни осуществляют дальнейшую концентрацию трафика, собирая его и мультиплексируя в более скоростные каналы.

Магистральная сеть объединяет отдельные сети доступа, выполняя функции транзита трафика между ними по высокоскоростным каналам. Коммутаторы магистрали могут оперировать не только с информационными соединениями между отдельными пользователями, но и с агрегированными информационными потоками, переносящими данные большого количества пользователей.

ских соединений. В результате информация с помощью магистрали попадает в сеть доступа получателей, демультиплексируется там и коммутируется таким образом, что на входной порт оборудования пользователя поступает только та информация, которая ему адресована.

В том случае, когда абонент-получатель подключен к тому же коммутатору доступа, что и абонент-отправитель (непосредственно и через подчиненные по иерархии связи коммутаторы), последний самостоятельно выполняет необходимую операцию коммутации.

Информационные центры, или центры управления сервисами, – это собственные информационные ресурсы сети, на основе которых осуществляется обслуживание пользователей. В таких центрах может храниться информация двух типов:

- пользовательская информация, то есть та, которая непосредственно интересует конечных пользователей сети;
- вспомогательная служебная информация, помогающая предоставлять некоторые услуги пользователям.

Примером информационных ресурсов первого типа могут служить Web-порталы, на которых расположена разнообразная справочная новостная информация, информация электронных объектов и т. п.

Ресурсами второго типа являются различные системы аутентификации и авторизации пользователей, с помощью которых организация, владеющая сетью, проверяет права пользователей на получение тех или иных услуг; системы биллинга, которые в коммерческих сетях подсчитывают плату за полученные услуги; базы данных учетной информации пользователей, на которые подписан каждый пользователь. Еще одним видом из распространенных видов вспомогательного информационного центра является централизованная система управления сетью, которая представляет собой программное обеспечение, работающее на одном или нескольких компьютерах.

Естественно, у сетей каждого конкретного типа интранет имеет много особенностей, тем не менее их структура в целом соответствует описанной выше. В то же время, в зависимости от назначения и размера сети, в ней могут отсутствовать или же иметь несущественное значение некоторые составляющие обобщенной структуры. Например, в небольшой локальной компьютерной сети нет ярко выраженных сетей доступа и магистрали – они сливаются в общую и достаточно простую структуру. В корпоративной сети, как правило, отсутствует система биллинга, так как услуги сотрудникам предприятия оказываются не на коммерческой основе.

Для конечного пользователя сеть представляет собой прежде всего набор сетевых служб, обеспечивающих ему совместное с другими пользователями сети использование сетевых ресурсов. Основные службы – файловая служба и служба печати – обычно предоставляются сетевой оперативной системой, а вспомогательные, например служба баз данных, факса или передачи голоса, – системными сетевыми приложениями. Важнейшим свойством службы является ее прозрачность, то есть интранет – открытая система, взаимодействующая с другими системами в соответствии с принятыми стандартами.

7.2. Архитектура интранет

Архитектура интранет полностью соответствует семиуровневой модели взаимодействия открытых систем (ВОС), которая является наиболее популярной сетевой архитектурной моделью.

Основное достоинство семиуровневой модели ВОС состоит в том, что изменение одного из уровней не влечет за собой необходимости внесения изменений в другие уровни, если между уровнями определены однозначно интерфейсы.

В качестве примера рассмотрим архитектуру NWAYS – Nways architecture – архитектуру скоростной передачи данных, предложенную корпорацией IBM для передачи в локальной сети больших потоков, реализующую как пакетный способ передачи (PTM), так и асинхронный способ передачи (ATM). Архитектура определяет три типа устройств, обеспечивающих передачу данных (рис. 62).

Абонентами сети являются персональные компьютеры и рабочие станции, работающие через абонентские каналы со скоростью 25 Мбит/с.

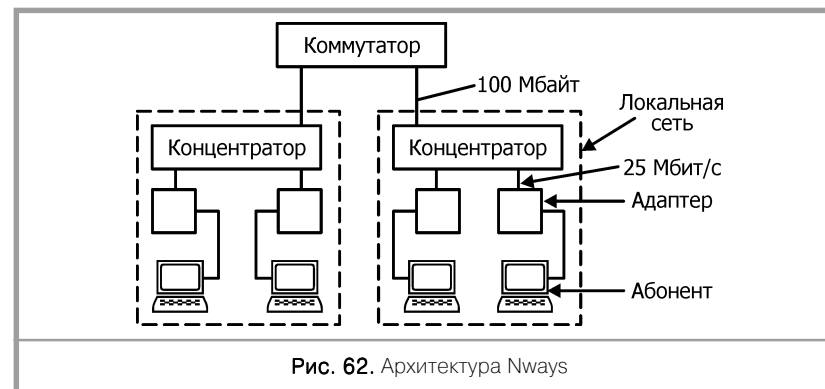


Рис. 62. Архитектура Nways

На втором уровне (ярусе) архитектуры располагаются концентраторы, каждый из которых соединяет группу абонентских каналов с одним оптическим каналом, работающим со скоростью 100 Мбит/с и образующим локальную сеть.

Наконец, на третьем ярусе (уровне) устанавливается коммутатор, связывающий оптические каналы в ассоциацию локальных сетей. Коммутаторы обеспечивают пропускную способность до десятков гигабит в секунду. Двухъярусность используемых устройств (концентратор/коммутатор) позволяет значительно снизить затраты, связанные с созданием сети, за счет уменьшения стоимости адаптеров абонентов.

В каждый концентратор загружается сервер, обеспечивающий эмуляцию локальной сети под стандарты PTM и ATM.

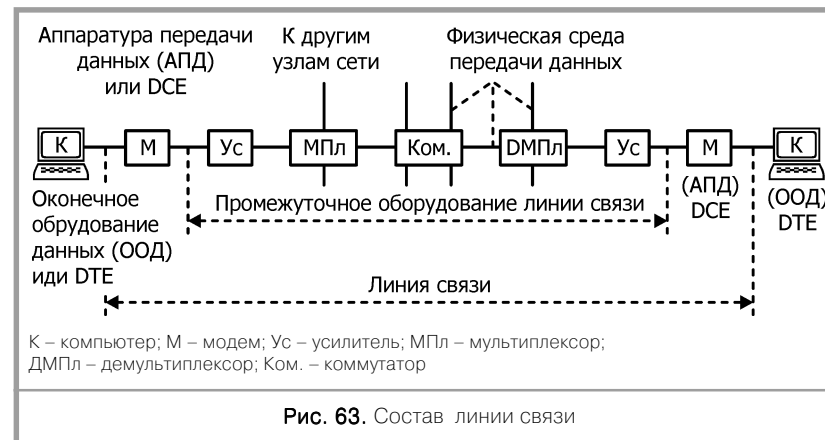
К коммутатору могут также подключаться объекты, имеющие непосредственный интерфейс PTM и ATM, например базовые компьютеры. Архитектура Nways согласуется с архитектурой сети (BNA) – Broad Network Architecture.

Следует иметь в виду, что физические каналы связи и другие элементы первичной транспортной инфраструктуры играют ключевую роль при оказании любых коммуникационных услуг, так как без них передача информации на расстояние становится просто невозможной. Фактически от наличия или отсутствия собственной транспортной инфраструктуры зависит и название предприятия, оказывающего информационно-коммуникационные услуги, – «поставщик (провайдер) услуг».

7.3. Средства построения сети интранет

При построении сетей применяются линии (каналы) связи, использующие различную физическую среду: телефонные и телеграфные провода, подвешенные в воздухе, медные коаксиальные кабели, медные витые пары, волоконно-оптические кабели, радиоволны. При выборе того или иного типа линий связи учитываются их технические характеристики, стоимость и простота монтажа. Сегодня наиболее перспективными являются волоконно-оптические кабели. На них строятся как магистрали крупных территориальных и городских сетей, так и высокоскоростные линии связи локальных сетей. Популярной средой является также витая пара, которая характеризуется отличным отношением количество/стоимость.

Линия связи состоит в общем случае из физической среды, по которой передаются информационные сигналы аппаратуры передачи данных и промежуточной аппаратуры (рис. 63). Синонимом термина «линия связи» (line) является термин «канал связи» (channel).



Физическая среда передачи данных (medium) может представлять собой кабель, то есть набор проводов, изолированных и защитных оболочек, соединительных разъемов, а также земную атмосферу или космическое пространство, через которые распространяются информационные каналы. В современных телекоммуникационных системах информация передается с помощью электрических сигналов (тока или напряжения), радиосигналов или световых сигналов – все эти физические процессы представляют собой колебания электромагнитного поля различной частоты и природы. В зависимости от среды передачи данных линии связи разделяются на:

- проводные (воздушные);
- кабельные (медные и волоконно-оптические);
- радиоканалы наземной и спутниковой связи.

Проводные (воздушные) линии связи представляют собой провода без каких-либо изолирующих или экранирующих оплеток, проложенные между столбами или висящие в воздухе. По таким линиям связи традиционно передаются телефонные или телеграфные сигналы, но при отсутствии других возможностей эти линии используются и для передачи компьютерных данных. Скоростные качества и помехозащищенность этих линий оставляют желать лучшего. Сегодня проводные линии связи быстро вытесняются кабельными.

Кабельные линии имеют достаточно сложную конструкцию. Кабель состоит из внутренних проводников, заключенных в несколько слоев изоляции: электрической, электромагнитной, механической, а также, возможно, климатической. В компьютерных сетях применяются три основных типа кабеля: кабели на основе

скрученных пар медных проводов, коаксиальные кабели с медной жилой, а также волоконно-оптические кабели.

Радиоканалы наземной и спутниковой связи образуются с помощью передатчика и приемника радиоволн. Существует большое количество и разнообразие типов радиоканалов, отличающихся по частотным диапазонам и дальностью радиосвязи. Диапазон коротких (КВ), средних (СВ) и длинных (ДВ) волн, называемый также диапазоном амплитудной модуляции (Amplitude Modulation, AM) по типу используемого в нем метода модуляции сигнала, обеспечивает дальнюю связь, но при невысокой скорости передачи данных. Более скоростными являются каналы, работающие на диапазонах ультракоротких волн (УКВ), для некоторых характерна частотная модуляция (Frequency Modulation, FM), а также диапазонах сверхвысоких частот (СВЧ, или microwaves). В этих диапазонах радиосигналы не отражаются от ионосферы Земли и для устойчивой связи требуется наличие прямой видимости между передатчиком и приемником с учетом кривизны поверхности земли. Поэтому такие частоты используют либо спутниковые каналы связи, либо радиорелейные каналы связи, где условия прямой видимости выполняются.

Аппаратура передачи данных, или **АДП** (Data Circuit Termination Equipment, DCE), в компьютерных сетях непосредственно присоединяет компьютеры или локальные сети пользователя к линии связи и является, таким образом, пограничным оборудованием. Традиционно аппаратуру передачи данных включают в состав линии связи. Примерами DCE являются модемы, терминальные адаптеры сетей ISDN (Integrated Services Digital Network) – цифровая сеть с интегральным сервером, устройства подключения к цифровым каналам. Обычно DCE работает на физическом уровне, отвечая за передачу информации в физическую среду (в линию) и прием из нее сигналов нужной формы и мощности.

Оконечное оборудование данных, или **ООД** (Data Terminal Equipment, DTE), вырабатывает данные для передачи по линии связи и подключается непосредственно к аппаратуре передачи данных в компьютерных сетях. Примерами DTE могут служить компьютеры, коммутаторы или маршрутизаторы. Эту аппаратуру не включают в состав линии связи.

Разделение оборудования на классы DCE и DTE в локальных сетях является достаточно условным. Например, адаптер локальной сети можно считать как принадлежностью компьютера, то есть DTE, так и составной частью канала связи, то есть DCE. В глобальной сети это разделение становится более четким, при этом устройство DCE обычно относят к глобальной сети, а устройство DTE – уже нет.

Промежуточная аппаратура чаще всего используется на линиях связи большой протяженности. Она решает две основные задачи:

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

В локальных сетях промежуточная аппаратура может совсем не использоваться, если протяженность физической среды – кабелей и радиоэфира – 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 расстояние в сотни и тысячи километров. Поэтому без усилителей (повышающих мощность сигнала) и регенераторов (наряду с повышением мощности восстанавливающих форму импульсных сигналов, исказившихся при передаче на большие расстояния), установленных через определенные расстояния, построить территориальную линию связи невозможно. В глобальной сети необходима также и промежуточная аппаратура другого рода – мультиплексоры, демультимплексоры и коммутаторы. Эта аппаратура решает вторую указанную задачу, то есть создает между двумя абонентами сети непрерывный составной канал из отрезков физической среды – кабелей с усилителями. Эти отрезки с широкой полосой пропускания сигналов одновременно участвуют в образовании сразу нескольких составных каналов. Такой высокоскоростной канал, по которому передаются одновременно данные от большого числа сравнительно низкоскоростных абонентских линий, обычно называют уплотненным каналом. Наличие промежуточной коммутационной аппаратуры избавляет создателей глобальной сети от необходимости прокладывать отдельную кабельную линию связи для каждой пары соединяемых узлов сети.

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

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

В **цифровых линиях** связи передаваемые сигналы имеют конечное число состояний. С помощью таких сигналов передаются как компьютерные данные, так и в цифровой форме речь и изображение.

В цифровых каналах связи используется специальная промежуточная аппаратура – регенераторы, которые улучшают форму импульсов и обеспечивают их синхронизацию, то есть восстанавливают период их следования. Промежуточная аппаратура мультиплексирования и коммутации первичных сетей работает по принципу временного мультиплексирования каналов (Time Division Multiplexing, TDM), когда каждому низкоскоростному каналу выделяется определенная доля времени (тайм-слот или квант) высокоскоростного канала.

В настоящее время аналоговые каналы стали применяться в первичных сетях нового типа, использующих метод мультиплексирования по длине световой волны (Wavelength Division Multiplexing, WDM). В первичных сетях WDM каждый канал передает свою информацию с помощью световой волны определенной длины и, соответственно, частоты (спектра). Такой канал также называется спектральным каналом, так как ему выделяется определенная полоса спектра светового излучения.

Для установления интранет необходимы следующие компоненты:

- компьютерная сеть для совместного использования ресурсов или сеть взаимосвязанных ЛВС и УПК (удаленных персональных компьютеров);
- программное обеспечение сервера, поддерживающее запросы браузеров в формате протокола гипертекстовых сообщений HTTP (Hyper Text Transfer Protocol – служба глобального соединения: «www» – World Wide Web);
- компьютер-сервер, который может работать как сервер Инtranета;
- сетевая операционная система, поддерживающая протокол TCP/IP, – Unix, Windows NT или Windows 2000, NetWare, OS/2;
- компьютеры-клиенты, на которых имеется сетевое программное обеспечение, позволяющее посылать и принимать пакетные данные по протоколу TCP/IP;
- программное обеспечение браузера для различных компьютеров-клиентов (Netscape Navigator, Microsoft Internet Explorer).

Эти требования к оборудованию и программному обеспечению интранет дополняются требованиями к знанию технологии составления документов на языке написания гипертекста HTML (HyperText Markup Language – гипертекстовый высокоуровневый язык).

Для удобства редактирования объектов используется браузер объектов (Browser) – интегрированный отладчик, позволяющий

выполнять пошаговую трассировку кода, задавать точки остановок (Break points).

7.4. Обзор языков программирования в интранет

Эффективность использования интранет (КВС) зависит от успешного решения как технологических, так и организационных вопросов в аспекте программного обеспечения и языков программирования.

Для поддержки информационных технологий в сетях интранет выделяют три класса программных продуктов (рис. 64):

- системное программное обеспечение;
- пакеты прикладных программ;
- инструментарий технологии программирования.

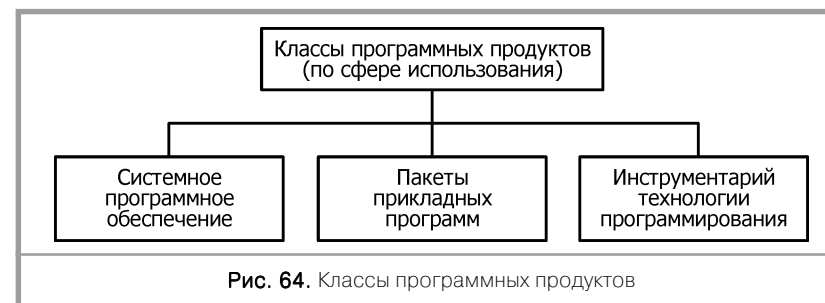


Рис. 64. Классы программных продуктов

Структура системного программирования состоит из базового программного обеспечения и сервисного программного обеспечения.

В базовое программное обеспечение входят:

- операционная система;
- операционные оболочки (текстовые и графические);
- сетевая операционная система.

Язык программирования – формализованный язык для описания алгоритма решения задачи на компьютере.

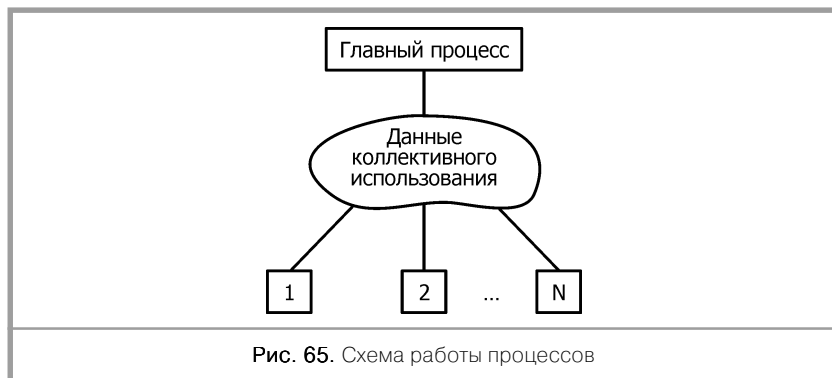
Разработано и создается большое число различных языков. Например:

Язык АССЕМБЛЕРА – assembly language – язык, понятия которого отражают архитектуру компьютера. Он содержит предложения, форматы команд и данных, которые однозначно соответствуют

возможностям конкретного компьютера. Рассматриваемый язык является системой обозначений, используемой для представления программ, записанных в коде компьютера.

Язык описания технических средств (HDL) – Hard-Ware Description language – язык моделирования, разработки и тестирования устройств, предназначенных для обработки дискретных сигналов. Он позволяет разработчикам создавать иерархические структуры, реализовывать разнообразные функции, выполнять сложные арифметические операции и логику сравнения.

Язык параллельной обработки (parallel processing language) – язык, обеспечивающий описание процесса при параллельной обработке данных. Он основывается на модели, в которой прикладной процесс делится на главный и группу рабочих процессов (рис. 65). Главный процесс разделяет данные коллективного использования, компонует их в кортежи и выдает задания. Рабочие процессы решают порученные им задания. При этом обеспечивается синхронизация всех этих процессов. Одним из языков, осуществляющим распараллеливание данных, является язык Linda.



Языки программирования:

- машинные языки (computer language) – языки программирования, воспринимаемые аппаратной частью компьютера (машинные коды);
- машинно-ориентированные языки (computer-oriented language) – языки программирования, которые отражают структуру конкретного типа компьютера (ассемблера);
- алгоритмические языки (algorithmic language) – не зависящие от архитектуры компьютера языки программирования для отражения структуры алгоритма (Паскаль, Фортран, Бейсик и др.);

- процедурно-ориентированные языки (procedure-oriented language) – языки программирования, где имеется возможность описания программ как совокупности процедур (подпрограмм);
- проблемно-ориентированные языки (universal programming language) – языки программирования, предназначенные для решения задач определенного типа (Лиск, РПГ, Симула и др.);
- интегрированные системы программирования.

Язык программирования обеспечивает:

- поиск, обработку и передачу данных;
- определение функций, выявляемых устройством;
- описание процессов, в том числе в понятиях логики.

Выделяют несколько разновидностей языков программирования. К первой из них относятся «языки высокого уровня». Это языки, которые отображают естественные для человека понятия, не связанные с конкретной структурой какой-нибудь системы. К ним в первую очередь относятся универсальные: язык Ada, язык «С», язык Pascal и простой язык Basic. Кроме того, существуют языки, предназначенные для определения областей их использования:

- язык ALGOL для математических задач;
- язык Chill для телекоммуникационных систем;
- язык COBOL для экономических задач;
- язык PostScript для описания изображений;
- язык SGML для описания документов.

Среди языков высокого уровня особо выделяют языки параллельного программирования, например язык Linda. Языки высокого уровня отражают потребности программиста, а не возможности системы обработки данных.

Вторую разновидность языков программирования образуют «языки низкого уровня», например язык ассемблера.

В третью группу входят «машинные языки» – языки, элементами которых являются команды компьютеров. В этих языках управление и структура данных непосредственно отображают архитектуру конкретного компьютера. Программы, разработанные на языках высокого уровня, необходимо трансформировать в какой-нибудь язык низкого уровня. Эту операцию выполняют трансляторы.

Широкое распространение получили языки описания процессов искусственного интеллекта. Они ориентированы на рабочие станции, оснащенные мощными средствами компьютерной графики. Первым шагом в этом направлении было появление «языка обработки информации» (IPL). Этот язык позволил программам обрабатывать не числа, а понятия, став первым средством имитации функ-

ции мышления. Позже появились язык LISP, предназначенный для описания задач символьного представления и обработки произвольных объектов, и язык PROLOG, обеспечивающий обработку списков.

Язык описания языков программирования называют «метаязыком», например VDL, используемый в методе VDM – Vienna Development Method – метод из Вены, предназначенный для разработки программ. Он охватывает все фазы жизненного цикла программ: анализ, спецификацию требований, проектирование, реализацию, тестирование.

Остановимся на рассмотрении языков.

Язык ADA – Ada language – универсальный язык программирования, назван в честь Августа Ады Лавлейс, одного из первых в истории программистов. Он используется для разработки программно-обеспечения встроенных и управляющих комплексов (например, аэронавигационных). Язык основан на идеях структурного программирования и обеспечивает создание сложномодульных систем, характеризуется высокой степенью независимости от операционных систем. В нем обеспечивается поддержка средств параллельной обработки данных в реальном времени.

Язык ALGOL – ALGOL language – язык программирования для решения математических задач.

Он в первую очередь предназначен для решения численных задач. Точное определение синтаксиса языка позволило сделать его независимым от конкретного типа и структуры компьютера. Особенностями языка стала блочная структура. Наибольшее распространение ALGOL получил в Европе и стал важной вехой на пути новых языков, например языка Pascal.

Язык BASIC – BASIC language – простой язык программирования. Он вначале был разработан в Дартмутском колледже для использования студентами в системах с разделением времени. Язык оказался простым и удобным, получил большую популярность при работе пользователей в режиме диалога с небольшими системами. В него включены средства редактирования, избавляющие пользователя от изучения сложной операционной системы. Он снабжен всем необходимым для разработки программ – синтаксисом языка, поддержкой структурного программирования, средствами использования процедур компоновки многомодульных программ, отладки и подсказки программистам.

Язык «С» – «C» language – универсальный язык программирования, разработанный первоначально для операционной системы UNIX – UNIX operation system, созданной корпорацией Bell Laboratory. Простота, эффективность и переносимость на другие операционные системы сделали этот язык одним из наиболее распрост-

раненных. В языке «С» удачно сочетаются особенности современных языков высокого уровня и возможность адресации технических средств компьютера, ассоциируемой с языком ассемблера. Язык «С» имеет удобный синтаксис, привлекающий профессиональных программистов. Рассматриваемый язык утвержден Международной организацией по стандартизации (ISO) – International Organization for Standardization. Использование объектно-ориентированной архитектуры привело к появлению расширенного языка С++.

Язык COBOL – COBOL language – язык, предназначенный для решения экономических задач. Он получил распространение в области обработки коммерческой информации. Отличается развитыми средствами работы с файлами и формой записи, приближенной к английскому языку. На языке COBOL хорошо описываются данные со структурой, типичной для деловых документов. На нем можно излагать задачи без особой предварительной подготовки. Корпорация IBM разработала COBOL для операционной системы OS/2. В этой версии язык, широко распространенный среди пользователей крупных компаний, опирается на объектно-ориентированную архитектуру.

Язык FORTRAN – FORTRAN language – язык, предназначенный в первую очередь для научных расчетов. Он описывает преобразование целых действительных чисел. Они определяются как переменные величины, обозначаемые символами. В языке легко выполняются арифметические операции, задачи с разветвляемой логикой, составлением форм ведомостей, экономические расчеты.

Запись фортрановских программ сильно напоминает алгебраические уравнения. Информационной базой языка служит массив, соответствующий матрицам, применяемым в расчетах. Имеется возможность создания библиотек научных программ. В 1991 г. вышла версия языка FORTRAN 90. В нее добавлены новые глобальные декларации, операции над массивами, имена встроенных программ. Идентификация осуществляется не шестью, как ранее, а 31 символом.

Язык LINDA – Linda language – язык, предназначенный для параллельной обработки данных. При использовании этого языка прикладной процесс делится на группу процессов, в среде которых выделяется главный. Указанные процессы осуществляются одновременно и синхронизированы один относительно другого. Язык предоставляет программистам эффективные средства создания параллельных программ.

Ядром языка является пространство кортежей – совокупность взаимосвязанных данных. Каждая из них содержит ключевое слово для выборки кортежа.

Язык Linda позволяет писать параллельные программы на обычных: языке «С», языке FORTRAN, языке LIPS, дополняя их операциями организации межпроцессорного обмена данными. Важной характеристикой языка Linda является то, что он может использоваться для написания программ, выполняемых на суперкомпьютерах с различной архитектурой.

Язык LIPS – LIPS language – язык обработки списков, основанный на понятии списка. Это универсальный, высокоуровневый язык, все объекты которого (программы и данные) рассматриваются как списки. Обеспечивается работа пользователя с системой в режиме диалога. Язык широко используется для программирования задач искусственного интеллекта.

В отличие от числовой обработки, представление данных в виде списков позволяет выражать мысли и понятия. LIPS может быть также использован для программирования прикладных процессов, не связанных с искусственным интеллектом, например с обычными расчетами. На основе LIPS разработаны средства программирования с удобными интерфейсами пользователя. Он позволяет также работать со структурами данных, размеры и состав которых видоизменяются в ходе выполнения прикладных процессов.

Язык PASCAL – Pascal language – язык структурного программирования, вначале использовался для обучения программированию. Но затем получил широкое распространение для составления разнообразных программ. Язык простой, но позволяет обрабатывать разнообразные типы данных: массивы, записи, файлы. Он легко реализуется на компьютерах разных типов. Версия Pascal-Plus применяется при параллельной обработке данных.

Язык PROLOG – PROLOG language – язык программирования, используемый в задачах искусственного интеллекта, программа которого представляет собой совокупность логических утверждений и правил. Утверждения состоят из условий (предикатов), связей, констант, образуя базу данных. Правила имеют вид: «А, если В и D». Основным элементом языка является так называемый «атом», который выражает отношения между отдельными объектами.

PROLOG – это новый подход к использованию формальной логики. Пользуясь этим языком, программист имеет дело непосредственно с логическими связями между понятиями. Для этого в языке определяются правила, которые могут выполняться для достижения целей, поставленных перед прикладным процессом.

Язык SGML – SGML language – язык логического описания структуры документов. Он оперирует с разнообразными объектами: текстами, изображениями, фрагментами звука – и предоставляет пользователям общие языковые средства описания этих компонентов документов. Благодаря этому стандартизируются форма и логиче-

ская структура представления документов, их редактирование и размещение в базах данных.

Язык состоит из трех частей:

- описания, дающего заголовок файлу, содержащему информацию о системе, в которой будет использоваться документ;
- определения типа документа, задающего его структуру, категории лиц, имеющих к нему доступ, иерархию объектов;
- текстового содержания документа.

Чтобы воспользоваться SGML-документом, необходима программа, преобразующая его в формат, удобный для поиска, редактирования, включения в прикладные программы или направления на принтер.

7.5. Интегрированный язык HTML в интранет

Создан гипертекстовый высокоуровневый язык (HTML) – Hyper Text Markup Language, – который органично введен в язык SGML и предназначен для работы со Службой глобального соединения (WWW). При использовании HTML можно не только форматировать документы, но и обеспечивать связь с другими документами, расположенными в различных абонентских системах информационной сети.

Гипертекст – hypertext – текст, представленный в виде ассоциативно связанных блоков, содержащий в себе связи с другими текстами, графической и иной информацией.

Модель гипертекста определяется графом, в вершинах которого располагаются части (блоки) текста, содержащие какие-либо сведения, предложения обобщения. Эти блоки, например статья «сеть» (рис. 66), взятая из справочной базы знаний, имеют смысловую связь, фиксируемую ребрами графа. Благодаря этому гипертекст значительно отличается от обычного текста. Обычные, «линейные тексты» имеют пользовательскую структуру и предусматривают (в христианском мире) их чтение слева направо и сверху вниз. Между тем использование гипертекста позволяет фиксировать отдельные мысли, факты, сведения, а затем связывать их друг с другом, двигаясь в любых направлениях, определяемых ассоциативными связями. В результате образуется «нелинейный текст».

Создается гипертекст в **три этапа**:

- идеи (их сбор);
- план (связь идей);
- реализация ветвящейся структуры гипертекста.

В свою очередь, созданный гипертекст может развиваться и далее, обеспечивая основу для базы знаний и возможность выполне-



ния последующих циклов его маршрутизации. В тех случаях, когда к блокам текста добавляются большое число изображений и записи звука, гипертекст превращается в «**гиперсреду**».

В корпоративных сетях используется язык описания гипертекстовых документов HTML, который, не будучи языком программирования, представляет собой мощное средство обработки документов. Для создания HTML-документов необходим текстовый редактор, а для их просмотра – браузер. Пользуясь HTML, следует включать в свой документ специальные символы – «теги» (коды), которые предоставляют браузеру определенную информацию для вывода содержимого документа на экран. Каждый HTML-документ имеет две части: головную, содержащую заголовок документа, и тело, состоящее из содержимого документа. Язык HTML обеспечивает связь документов ссылками, причем есть возможность создавать ссылки на различные секции того же или других документов, что обеспечивает пользователям более быстрый доступ необходимой им информации.

Эффективность функционирования корпоративной сети во многом определяется возможностями пользователей взаимодействовать с их Web-страницами. Среди различных методов расширения интерактивных возможностей корпоративных сетей, создания интерактивных Web-станций в настоящее время чаще всего используются CGI-«сценарии» (Common Gateway Interface – интерфейс общего шлюза). CGI-сценарий представляет собой программу, которая осуществляет связь с Web-сервером для обработки и предоставления данных. Обычно он применяется в узлах для создания интерактивных HTML-форм (бланков) заполняемых пользователями, которые затем передаются на сервер для обработки. При исполнении

CGI-сценария пользователь взаимодействует с браузером при заполнении формы, после чего браузер должен взаимодействовать с сервером для обработки содержимого формы. Сервер, в свою очередь, выполняет набор запрограммированных команд (сценарий) обработки содержимого формы.

HTML-форма аналогична стандартному HTML-документу с добавлением тегов <FORM> и </FORM> и связи CGI-сценарием. Для разработки разнообразных интерактивных HTML-форм можно использовать набор стандартных CGI-сценариев.

7.6. Связь интранет с Интернетом

Как известно, структура Интернета представляет собой глобальную компьютерную сеть, и само ее название означает «между сетей», то есть соединяющая отдельные сети.

Логическая структура Интернета представляет собой некоторое виртуальное объединение, имеющее свое собственное информационное пространство. Она обеспечивает обмен информацией между всеми компьютерами, которые входят в сети, подключенные к ней. Тип компьютера и используемая система значения не имеют. Соединение сетей обладает громадными возможностями и предоставляет в распоряжение своих пользователей множество всевозможных ресурсов.

Основные ячейки Интернета – локальные вычислительные сети. Это значит, что Интернет не просто устанавливает связь между отдельными компьютерами, а создает пути соединения для более крупных единиц – групп компьютеров. Если некоторая локальная сеть непосредственно подключена к Интернету, то каждая рабочая станция этой сети также может подключаться к Интернету. Существуют также компьютеры, самостоятельно подключенные к Интернету, которые называются хост-компьютеры (*host* – хозяин). Каждый подключенный к сети компьютер имеет свой адрес, по которому его может найти абонент из любой точки света.

Схема подключения локальной сети к Интернету приведена на рис. 67. Поскольку в Интернете все компьютеры, подключенные к сети, равноправны, то, объединяя различные сети, система не создает при этом никакой иерархии (рис. 68).

Интернет самостоятельно осуществляет передачу данных. К адресам станций предъявляются специальные требования. Адрес должен иметь формат, позволяющий вести его обработку автоматически, и должен нести некоторую информацию о своем владельце.

С этой целью для каждого компьютера устанавливаются **два адреса: цифровой IP-адрес** (IP – Internet-work Protocol – межсетевой протокол) и **доменный адрес**. Оба эти адреса могут применяться

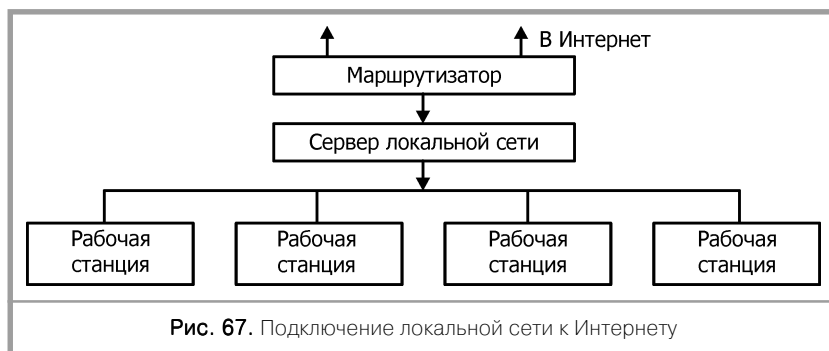


Рис. 67. Подключение локальной сети к Интернету

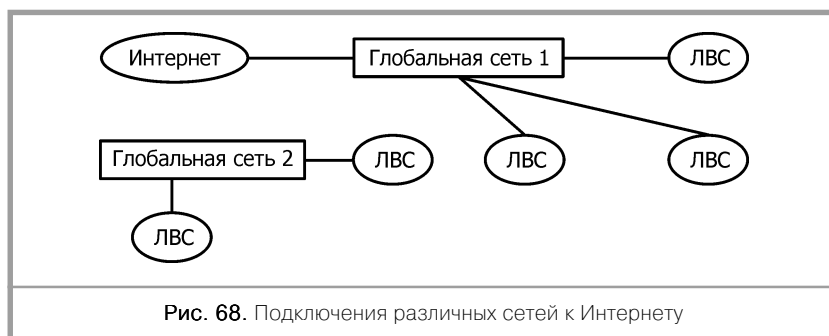


Рис. 68. Подключения различных сетей к Интернету

равноценно. **Цифровой адрес** удобен для обработки на компьютере, а доменный адрес – для восприятия пользователем. Цифровой адрес имеет длину 32 бита. Для удобства он разделен на четыре блока по 8 бит, которые можно записать в десятичном виде. Адрес содержит полную информацию для идентификации компьютера. Два блока определяют адрес сети, а два других – адрес компьютера внутри этой сети. Существует определенное правило для установления границы между этими адресами. Поэтому IP-адрес включает в себя три компонента:

- адрес сети;
- адрес подсети;
- адрес компьютера в подсети.

Доменный адрес определяет область, представляющую ряд хост-компьютеров. В отличие от цифрового адреса, он читается в обратном порядке. Вначале идет имя компьютера, затем имя сети, в которой он находится. В системе адресов Интернета приняты

длины, представленные географическими регионами, имеющие имя, состоящее из двух букв (например, США – US, Россия – ru, Франция – fr, Канада – ca и т. д.).

Существуют и домены, разделенные по тематическим признакам. Они имеют и трехбуквенное сокращенное название (например, коммерческая организация – com, учебное заведение – edu и т. д.).

Компьютерное имя включает как минимум два уровня доменов. Каждый уровень отличается от другого точкой. Слева от домена верхнего уровня располагаются другие имена. Все имена, находящиеся слева, – поддомены для общего домена. Например, существует имя «tutor.sptu.edu». Здесь «edu» – общий домен для школ и университетов, «tutor» – поддомен «sptu», который является поддоменом «edu».

Для пользователей Интернета адресами могут быть просто их регистрационные имена на компьютере, подключенном к сети. За именем следует знак «@». Все слева присоединяется к имени компьютера. Например, пользователь, зарегистрировавшийся под именем «Victor» на компьютере, имеющем в Интернете имя «tutor.sptu.edu», будет иметь адрес «Victor@tutor.sptu.edu».

В Интернете могут использоваться не только имена отдельных людей, но и имена групп.

Для обработки пути поиска в доменах имеются специальные серверы имен, которые преобразуют доменное имя в соответствующий цифровой адрес.

Локальный сервер передает запрос на глобальный сервер, имеющий связь с другими локальными серверами имен. Для выхода в Интернет пользователь должен знать адрес домена, с которым необходимо установить связь.

Согласование глобальных сетей между собой, а также с локальными сетями осуществляется в основном на сетевом и транспортном уровнях.

В настоящее время существуют два подхода к обеспечению межсетевого взаимодействия:

- объединение сетей в рамках сети Интернет в соответствии с межсетевым протоколом IP;
- объединение сетей коммутации пакетов (X.25) в соответствии с рекомендациями МККТТХ.25 – международный консультативный комитет по телефонии и телеграфии.

Основное различие в этих подходах заключается в следующем: протокол IP относится к протоколам без установления логического соединения (дейтаграммным), а рекомендация X.75 предполагает организацию логического соединения (канала).

7.7. Услуги глобальной сети Интернет

Основные возможности интернет-технологий:

- создание Web-страниц на языках HTML, VRML;
- распространение через сеть справочной информации, включая руководства и полные программы через Web;
- разработка сценариев, позволяющих создавать анимационные Web-страницы;
- создание мультимедийных приложений внутри Web-страниц;
- проведение видео- и других типов конференций в реальном масштабе времени и др.

Интернет также обеспечивает возможность выполнения ряда других функций, а именно:

- электронная почта, позволяющая отправлять и принимать сообщения по конкретному адресу. Для общения с почтовым сервером программ клиент использует протокол SMTP (Simple Mail Transfer Protocol) для отправки почты, а POP3 (Post Office Protocol Version 3) – для получения;
- программа чтения новостей VSENet, помогающая определить нужную конференцию и регулярно получать обновляемую информацию по выбранным темам;
- передача и прием файлов FTP (File Transfer Protocol) – стандартный способ передачи файлов в Интернете;
- прослушивание звука в Web по мере того, как аудиофайлы присылаются на компьютер. Такую возможность реализует программа Read Audio Player. С ее помощью можно прослушивать радио, музыку и различные звуковые сообщения, не ожидая окончания загрузки аудиофайла;
- просмотр документов в PDF-формате. Программа Adobe Acrobat позволяет просматривать документы правительственных и коммерческих организаций, распространяемых в Web в форме бумажных документов;
- создание объектов ActiveX. Это небольшие программы, которые могут вставляться в Web-страницы. При просмотре такой страницы Web-браузер пытается загрузить и запустить любые программы, находящиеся на ней;
- аналог телефонного разговора IRC (Internet Relay Chat) позволяет организовать беседу с помощью клавиатуры;
- создание гипертекстовых документов Word Wide Web. Язык HTML (Hyper Text Markup Language), определяющий, как выводятся на экран текст и графика, какие используются шрифты, какие поля ввода данных изображены и как определяются свя-

зи с другими Web-страницами. Для передачи гипертекста используется протокол HTTP (Hypertext Transfer Protocol). Каждая Web-страница имеет свой адрес, называемый URL (Uniform Resource Locator) – универсальный локатор ресурса. Следующим шагом является язык моделирования виртуальной реальности VRML (Virtual Reality Modeling Language), позволяющий моделировать в Web трехмерные объекты.

Для входа в сеть Интернет необходима специальная программа просмотра Web-страниц – Web-browser. Наиболее распространены программы просмотра Microsoft Internet Explorer и Netscape Navigator.

В настоящее время Интернет в действительности превратился в многопротокольную сеть, интегрирующую другие стандарты. Основные среди них – протоколы эталонной модели взаимодействия открытых систем, предложенные Международной организацией по стандартизации. В первую очередь это касается протоколов сетей коммутации пакетов (X.25). Хотя X.25 и TCP/IP – разные протоколы, идея, лежащая в их основе, одна и та же: данные транспортируются к месту своего назначения в виде пакетов с четко указанными адресами.

Как было сказано ранее, протокол TCP/IP состоит из двух протоколов: TCP и IP. Протокол TCP является стандартным транспортным протоколом и представляет сервис для надежной передачи информации между клиентами сети. В свою очередь, протокол IP обеспечивает сервис доставки пакетов между узлами сети Интернет. Протокол IP отвечает за адресацию сетевых узлов. В процессе функционирования протокол IP постоянно взаимодействует с протоколом межсетевых управляющих сообщений ICMP (Internet Control Message Protocol), образуя с ним так называемый «межсетевой модуль» (IP-модуль). Естественно, что IP-модуль должен быть реализован как в шлюзах, так и в подключаемых к ним абонентских системах (рис. 69).

При передаче информации из абонентской системы одной подсети в абонентскую систему другой подсети, например из абонентской системы 1 в абонентскую систему 2, осуществляются следующие процедуры (рис. 69).

Абонентская система – отправитель – средствами транспортного уровня делит полученный блок данных на сегменты, добавляя к каждому из них заголовок и порядковый номер. Сформированные таким образом дейтаграммы поступают на IP-модуль, где они упаковываются модулем сетевых протоколов подсети «Ф». Состав модуля сетевых протоколов определяется типом используемой подсети. Например, для подсети Ethernet такими протоколами являются IEEE 802.2 и 802.3. Модуль сетевых протоколов формирует соответ-

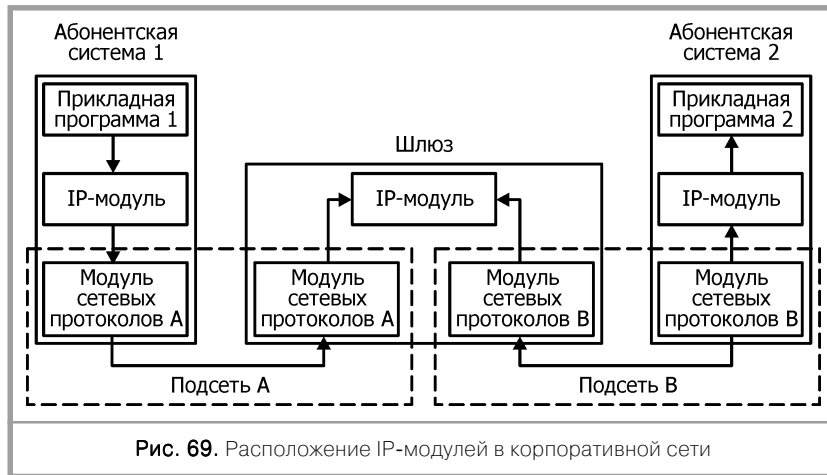


Рис. 69. Расположение IP-модулей в корпоративной сети

ствующий кадр данных, помещая в него IP-пакет. В соответствии с протоколом канального уровня подсети кадр передается шлюзу, подключенному к абонентской системе 1. Получив кадр данных, шлюз с помощью модуля сетевых протоколов извлекает из него IP-пакет, который затем обрабатывается IP-модулем. Затем с помощью модуля сетевых протоколов формируется новый кадр данных, при этом определяется адрес следующего шлюза или, как в данном случае, адрес получателя. Формат кадра данных и содержимое его заголовка определяются протоколами канального уровня подсети «В» и, естественно, могут отличаться от исходных. Достигнув адреса, кадр данных последовательно распаковывается, а именно из него извлекаются IP-пакет, дейтаграмма и исходный блок данных.

7.8. Протоколы модуля сетевого управления

Универсальность данной схемы взаимодействия систем заключается в том, что одноименные модули в различных системах взаимодействуют между собой как бы напрямую, аналогично взаимодействию протоколов в рамках эталонной модели взаимодействия открытых систем. Таким образом, различие в протоколах модуля сетевого управления не сказывается на IP-протоколах.

Протоколы TCP и IP располагаются в середине эталонной модели взаимодействия открытых систем и тесно связаны с протоколами других уровней. В связи с этим термин «TCP/IP» обычно охватывает все, что связано с протоколами TCP и IP. Сюда входят целое семейство протоколов, прикладные программы и даже сама сеть.

Протокол UDP (User Datagram Protocol) – протокол пользовательских дейтаграмм, является одним из двух протоколов, расположенных непосредственно над протоколом IP. Он представляет прикладным процессам ограниченный набор транспортных услуг, не обеспечивая надежной доставки дейтаграмм. Протокол UDP использует такие сетевые приложения, как NFS (Network File System) – сетевая файловая система и SNMP (Simple Network Management Protocol) – простой протокол управления сетью.

Протокол Telnet является протоколом эмуляции терминала и позволяет рассматривать все удаленные терминалы как стандартные сетевые виртуальные терминалы.

Протокол FTP (File Transfer Protocol) – протокол передачи файлов, позволяет пользователю просмотреть каталог удаленного компьютера, скопировать один или несколько файлов.

Протокол SMTP (Simple Mail Transfer Protocol) – простой протокол передачи почты, поддерживает передачу электронной почты между произвольными узлами сети Интернет.

Протокол ARP (Address Resolution Protocol) – протокол разрешения адресов, осуществляет преобразование (отображение) IP-адресов в Ethernet-адреса. Обратное преобразование осуществляется с помощью протокола RARP (Reverse Address Resolution Protocol) – обратного протокола разрешения адресов.

Последовательность протоколов, непосредственно участвующих в передаче информации, называется стеком протоколов, или протокольным стеком. Например, при передаче файлов через сеть Ethernet протокольный стек содержит FTP/TCP/IP/IEEE 803.2 протоколы.

Кроме описанных выше услуг, сетью Интернет предоставляются следующие услуги:

- **Telnet** – удаленный доступ, пользователь может работать на любом компьютере сети как на своем собственном;
- **Rsh** (Remote Shell) – удаленный доступ, в отличие от Telnet, работает только в случае, когда на удаленном компьютере установлена операционная система UNIX;
- **NFS** (Network File System) – распределенная файловая система, абонент может пользоваться файловой системой удаленного компьютера как своей собственной;
- **Rexec** (Remote Execution) – выполнение одной команды на удаленном Unix-компьютере;
- **Lpr** – сетевая печать, то есть отправка файлов на печать на удаленном (сетевом) принтере;
- **Lpq** – сетевая печать, показывает файлы, стоящие в очереди на печать на сетевом принтере;

- **Talk** – обеспечивает установление взаимодействия («разговора») с пользователем удаленного компьютера, при этом на экране дисплея одновременно видны вводимый текст и ответ удаленного пользователя;
- **Intunnet** – позволяет получить доступ к серверу ЛВС NetWare, с которым отсутствует непосредственная связь по ЛВС, но есть связь по сети Интернет;
- **Whois** – адресная книга сети Интернет, абонент которой имеет возможность по своему запросу получить информацию о принадлежности удаленного компьютера и о пользователях;
- **Finger** – получение информации о пользователях удаленного компьютера;
- **Webster** – сетевая версия толкового словаря английского языка;
- **Факс-сервис** – дает возможность пользователю отправлять сообщения по факсимильной связи, пользуясь факс-сервером сети;
- **Электронный переводчик** – осуществляет перевод текста с одного языка на другой, обращение к электронным переводчикам происходит через электронную почту;
- **Шлюзы** – дают возможность абоненту отправлять сообщения в сети, не работающие с протоколами TCP/IP.

Для подключения индивидуального компьютера к сети Интернет необходимо иметь модем, телефонную линию и провайдера – поставщика сетевых услуг, имеющего шлюз и эту сеть. Обычно при этом предлагается **коммутируемый** (dial-up) доступ к ресурсам сети и предоставляется возможность использовать компьютер поставщика, непосредственно подключенный к Интернету.

Такой компьютер называется **хостом**. На хосте запускаются имеющиеся у поставщика программы-клиенты, которые позволяют получить доступ к каждому серверу сети. Подобное коммутируемое подключение отличается тем, что обеспечивает доступ только к тем клиентам, которые имеются на хост-машине. Пересылаемая из сети информация сначала попадает на хост, а затем перекачивается на индивидуальный компьютер.

Более полноценным является такое подключение индивидуального компьютера к сети, когда провайдеры обеспечивают связь по коммутируемым линиям по протоколу SLIP или PPP. В этом случае индивидуальный компьютер превращается как бы в хост-компьютер; программы и файлы, получаемые из сети, хранятся на индивидуальном компьютере.

Локальная сеть подключается к Интернету обычно не по коммутируемым линиям, а по выделенным арендуемым линиям связи через шлюз с использованием дополнительного программного обес-

печения. Это прямое (on-line) подключение, обеспечивающее оперативное предоставление сетевых ресурсов организации, располагающей большим количеством компьютеров, объединенных в локальную сеть. Для доступа к Web-серверам и другим ресурсам сети Интернет каждый компьютер ЛВС должен иметь IP-адрес. Такой доступ обеспечивает шлюз (коммуникационный узел), связывающий ЛВС с Интернетом.

Организация связи в многоточечном режиме, обеспечивающем экономию на транспортных расходах, популярна в компьютерных сетях (особенно в ЛВС) из-за снижения затрат по сравнению с затратами при большом количестве монопольно используемых сетевых ресурсов в двухточечном режиме.

Коммутируемой транспортной сетью называется сеть, в которой между двумя (или более) конечными пунктами устанавливается связь по запросу.

Существуют соединяющие методы коммутации:

- коммутация каналов (целей);
- коммутация с промежуточным хранением, в свою очередь, разделяемая на коммутацию сообщений и коммутацию пакетов.

Сети с коммутацией каналов используются в корпоративных сетях в основном для удаленного доступа многочисленных домашних пользователей и гораздо реже – для соединения локальных сетей.

Отличительными особенностями всех сетей с коммутацией каналов являются: работа в режиме установления соединений; возможность блокировки вызова конечным абонентом или промежуточным коммутатором, необходимость использования на общих концах сети устройств, поддерживающих одну и ту же скорость передачи данных, так как этот вид сетей не выполняет промежуточной буферизации данных.

Сети с коммутацией каналов делятся на аналоговые и цифровые. Аналоговые сети могут использовать аналоговую (FDM – Frequency Division Multiplexing) и цифровую (TDM – Time Division Multiplexing) коммутации, на в них всегда абонент подключен по аналоговому двухпроводному окончанию. В цифровых сетях мультиплексирование и коммутация всегда выполняются по способу коммутации TDM, а абоненты всегда подключаются по цифровому абонентскому окончанию DSL – Digital Subscriber Line.

Указанные сети хорошо коммутируют потоки данных постоянной интенсивности, но не могут перераспределять пропускную способность магистральных каналов между потоками абонентских каналов динамически.

Общая структура сети с коммутацией каналов представлена на рис. 70.

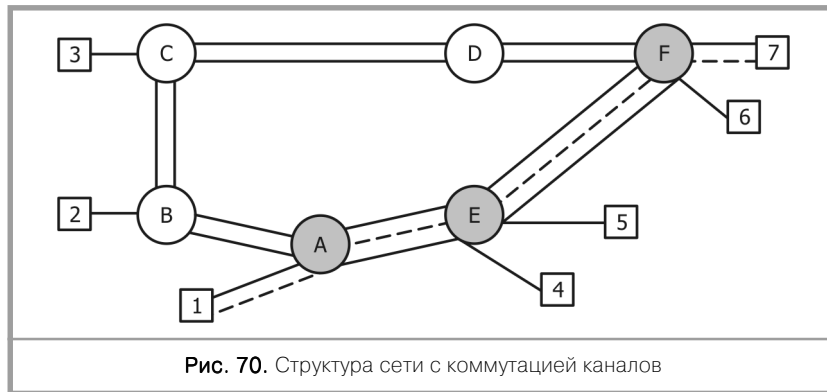


Рис. 70. Структура сети с коммутацией каналов

Сети с коммутацией пакетов были специально разработаны для эффективной передачи пульсирующего компьютерного трафика. Буферизация пакетов разных абонентов в коммутаторных позволяет сгладить неравномерности интенсивности трафика каждого абонента и равномерно загрузить каналы связи между коммутаторами.

При коммутации пакетов все передаваемые пользователи сети сообщения разбиваются в исходном узле на сравнительно небольшие части, называемые пакетами. **Сообщение** – логически завершенная порция данных – запрос на передачу файла, ответ на этот запрос, содержащий весь файл. Сообщения могут иметь произвольную длину, от нескольких байт до многих мегабайт. Пакеты обычно тоже могут иметь переменную длину, но в узких пределах, например от 46 до 1500 байт. Каждый пакет снабжается заголовком, в котором указывается адресная информация, необходимая для доставки пакета узлу назначения, а также номер пакета, который будет использоваться узлом назначения для сборки сообщения (рис. 71). Пакеты трансформируются в сети как независимые информационные блоки. Коммутаторы сети принимают пакеты от конечных узлов и на основании адресной информации передают их друг другу, а в конечном итоге – узлу назначения.

Коммутаторы пакетной сети отличаются от коммутаторов каналов тем, что имеют внутреннюю буферную память для временного хранения пакетов, когда выходной порт коммутатора в момент принятия пакета занят передачей другого пакета.

Сети с коммутацией пакетов были разработаны для эффективной передачи пульсирующего компьютерного трафика. Буферизация пакетов разных абонентов в коммутаторах позволяет сгладить

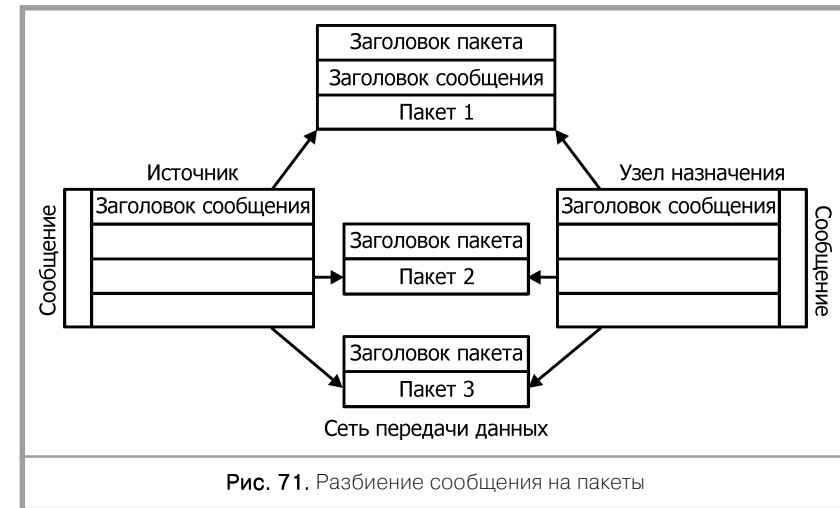


Рис. 71. Разбиение сообщения на пакеты

неравномерности интенсивности трафика каждого абонента и равномерно загрузить каналы связи между коммутаторами. Объем передаваемых данных от всех абонентов сети в единицу времени больше, чем при использовании сети с коммутацией каналов.

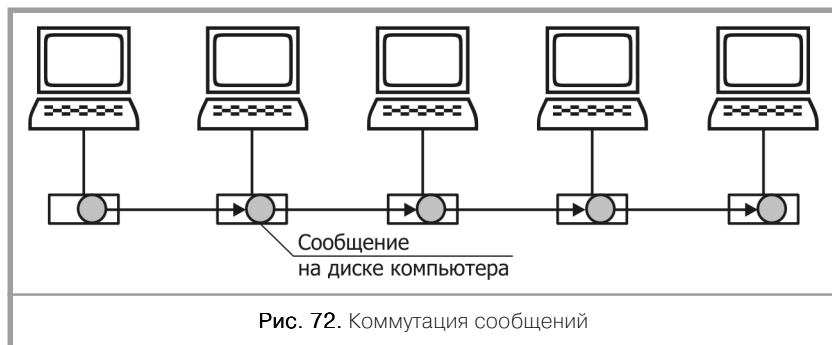
Сети с коммутацией пакетов могут работать в одном из двух режимов: дейтаграммном режиме или режиме виртуальных каналов. Дейтаграммный способ передачи данных основан на том, что все передаваемые пакеты обрабатываются независимо друг от друга, пакет за пакетом. Механизм виртуальных каналов учитывает существование в сети потоков данных и прокладывает для всех пакетов потока единый маршрут.

Если предприятие (корпорация) не строит свою территориальную сеть, а пользуется услугами общественной, то внутренняя структура этой сети его не интересует. Для абонента общественной сети главное – это предоставление сетью услуги определения интерфейса взаимодействия с сетью, чтобы его окончательное оборудование и программное обеспечение корректно сопрягались с соответствующим оборудованием и программным обеспечением общественной сети.

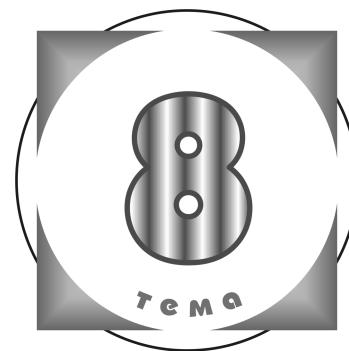
Поэтому в глобальной сети с коммуникацией пакетов обычно строго описан и стандартизирован **интерфейс** «пользователь – сеть» (User-to-Network Interface, UNI). Это необходимо для того, чтобы пользователи могли без проблем подключаться к сети с помощью коммутационного оборудования любого производителя, который соблюдает стандарт UNI данной технологии (например, X.25).

Протокол взаимодействия коммутаторов внутри глобальной сети называется **интерфейс** «сеть – сеть» (Network-to-Network Interface, NNI) и используется в основном при взаимодействии двух территориальных сетей различных операторов.

Коммутация сообщений по своим принципам близка к коммутации пакетов. Под **коммутацией сообщений** понимается передача единого блока данных между транзитными компьютерами сети с временной буферизацией этого блока на длине каждого компьютера (рис. 72). Сообщение, в отличие от пакетов, имеет произвольную длину, которая определяется не технологическими соображениями, а содержанием информации, составляющей сообщение. Транзитные компьютеры могут соединяться между собой как сетью с коммутацией пакетов, так и сетью с коммутацией каналов. Сообщение хранится в транзитном компьютере на диске, причем время хранения может быть достаточно большим, если компьютер загружен другими работами или сеть временно перегружена.



По такой схеме передаются сообщения, не требующие немедленного ответа, чаще всего сообщения электронной почты. Режим передачи с промежуточным хранением на диске называется режимом хранения и передачи (store-and-forward) и предназначен для организации взаимодействия пользователей в автономном (off-line) режиме.



Распределенные базы данных

8.1. Понятия базы данных и базы знаний

База данных (*database*) – совокупность взаимосвязанных данных, организованная по определенным правилам (БД).

База знаний (*Knowledge base*) – организованная совокупность знаний, относящихся к какой-нибудь предметной области (БЗ).

База данных является источником и своего рода полуфабрикатом при подготовке информационных услуг соответствующим службам. С появлением компьютеров существенно увеличиваются объемы хранимых баз данных и соответственно расширяется круг информационных услуг.

Строго говоря, базой данных являются специальным образом организованные один или группа файлов на компьютере. Для работы с ними используется **система управления базой данных (СУБД)**. При этом подразумевается, что база данных определена по схеме, не зависящей от программ, которые к ней обращаются. База данных характеризуется ее концепцией – совокупностью требований, обусловленных представлениями пользователей о необходимой им информации. Каждая БД состоит из записей, которые образуют подмножество данных, служащих для описания единичного объекта.

По технологии обработки данных БД подразделяются на **централизованные** и **распределенные**.

Централизованная база данных хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе. Такой способ использования баз данных часто применяют

в локальных сетях персональных компьютеров (ПК). Системы централизованных баз данных с сетевым доступом имеют следующие архитектуры систем:

- файл-сервер;
- клиент-сервер.

Распределенная база данных (distributed database) – БД, содержимое которой находится в нескольких абонентских системах информационной сети.

Сущность распределенной базы данных заключается в организации доступа пользователей к большим объемам информации. Это позволяет располагать данные так, что они, с одной стороны, находятся в пунктах наибольшего их спроса, а с другой – с помощью транзакции (короткий во времени цикл взаимодействия объектов, включающий запрос – выполнение задания – ответ), обеспечивается доступ к любым данным независимо от того, где они находятся.

Транзакция (*transaction*) – короткий по времени цикл взаимодействия объектов, включающий запрос – выполнение задания – ответ.

Характерным примером транзакции является обращение к БД в режиме диалога. С терминала либо ПК направляется задание на поиск информации. После этого в режиме реального времени следуют быстрый поиск и выдача пользователю найденной информации.

Транзакции широко используются в сетевой службе EDI-Electronic Data Interchange (электронный обмен данными), видеотексте, телетексте, видеоконференциях.

Распределения базы данных, создаваемая заново, является **однородной**. Вместе с этим она нередко образуется как совокупность группы баз данных, уже функционирующих в ряде систем. В этом случае возникает **неоднородная** распределенная база. Оба типа баз погружаются в распределенную систему управления базой данных (СУБД).

Реляционная база данных (*relation database*) – база данных, логически организованная в виде набора ее компонентов. Понятие реляционный – relation (отношение) – это модели, которые характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Характерной особенностью реляционной базы является архитектура, выполненная в виде таблиц. Строки таких таблиц соответствуют записям, столбцы – атрибутам (признакам хранимых данных). Например, таблица, в которой имеются столбцы: фамилия, имя, отчество, год рождения, место работы, домашний адрес, телефон, – а в строках записываются эти сведения о сотрудниках предприятия. Такие данные являются ядром реляционной базы.

Язык запросов к рассматриваемой базе основан на реляционной алгебре, основными операциями которого являются: проекция, соединение, пересечение и объединение. Этот язык позволяет описывать условия поиска информации, не задавая для этого последовательность действий, нужных для получения ответа.

Использование реляционных баз данных позволяет:

- собирать и хранить данные в виде таблиц;
- обновлять их содержание;
- получать разнообразную информацию по атрибутам или записям;
- отображать полученные данные в виде диаграмм или таблиц;
- выполнять необходимые расчеты по материалам базы.

Базы данных и программные средства их создания и ведения (СУБД) имеют **многоуровневую архитектуру**, представленную на рис. 73. Различают концептуальный, внутренний и внешний уровни представления БД, которым соответствуют модели аналогичного значения.

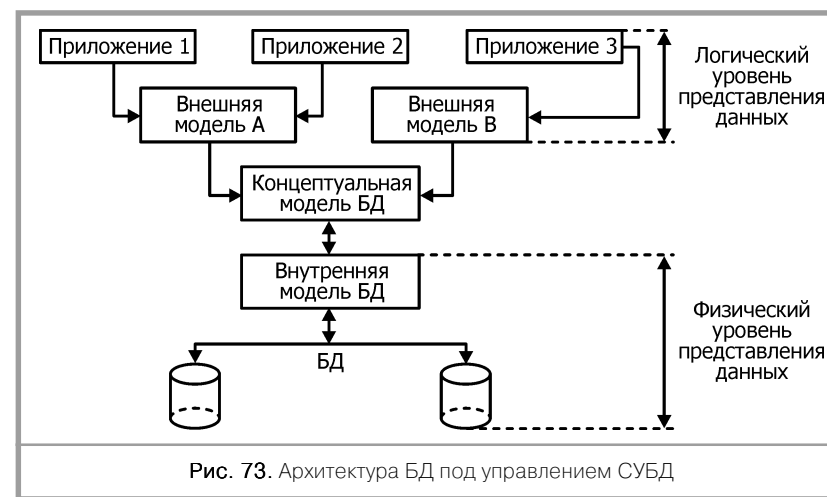


Рис. 73. Архитектура БД под управлением СУБД

Концептуальный уровень соответствует логическому уровню представления данных предметной области в интегрированном виде. **Концептуальная модель** состоит из множества различных типов данных, структурированных в соответствии с требованиями СУБД к логической структуре БД.

Внутренний уровень отображает требуемую организацию данных в среде хранения и соответствует физическому уровню пред-

ставления данных. **Внутренняя модель** состоит из отдельных экземпляров записей, физически хранимых во внешних носителях.

Внешний уровень поддерживает частные представления данных, требуемые конкретным пользователям. **Внешняя модель** является подмножеством концептуальной модели. Возможно пересечение внешних моделей по данным. Частная логическая структура данных для отдельного приложения (задачи) или пользователя соответствует внешней модели или подсхеме БД. С помощью внешних моделей поддерживается санкционированный доступ к данным БД приложений (ограничены состав и структура данных концептуальной модели БД, доступных в приложении, а также заданы допустимые режимы обработки этих данных: ввод, редактирование, удаление, поиск).

8.2. Система управления базой данных

Система управления базой данных (*database management system*) – комплекс языков и программ, позволяющих создавать базу данных и управлять ее функционированием.

Система управления базой данных (СУБД) обрабатывает обращение к базе данных, поступающее от пользователей, прикладных процессов, и выдает необходимые сведения. СУБД характеризуется используемой моделью и средствами администрирования, разработки прикладных процессов, работы в информационной сети.

СУБД обеспечивает:

- описание и сжатие данных;
- манипулирование данными (запись, поиск, выдачу, изменение содержания);
- физическое размещение (изменение размеров блоков данных, записей, использование занимаемого пространства, сортировку);
- защиту от сбоев, поддержку целостности и восстановление;
- работу с транзакциями и файлами;
- безопасность данных.

Существует несколько типов СУБД. В иерархической системе управления базой данных данные в соответствии с ветвящимся деревом их признаков располагаются в двухмерных файлах и образуют деревья признаков (рис. 74). Соответственно этому происходит и поиск необходимых сведений.

В **реализационных системах** управления базами данных данные представляются в форме таблиц, определяющих взаимосвязь записей. Реляционные СУБД характеризуются простотой, гибкостью и точностью. Каждая из них одновременно работает с данны-

№ п/п	Характеристики	Иерархические СУБД	Реляционные СУБД
1	Хранение данных	В двухмерных файлах	В двухмерных таблицах
2	Запись, упорядочение и считывание данных	В каждом файле по очереди	В таблицах, между которыми установлена связь
3	Управление связями	Обеспечивается программой, обрабатывающей данные	При помощи ссылок и ключей

Рис. 74. Типы СУБД

ми, размещенными в нескольких таблицах. Поэтому реляционные базы данных ориентированы на быстрый доступ к небольшим объектам данных.

Объектно-ориентированные системы управления базами данных основываются на объектно-ориентированной архитектуре. Они позволяют работать со сложными типами данных, хранимыми в виде объектов. Рассматриваемые СУБД отличаются высокой производительностью при обработке транзакций и поэтому особенно эффективны при обработке изображений.

Гибридные системы управления базами данных определяют положительные качества реляционных и объектно-ориентированных систем. Они объединяют средства обработки транзакций реляционных СУБД с поддержкой многочисленных типов данных объектно-ориентированных запросов (SQL).

СУБД определяет модель представления данных. В последние годы быстро развивается архитектура клиент-сервер, которая базируется на широком использовании больших и малых абонентских систем. СУБД типа «клиент-сервер» использует распределенную базу данных (рис. 75). Здесь прикладной процесс системы «А» является клиентом базы. Сама база данных распределена по серверам «В», «С». В соответствии с этим через коммуникационную сеть выполняются два вида взаимодействия. Первый связан с распределенной разработкой данных. Второй вид осуществляет выполнение транзакций между клиентом и серверами, представляющими базу. Работа распределенной базы данных обеспечивается взаимодействием прикладных процессов систем «В», «С». Работу СУБД поддерживают функции, выполняемые областью взаимодействия.

Разновидностью СУБД является информационно-поисковая система, задача которой – выполнение заявок пользователей о пред-

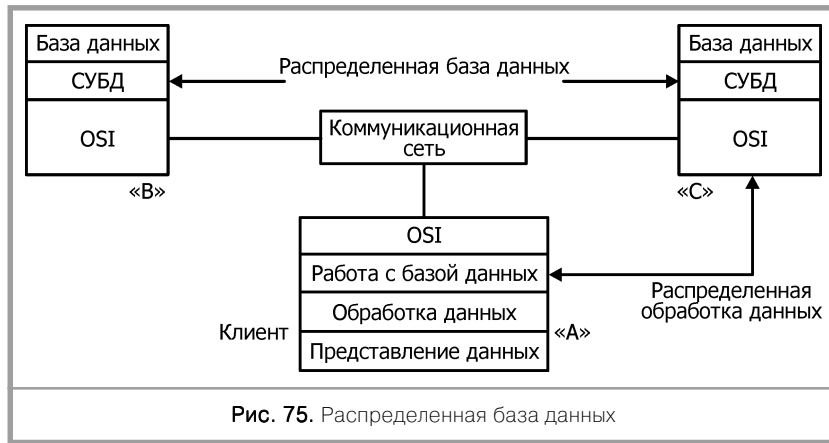


Рис. 75. Распределенная база данных

ставлении им необходимых документов, хранящихся в данной базе. Широкую известность получили система управления базой данных dBase и система управления базой данных Oracle.

Система управления базой данных DBASE (dBase database management system) – система управления базой данных, распространяемая корпорацией Borland Internation.

dBase характерна символьным языком программирования с графическим интерфейсом пользователя. Важной особенностью этой системы является возможность ее погружения в значительное число операционных систем, в том числе в операционную систему UNIX. Система характеризуется простой работой с ней пользователей и широкими функциональными возможностями. Пользователь работает с базой данных с помощью не только клавиатуры, но и мыши.

dBase предназначена для работы с персональными компьютерами, на основе которых строятся развитые информационные комплексы. Последние могут функционировать как автономно, так и включаться в сеть.

dBase рассчитана на разработчиков информационных комплексов и пользователей, работающих с этими комплексами в режиме диалога.

dBase использует развитый язык программирования задач обработки данных. В системе dBase предусмотрены средства создания баз данных и управления этими базами. Обеспечивается также использование экранных форм ввода-вывода данных, генерации отчетов. Система имеет свой текстовый редактор и реализована на языке «С».

Система управления базой данных ORACLE (Oracle database management system) – система управления базой данных, разработанная корпорацией Oracle.

Система создана для компьютеров системы DEC серии PDP-11, а затем версии для разнообразных технических устройств от персональных компьютеров до суперкомпьютеров.

Характерными особенностями системы являются:

- высокая производительность и работа с распределенными базами данных;
- возможность выполнения нескольких прикладных процессов;
- использование языка запросов (SQL);
- применение архитектуры клиент-сервер;
- развитое инструментальное программное обеспечение создания сложных прикладных программ;
- работа в операционной системе UNIX;
- мощные средства обеспечения безопасности данных.

Oracle может одновременно управлять обработкой до десятков тысяч потоков самых разнообразных данных: текстов, речи, изображений. Информация из базы данных может выдаваться со скоростью 30 изображений в секунду и сопровождаться стереофоническим звуком.

Основой системы является многопроцессорный сервер.

Oracle может автоматически перестраивать свою архитектуру в зависимости от величины загрузки, оптимально использовать оперативную память, процессоры, каналы ввода-вывода данных. Выполнением прикладных процессов занимаются процессы-диспетчеры.

Фирма Oracle совместно с корпорацией Novell создала версию, предназначенную для работы в сетях NetWare. Наряду с этим Oracle погружается практически во все операционные системы, используемые в локальных сетях.

8.3. Распределенная система управления базой данных

Распределенная система управления базой данных (distributed database management system) – система управления базой данных, содержимое которой располагается в нескольких абонентских системах информационной сети. Задачей распределенной системы управления базой данных (РСУБД) является обеспечение функционирования распределенной базы данных. РСУБД должна действовать так, чтобы у пользователей возникла иллюзия того, что они работают с базой данных, расположенной в одной абонентской сис-

теме. Использование РСУБД по сравнению с группой независимых баз данных позволяет сокращать затраты на передачу данных в информационной сети. РСУБД так распределяет файлы по сети, что в каждой системе хранятся те данные, которые чаще всего используются именно в этом месте.

РСУБД характеризуются следующими особенностями:

- использованием распределенного словаря, содержащего сведения о характере имеющихся данных, их размещении и способе доступа к ним;
- выполнением транзакций или обеспечением работы электронной почты между всеми абонентскими системами;
- обеспечением целостности данных при связи между электронными таблицами, находящимися в различных системах;
- независимостью способа и процедуры доступа к файлу от места его хранения;
- работа с частью базы данных, расположенной в одной системе, не может быть прервана обращением из другой системы;
- администратор части базы, находящейся в одной системе, работает независимо от администраторов частей базы, расположенных в других системах.

В РСУБД осуществляется принцип дублирования. Его сущность заключается в том, что изменение, вносимое в одну часть базы данных, в течение определенного времени отражается и в других частях базы.

Распределенная среда обработки данных (DCE) – Distributed Computing Environment – технология распределенной обработки данных, предложенная организацией OSF (Open Software Foundation – фонд открытого программного обеспечения, занимающийся разработкой технологий совместимости программного обеспечения).

Среда предназначена для выполнения прикладных процессов, рассредоточенных по группе абонентских систем гетерогенной сети. Системы, имеющие программы распределенной среды, соответственно являются серверами и клиентами. Серверы связаны друг с другом логическими каналами, по которым передают друг другу файлы (рис. 76). Каждый сервер имеет свою группу клиентов.

Среда имеет трехступенчатую архитектуру:

- прикладная программа;
- база данных;
- клиент.

Функции, выполняемые средой, записаны на языке «С» и включают:

- службу каталогов, позволяющих клиентам находить нужные им серверы;

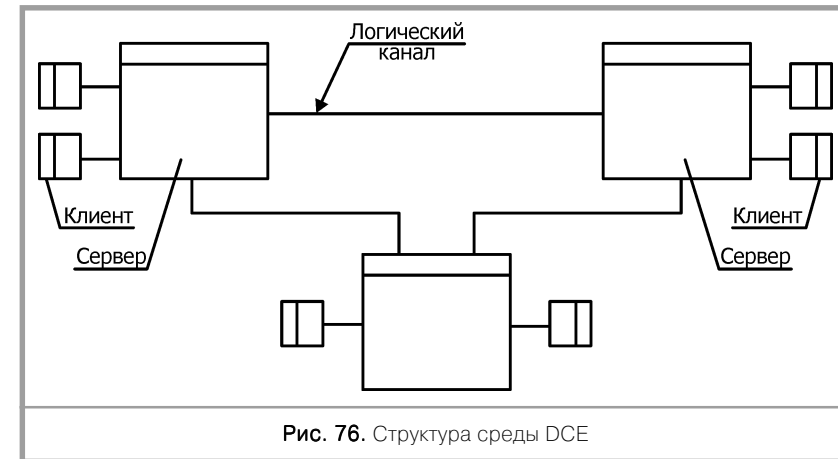


Рис. 76. Структура среды DCE

- программу, обеспечивающую интерфейс многопоточной обработки;
- службу удаленного вызова процедур;
- средства обслуживания файлов;
- службу безопасности данных;
- службу времени, синхронизирующую часы в абонентских системах.

Программное обеспечение среды погружается в сетевую операционную систему. Серверы имеют свои различные операционные системы. В роли сервера может выступать главный компьютер со своей операционной системой.

Функционирование распределенной среды требует выполнения ряда административных задач.

К ним в первую очередь относятся средства:

- регистрации и контроля за лицензиями пользователей на работу с прикладными программами;
- унифицированных интерфейсов прикладных программ;
- обеспечения безопасности данных;
- инвентаризации программного и технического обеспечения абонентских систем, работающих в сети.

Распределенная среда создается различными фирмами. Так, корпорация Novell создала среду программирования AppWare – AppWare programming environment – инструментарий, технологий и прикладные программы многократного применения, использующие объектно-ориентированную архитектуру.

Корпорация Novell предложила так называемую открытую AppWare, которая позволяет программистам вместо написания монолитных прикладных сетевых программ решать поставленные перед ними задачи путем блочного структурирования загружаемых модулей AppWare (ALM – Loadable Module – загружаемые модули – среда программирования). Характеристики модулей ALM, разработанные в стандартах AppWare, не зависят от особенностей платформ и операционных систем, которые обеспечивают выполнение прикладных процессов. Это позволяет осуществлять сегментацию прикладных программ и разрабатывать только недостающие части, в результате чего резко увеличивается производительность труда разработчиков.

Загружаемые модули ALM и модули, созданные разработчиками, опираются на общий программный механизм, именуемый AppWare (рис. 77). Он обеспечивает взаимодействие всех модулей и управление этим процессом.

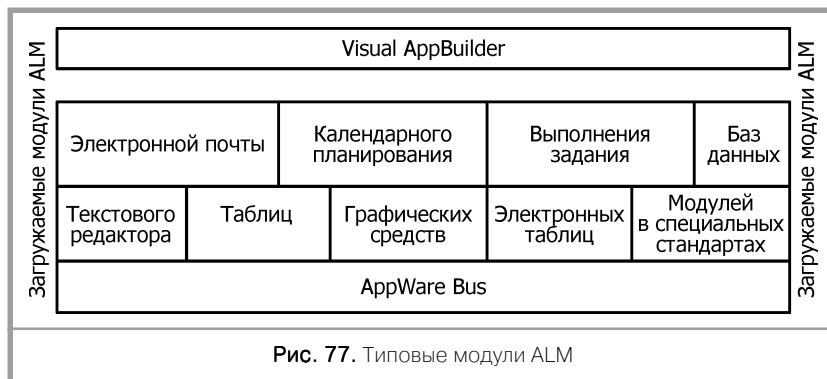


Рис. 77. Типовые модули ALM

Компонент Visual App Builder определяет инструментальное программное обеспечение, с помощью которого программисты могут описать свои программы. Простота применения этого продукта обеспечивается заранее подготовленными блоками, представляемыми пиктограммами. Наряду с модулями ALM здесь могут использоваться модули-полуфабрикаты, обеспечивающие доступ к базе данных, создание службы каталогов, организацию окон, через которые осуществляется диалог, и т. д.

Компонент AppWare Foundation, на который опирается рассмотренный программный механизм, является фундаментом среды, определяющим интерфейс прикладной программы (API). Он связан

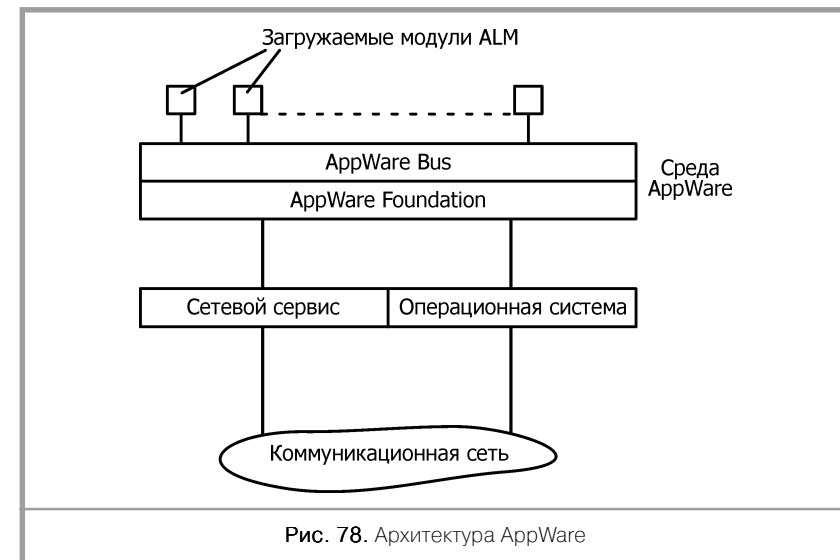


Рис. 78. Архитектура AppWare

с сетевым сервисом и операционной системой, обеспечивающей выполнение прикладных программ (рис. 78).

Рынок предлагаемых модулей многократного использования быстро расширяется, а среда AppWare позволяет выбирать и связывать друг с другом такие модули, которые наилучшим образом соответствуют требованиям, поставленным перед проводимыми разработками. В свою очередь, AppWare обеспечивает доступ к этим модулям путем удобного графического интерфейса.

Прикладные процессы, созданные в среде AppWare, стыкуются с областью взаимодействия, определяемой широко используемыми этажерками протоколов. Например, Протокол управления передачей / Межсетевой протокол (TCP/IP), Протокол IPX/SPX, системная сетевая архитектура (SNA), архитектура дискретной сети (DNA).

Среда AppWare позволяет использовать технологию OpenDoc, применяя сетевую операционную систему AIX, операционную систему Macintosh, операционную систему OS/2, операционную систему OS/2, операционную систему UNIX Ware, операционную систему Windows и ряд других операционных систем.

По степени универсальности различают два класса СУБД:

- системы общего назначения;
- специализированные системы.

8.4. Классы СУБД

СУБД общего назначения не ориентированы на какую-либо предметную область или на информационные потребности какой-либо группы пользователей. Каждая система такого рода реализуется как программный продукт, способный функционировать на некоторой модели ЭВМ в определенной операционной системе, и поставляется многим пользователям как коммерческое издание. Такие СУБД обладают средствами настройки на работу с конкретной базой данных. Использование СУБД общего назначения в качестве инструментального средства для создания автоматизированных информационных систем, основанных на технологии баз данных, позволяет существенно сокращать сроки разработки, экономить трудовые ресурсы. Этим СУБД присущи развитые функциональные возможности и даже определенная функциональная избыточность.

Специализированные СУБД создаются в редких случаях при невозможности или нецелесообразности использования СУБД общего назначения.

СУБД общего назначения – это сложные программные комплексы, предназначенные для выполнения всей совокупности функций, связанных с созданием и эксплуатацией базы данных информационной системы.

Рынок программного обеспечения ПК располагает большим числом разнообразных по своим функциональным возможностям коммерческих систем управления базами данных общего назначения, а также средствами их окружения практически для всех массовых моделей машин и для различных операционных систем.

Используемые в настоящее время СУБД обладают средствами обеспечения целостности данных и надежной безопасности, что дает возможность разработчикам гарантировать большую безопасность данных при меньших затратах сил на низкоуровневое программирование.

Продукты, функционирующие в среде Windows, выгодно отличаются удобством пользовательского интерфейса и встроенными средствами повышения производительности.

Основные характеристики некоторых СУБД – лидеров на рынке программ, предназначенных как для разработчиков информационных систем, так и для конечных пользователей, следующие:

- dBase N. 2.0 компании Borland International;
- Microsoft Access 2.0;
- Microsoft FoxPro 2.6 for DOS;
- Microsoft FoxPro 2.6 for Windows корпорации Microsoft Corp;
- Paradox for DOS 4.5;
- Paradox for Windows, версия 4.5 компании Borland.

На рис. 79 показаны места (условные), которые занимают рассматриваемые программные средства относительно друг друга. Например, «1» означает, что в указанной позиции данная программа обладает лучшими характеристиками, «5» – худшими, нет – указанной характеристикой данная программа не обладает.

Производительность СУБД оценивается:

- временем выполнения запросов;
- скоростью поиска информации в неиндексированных полях;
- временем выполнения операций импортирования базы из других форматов;
- скоростью создания индексов и выполнения таких массовых операций, как обновление, вставка, удаление данных;
- максимальным числом параллельных обращений к данным в многопользовательском режиме;
- временем генерации отчета.

На производительность СУБД оказывают влияние два фактора:

- СУБД, следящие за соблюдением целостности данных, несут дополнительную нагрузку, которую не испытывают другие программы;
- производительность собственных прикладных программ сильно зависит от правильного проектирования и построения базы данных.

Самые быстрые программные изделия отнюдь не обладают самыми развитыми функциональными возможностями на уровне процессора СУБД.

Из рис. 79 можно заключить, что самой быстрой СУБД является FoxPro 2.6, однако она не обладает средствами соблюдения целостности данных, в отличие от более медленной СУБД Access 2.0.

8.5. Обеспечение целостности данных на уровне базы данных

Эта характеристика подразумевает наличие средств, позволяющих удостовериться, что информация в базе данных всегда остается корректной и полной. Должны быть установлены правила целостности, и они должны храниться вместе с базой данных и соблюдаться на глобальном уровне.

Целостность данных должна обеспечиваться независимо от того, каким образом данные заносятся в память (в интерактивном режиме, посредством импорта или с помощью специальной программы).

К средствам обеспечения целостности данных на уровне СУБД относятся:

Наименование	dBase IV 2.0	Microsoft Access 2.0	Microsoft FoxPro for DOS 2.6	Microsoft FoxPro for Windows 2.6	Paradox for DOS 4.5	Paradox for Windows 4.5
Производительность	4	3	1	1	2	2
Обеспечение целостности данных на уровне базы данных	Нет	1	Нет	Нет	2	2
Обеспечение безопасности	2	1	5	5	3	4
Работа в многопользовательских средах	2	2	4	4	1	3
Импорт / экспорт	2	1	2	1	1	1
Доступ к данным SQL	2	1	2	2	Нет	3
Возможности запросов и инструментальные средства разработки прикладных программ	3	3	1	1	2	4

Рис. 79. Характеристики СУБД

- встроенные средства для назначения первичного ключа. В том числе средства для работы с типом полей с автоматическим приращением, когда СУБД имеют самостоятельно новое, уникальное значение;
- средства поддержания ссылочной целостности, которые обеспечивают запись информации о связях таблиц и автоматически пересекают любую операцию, приводящую к нарушению ссылочной целостности.

Некоторые СУБД имеют хорошо разработанный процессор СУБД для реализации таких возможностей, как уникальность первичных ключей, ограничение (пресечение) операций и даже прикладное обновление и удаление информации. В подобных системах проверка корректности, назначаемая полю или таблице, будет проводиться всегда после изменения данных, а не только во время ввода информации с помощью экранной формы. Это свойство можно настраивать для каждого поля и для записи в целом, что позволяет контролировать не только значения отдельных полей, но и взаимосвязи между несколькими полями данной записи.

Access и Paradox for Windows гораздо ближе других СУБД соответствуют реляционной модели по надежности сохранения целостности данных на уровне базы данных; правила хранятся вместе с базой данных и автоматически соблюдаются.

СУБД dBase IV и FoxPro 2.6 (DOS и WINDOWS) совсем не обладают средствами этого рода, и ввод в программу процедур, обеспечивающих выполнение правил целостности, возлагается на программиста.

Обеспечение безопасности. Некоторые СУБД предусматривают средства обеспечения безопасности данных. Такие средства обеспечивают выполнение следующих операций:

- шифрование прикладных программ;
- шифрование данных;
- защиту паролем;
- ограничение уровня доступа (к базе данных, к таблице, к словарю, для пользователя).

Самый высокий уровень безопасности данных реализован в СУБД dBase IV. Администратор может назначать системе различные права доступа на уровне файла, поля, а также организовать автоматическое шифрование данных.

Хорошими характеристиками обеспечения безопасности отличается Access 2.0. Он предусматривает назначение паролей для индивидуальных пользователей и присвоение различных прав доступа отдельно таблицам, запросам, отчетам, макрокомандам или новым объектам на уровне пользователя или группы.

Практически все рассматриваемые СУБД предназначены для работы в **многопользовательских средах**, но обладают для этого различными возможностями.

Обработка данных в многопользовательских средах предполагает выполнение программным продуктом следующих функций:

- блокировку базы данных, файла, записи, поля;
- идентификацию станции, установившей блокировку;
- обновление данных после модификации;
- контроль за временем и повторение обращения;
- обработку транзакции, то есть последовательность операций пользователя над базой данных, которая сохраняет ее логическую целостность (короткий во времени цикл взаимодействия объектов, включающий запрос – выполнение задания – ответ);
- работу с сетевыми системами (LAN Manager, NetWare, Unix).

Лучшими возможностями для работы в многопользовательских средах обладают СУБД Paradox for DOS 4.5, Access 2.0 и dBase IV.

8.6. Методы доступа к данным, находящимся в базах

Существует много методов доступа к данным, находящимся в базах. Особой популярностью пользуется метод, определяемый **языком структурированных запросов SQL** (Structured Query Language), который реализован в целом ряде популярных СУБД для различных типов ЭВМ либо как базовый, либо как альтернативный. В силу своего широкого использования он является международным стандартом языка запросов. Язык SQL предоставляет развитые возможности как конечным пользователям, так и специалистам в области обработки данных.

Совместимость с SQL-системами играет большую роль, когда предполагается проведение работы с корпоративными данными. СУБД, хорошо подготовленные к работе в качестве средств первичной обработки информации для SQL-систем, могут открыть дверь в системы с архитектурой клиент-сервер.

СУБД имеют доступ к данным SQL в следующих случаях:

- базы данных совместимы с ODBC (Open Database Connectivity – открытое соединение баз данных);
- реализована естественная поддержка баз данных SQL;
- возможна реализация SQL-запросов локальных данных.

Многие СУБД могут «прозрачно» подключаться к входным SQL-подсистемам с помощью ODBC или драйверов, являющихся их частью, поэтому существует возможность создания прикладных программ для них. Некоторые программные продукты совместимы также с SQL при обработке интерактивных запросов на получение данных, находящихся на сервере или на рабочем месте.

Access 2.0 и Paradox for Windows работают с источниками SQL-данных, совместимых с системой ODBC.

FoxPro (for DOS и for Windows) поставляются с дополнительными библиотеками, которые обеспечивают доступ к базам данных SQL, способным работать совместно с системой ODBC, но эта возможность менее интегрирована, чем средства первичного ввода информации в Access и Paradox for Windows.

Можно напрямую управлять базами данных Access с помощью языка SQL и передавать сквозные SQL-запросы совместимым со спецификацией ODBC базам данных SQL, таким как MS SQL Server и Oracle, так что Access способна служить средством разработки масштабируемых систем клиент-сервер.

СУБД, ориентированные на разработчиков, обладают развитыми средствами для создания приложений. К элементам **инструментария разработки приложений** можно отнести:

- мощные языки программирования;
- средства реализации меню, экранных форм ввода-вывода данных и генерации отчетов;
- средства генерации приложений (прикладных программ);
- генерацию исполнимых файлов.

Функциональные возможности моделей данных доступны пользователю СУБД благодаря ее языковым средствам.

Реализация языковых средств интерфейсов может быть осуществлена различными способами. Для высококвалифицированных пользователей (разработчиков сложных прикладных систем) языковые средства чаще всего представляются в их явной синтаксической форме.

В других случаях функции языков могут быть доступны косвенным образом, когда они реализуются в форме различного рода меню, диалоговых сценариев или заполняемых пользователем таблиц. По таким входным данным интерфейсные средства формируют адекватные синтаксические конструкции языка интерфейса и передают их на исполнение или включают в генерируемый программный код приложения. Интерфейсы с неявным использовани-

ем языка широко применяются в СУБД для персональных ЭВМ. Примером такого языка является язык Query-By-Example.

Языковые средства используются для выполнения двух основных функций:

- описания представления базы данных;
- выполнения операций манипулирования данными.

Первая из этих функций обеспечивается **языком описания (определения) данных (ЯОД)**. Описание базы данных средствами ЯОД называется **схемой базы данных**. Оно включает описание структуры базы данных и налагаемых на нее ограничений целостности в рамках тех правил, которые регламентированы моделью данных используемой СУБД. ЯОД некоторых СУБД обеспечивают также возможности задания ограничений доступа к данным или полномочий пользователей.

ЯОД не всегда синтаксически оформляется в виде самостоятельного языка. Он может быть составной частью единого языка данных, сочетающего возможности определения данных и манипулирования данными.

Язык манипулирования данными (ЯМД) позволяет запрашивать предусмотренные в системе операции над данными из базы данных.

Имеются многочисленные примеры языков СУБД, объединяющих возможности описания данных и манипулирования данными в единых синтаксических рамках. Популярным языком такого рода является реляционный язык SQL.

СУБД dBase IV и FoxPro поддерживают язык программирования XBASE, который до сих пор является важным стандартом для баз данных.

FoxPro 2.6 придает XBASE-программам оконные, событийно-управляемые качества. При составлении прикладной программы FoxPro использует диспетчер проекта, управляющий различными файлами исходного текста и данных. Эта составляющая отслеживает индивидуальные элементы: программы, набор экранных форм, отчеты и файлы баз данных – и позволяет компилировать прикладную программу в исполнимый файл.

Язык программирования Access Basic содержит функции обеспечения связи по протоколу OLE 2.0, позволяющие управлять проектами из других прикладных программ, совместимых с OLE 2.0. Кроме того, этот язык позволяет создавать объекты баз данных (запросы, таблицы), изменять структуру базы данных и создавать индексы непосредственно из прикладной программы.

8.7. Инструментальные средства СУБД

Все рассматриваемые программные средства обладают автоматизированными средствами создания экранных форм, запросов, отчетов, меню, наклеек, стандартных писем. Для создания указанных визуальных и структурных объектов ряд СУБД использует специальные инструментальные средства, называемые «мастерами», или «волшебниками».

При работе с СУБД на экран выводятся рабочее поле и панель управления. Панель управления при этом включает меню, вспомогательную область управления и строку подсказки. Расположение этих областей на экране может быть произвольным и зависит от особенностей конкретной программы. Некоторые СУБД позволяют вывести на экран окно директив (командное окно) или строку команд.

Строка меню содержит основные режимы программ. Выбрав один из них, пользователь получает доступ к ниспадающему подменю, содержащему перечень входящих в него команд. В результате выбора некоторых команд ниспадающего меню появляются дополнительные подменю.

Вспомогательная область управления включает:

- строку состояния;
- панели инструментов;
- вертикальную и горизонтальную линейки прокрутки.

В строке состояния (статусной строке) пользователь найдет сведения о текущем режиме работы программы, имени файла текущей базы данных и т. п. «Панель инструментов» (пиктографическое меню) содержит определенное количество кнопок (пиктограмм), предназначенных для быстрой активизации выполнения определенных команд меню и функций программы. Чтобы представить на экране области таблицы базы данных, формы или отчета, которые на нем в настоящий момент не отображены, используют «вертикальную» и «горизонтальную» линейки прокрутки.

«Строка подсказки» предназначена для выдачи сообщения пользователю относительно его возможных действий в данный момент.

Важная особенность СУБД – использование буфера промежуточного хранения при выполнении ряда операций. Буфер используется при выполнении команд копирования и перемещения для временного хранения копируемых или перемещаемых данных, после чего они направляются по новому адресу. При удалении данных они также помещаются в буфер. Содержимое буфера сохраняется до тех пор, пока в него не будет записана новая порция данных.

Программы СУБД имеют достаточное количество команд, у каждой из которых возможны различные параметры (опции). Такая система команд совместно с дополнительными опциями образует «меню» со своими особенностями для каждого типа СУБД. Выбор определенной команды из меню производится одним из следующих способов:

- наведением курсора на выбранную в меню команду при помощи клавиш управления курсором и нажатием клавиши ввода;
- вводом с клавиатуры первой буквы выбранной команды.

Получить дополнительную информацию о командах, составляющих меню СУБД, и их использовании можно, войдя в режим помощи.

Несмотря на особенности СУБД, совокупность команд, предоставляемых в распоряжение пользователю некоторой усредненной системой управления базами данных, может быть разбита на следующие типовые группы:

- команды для работы с файлами;
- команды редактирования;
- команды форматирования;
- команды для работы с окнами;
- команды для работы в основных режимах СУБД (таблица, форма, запрос, отчет);
- получение справочной информации.

При работе с файлами программа дает возможность пользователю:

- создавать новые объекты базы данных;
- сохранять и переименовывать ранее созданные объекты;
- открывать уже существующие базы данных;
- закрывать ранее открытые объекты;
- выводить на принтер объекты базы данных.

Ввод данных и изменение содержимого любых полей таблиц базы данных, компонентов экранных форм и отчетов осуществляются с помощью группы команд редактирования, главными из которых являются «перемещение», «копирование» и «удаление».

Наряду с вышеуказанными операциями большая группа программ СУБД обладает возможностями вставки диаграммы, рисунка и т. п., включая объекты, созданные в других программных средах, установление связей между объектами.

Большинство СУБД предоставляют в распоряжение пользователя большое число команд, связанных с оформлением выводимой информации. При помощи этих команд пользователь может варьировать направление выравнивания данных, виды шрифта, толщину и расположение линий, высоту букв, цвет фона и т. п. Выбор формата и направления выравнивания производится автоматически

в зависимости от характера вводимых данных. Автоматический выбор формата и способа выравнивания производится только в том случае, если для заполняемых ячеек пользователем предварительно не заданы другие параметры.

Большинство СУБД дает возможность открывать одновременно множество окон, организуя тем самым «многооконный режим» работы. При этом некоторые окна будут видны на экране, другие – находиться под ними.

СУБД имеют в своем составе электронные справочники, предоставляющие пользователю инструкции о возможностях выполнения основных операций, информацию по конкретным командам меню и другие справочные данные. В некоторых СУБД возможно нахождение потребной информации в справочнике путем задания темы поиска.

Каждая конкретная СУБД имеет свои особенности, которые необходимо учитывать. Однако, имея представление о функциональных возможностях любой СУБД, можно представить обобщенную технологию работы пользователей в этой среде. В качестве основных этапов обобщенной технологии работы с СУБД, которая схематично представлена на рис. 80, можно выделить следующие:

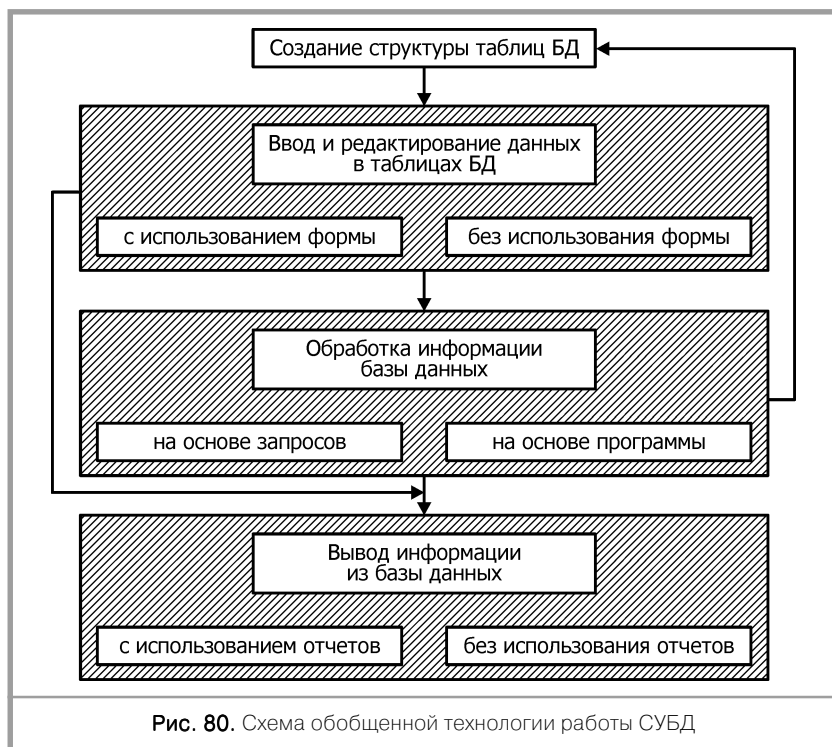
- создание структуры таблиц базы данных;
- ввод и редактирование данных в таблицах;
- обработка данных, содержащихся в таблицах;
- вывод информации из базы данных.

Практически все используемые СУБД хранят данные следующих типов: текстовый (символьный), числовой, календарный, логический, примечания. Некоторые СУБД формируют поля специального типа, содержащие уникальные номера записей и используемые для определения ключа.

СУБД, предназначенные для работы в Windows, могут формировать поля типа объекта OLE, которые используются для хранения рисунков, графиков, таблиц. Если обрабатываемая база данных включает несколько взаимосвязанных таблиц, то необходимо определение ключевого поля в каждой таблице, а также полей, с помощью которых будет организована связь между таблицами. Создание структуры таблицы не связано с заполнением таблиц данными, поэтому эти две операции можно разнести во времени.

Практически все СУБД позволяют вводить и корректировать данные в таблицах двумя способами:

- с помощью предоставляемой по умолчанию стандартной формы в виде таблицы;
- с помощью экранных форм, специально созданных для этого пользователями.



СУБД, работающие с Windows, позволяют вводить в созданные экранные формы рисунки, узоры, кнопки. Возможно построение форм, наиболее удобных для работы пользователя, включающих записи различных связанных таблиц базы данных.

Обрабатывать информацию, содержащуюся в таблицах базы данных, можно путем использования **запросов** или в процессе выполнения специально разработанной программы. Конечный пользователь получает при работе с СУБД удобное средство обработки информации – запрос, который представляет собой инструкцию на отбор записей.

Большинство СУБД разрешают использовать запросы следующих типов:

- запрос-выборка, предназначенный для отбора данных, хранящихся в таблицах, и не изменяющий этих данных;
- запрос-изменение, предназначенный для изменения или перемещения данных; к этому типу запросов относятся запрос на добавление записей и запрос на обновление;

- запрос с параметром, позволяющий определить одно или несколько условий отбора во время выполнения запроса.

Практически любая СУБД позволяет вывести на экран и принтер информацию, содержащуюся в базе данных, из режимов таблицы или формы. Такой порядок вывода данных может использоваться только как черновой вариант, так как позволяет выводить данные только точно в таком же виде, в каком они содержатся в таблице или форме.

Каждый пользователь, работающий с СУБД, имеет возможность использования специальных средств построения отчетов для вывода данных. Используя специальные средства создания отчетов, пользователь получает следующие дополнительные возможности вывода данных:

- включать в отчет выборочную информацию из таблиц базы данных;
- добавлять информацию, не содержащуюся в базе данных;
- при необходимости выводить итоговые данные на основе информации базы данных;
- располагать выводимую в отчете информацию в любом, удобном для пользователя виде (вертикальное или горизонтальное расположение полей);
- включать в отчет информацию из разных связанных таблиц базы данных.