Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

Тамбовский государственный технический университет

Кафедра

**КУРСОВАЯ РАБОТА**

по дисциплине «Информатика»

на тему: “Вычислительные сети. Протоколы”

Выполнил: студент группы

.

Проверил: преподаватель кафедры САПР

.

Тамбов 20

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение | 3 |
| Архитектура | 4 |
| Модель взаимодействия для ЛВС | 7 |
| Протоколы компьютерной сети | 8 |
| Основные типы протоколов | 9 |
| Стандарты протоколов вычислительных сетей | 10 |
| Сетевые протоколы | 11 |
| Протоколы канального уровня | 17 |
| Протоколы межсетевого уровня | 18 |
| Транспортные протоколы | 24 |
| Прикладные протоколы | 27 |
| Сквозные протоколы и шлюзы | 29 |
| Понятие топологии сети | 29 |
| Физическая и логическая топологии сетей | 30 |
| Логические топологии | 32 |
| Подключение к простейшей сети | 37 |
| Среда передачи | 38 |
| Заключение | 40 |
| Список литературы | 41 |

**ВВЕДЕНИЕ**

В современных локальных сетях используются различные технологии подключения, различное оборудование и различные среды передачи данных. Пару тройку лет назад единственным возможным и недорогим вариантом было объединение компьютеров на основе витой пары с пропускной способностью около 10 Мбит/с, позже появились сети, в которых в качестве среды передачи данных стали использовать оптоволокно, начали активно развиваться беспроводные локальные сети, в которых информация передается посредством широкополосных радиосигналов. Развитие сетевых технологий обусловлена совершенствованием самих компьютеров. Аналитиками подсчитано, что мощность процессоров современных ПК удваивается каждые 15-20 месяцев, соответственно, растет и трафик, передаваемый по линиям компьютерных коммуникаций (трафиком называется общий суммарный поток информации через один сетевой пк). Вместе с тем наиболее узкое место в любой распределенной вычислительной системе — это устаревшее оборудование, поскольку уже довольно давно специалистами по компьютерным сетям было сформулировано простая аксиома: максимальная пропускная способность локальной сети равна максимальной пропускной способности ее самого медленного компонента. Вполне справедливо, что эволюция сетевых стандартов во многом определяется ростом информационных потоков и производительностью компьютеров, причем график роста производительности локальных сетей стал похож на экспоненту: сети с пропускной способностью в 100 Мбит/с появились спустя 15 лет после возникновения 10-мегабитных сетей, сетевые системы с пропускной способностью в 1 Гбит/с были разработаны через 5 лет после 100-мегабитных сетей, первые проекты сетей со скоростью передачи данных в 10 Гбит/с родились спустя еще 2 года

**ПОНЯТИЕ АРХИТЕКТУРЫ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ.**

**1. Архитектура.**

*Архитектура*– концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов сложного объекта. Архитектура компьютерной сети определяет основные элементы сети; характер и топологию взаимодействия этих элементов; логическую, функциональную и физическую организацию технических, программных, организационных и информационных средств сети.

ЭТАЛОННЫЕ МОДЕЛИ ВЗАИМОДЕЙСТВИЯ СИСТЕМ

**Модель взаимодействия открытых систем.**

Для определения задач, поставленных перед сложным объектом, а также для выделения главных характеристик и параметров, которыми он должен обладать, создаются общие модели таких объектов. Общая модель вычислительной сети определяет характеристики сети в целом и характеристики и функции входящих в нее основных компонентов.

*Архитектура вычислительной сети* - описание ее общей модели.

Многообразие производителей вычислительных сетей и сетевых программных продуктов поставило проблему объединения сетей различных архитектур. Для ее решения МОС разработала *модель архитектуры открытых систем.*

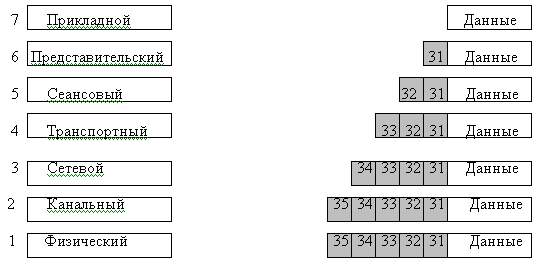
*Открытая система* - система, взаимодействующая с другими системами в соответствии с принятыми стандартами.

Предложенная модель архитектуры открытых систем служит базой для производителей при разработке совместимого сетевого оборудования. Эта модель не является неким физическим телом, отдельные элементы которого можно осязать. Модель представляет собой самые общие рекомендации для построения стандартов совместимых сетевых программных продуктов. Эти рекомендации должны быть реализованы как в аппаратуре, так и в программных средствах вычислительных сетей.

**Рис. 1**. Эталонная модель архитектуры открытых систем.

В настоящее время модель взаимодействия открытых систем (ВОС) является наиболее популярной сетевой архитектурной моделью. Модель рассматривает общие функции, а не специальные решения, поэтому не все реальные сети абсолютно точно ей следуют. Модель взаимодействия открытых систем состоит из семи уровней (рис.1).

* 7-й уровень - *прикладной* - обеспечивает поддержку прикладных процессов конечных пользователей. Этот уровень определяет круг прикладных задач, реализуемых в данной вычислительной сети. Он также содержит все необходимые элементы сервиса для прикладных программ пользователя. На прикладной уровень могут быть вынесены некоторые задачи сетевой операционной системы.
* 6-й уровень - *представительный* - определяет синтаксис данных в модели, т.е. представление данных. Он гарантирует представление данных в кодах и форматах, принятых в данной системе. В некоторых системах этот уровень может быть объединен с прикладным.
* 5-й уровень - *сеансовый* - реализует установление и поддержку сеанса связи между двумя абонентами через коммуникационную сеть. Он позволяет производить обмен данными в режиме, определенном прикладной программой, или предоставляет возможность выбора режима обмена. 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.
* 4-й уровень - *транспортный* - обеспечивает интерфейс между процессами и сетью. Он устанавливает логические каналы между процессами и обеспечивает передачу по этим каналам информационных пакетов, которыми обмениваются процессы. Логические каналы, устанавливаемые транспортным уровнем, называются *транспортными каналами.*
* 3-й уровень - *сетевой* - определяет интерфейс *оконечного оборудования данных* пользователя с сетью коммутации пакетов. Он также отвечает за маршрутизацию пакетов в коммуникационной сети и за связь между сетями - реализует межсетевое взаимодействие.

**Рис. 2.** Обработка сообщений уровнями модели ВОС

*Примечание.* В технике коммуникаций используется термин *оконечное оборудование данных.* Он определяет любую аппаратуру, подключенную к канал; связи, в системе обработки данных (компьютер, терминал, специальная аппаратура).

* 2-й уровень - *канальный* - уровень звена данных - реализует процесс передачи информации по информационному каналу. Информационный канал - логический канал, он устанавливается между двумя ЭВМ, соединенными физическим каналом Канальный уровень обеспечивает управление потоком данных в виде кадров, в которых упаковываются информационные пакеты, обнаруживает ошибки передачи и реализует алгоритм восстановления информации в случае обнаружения сбоев или потерь данных.
* 1-й уровень - *физический* - выполняет все необходимые процедуры в канале связи. Его основная задача - управление аппаратурой передачи данных и подключенным к ней каналом связи.

При передаче информации от прикладного процесса в сеть происходит ее обработка уровнями модели взаимодействия открытых систем (рис. 2). Смысл этой обработки заключается в том, что каждый уровень добавляет к информации 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 сообщения из коммуникационной сети.

Физический уровень заголовка не добавляет. Сообщение, обрамленное заголовками и концевиком, уходит в коммуникационную сеть и поступает на абонентские ЭВМ вычисли тельной сети. Каждая абонентская ЭВМ, принявшая сообщение, дешифрирует адреса и определяет, предназначено ли ей данное сообщение.

При этом в абонентской ЭВМ происходит обратный процесс - чтение и отсечение заголовков уровнями модели взаимодействия открытых систем. 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 процессу, которому она была адресована.

Каждый уровень модели взаимодействия открытых систем реагирует только на свой заголовок.

*Примечание.* На рис. 2 показан процесс прохождения данных через уровни модели. Каждый уровень добавляет свой заголовок - 3.

В чем же основное достоинство семиуровневой модели ВОС? В процессе развития и совершенствования любой системы возникает потребность изменять ее отдельные компоненты. Иногда это вызывает необходимость изменять и другие компоненты, что существенно усложняет и затрудняет процесс модернизации системы.

Здесь и проявляются преимущества семиуровневой модели. Если между уровнями определены однозначно интерфейсы, то изменение одного из уровней не влечет за собой необходимости внесения изменений в другие уровни. Таким образом, существует относительная независимость уровней друг от друга.

Необходимо сделать и еще одно замечание относительно реализации уровней модели ВОС в реальных вычислительных сетях. Функции, описываемые уровнями модели, должны быть реализованы либо в аппаратуре, либо в виде программ.

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

Функции остальных уровней реализуются в виде программных модулей - драйверов.

**Модель взаимодействия для ЛВС**

Для того чтобы учесть требования физической передающей среды, используемой в ЛВС, была произведена некоторая модернизация семиуровневой модели взаимодействия открытых систем для локальных вычислительных сетей. Необходимость такой модернизации была вызвана тем, что для организации взаимодействия абонентских ЭВМ в ЛВС используются специальные методы доступа к физической передающей среде. Верхние уровни модели ВОС не претерпели никаких изменений, а канальный уровень был разбит на два подуровня (рис. 3). Подуровень LLC (Logical Link Control) обеспечивает управление логическим звеном, т.е. выполняет функции собственно канального уровня. Подуровень MAC (Media Access Control) обеспечивает управление доступом к среде.

**Рис. 3.** Эталонная модель для локальных компьютерных сетей

**ПРОТОКОЛЫ КОМПЬЮТЕРНОЙ СЕТИ**

**Понятие протокола**

Как было показано ранее, при обмене информацией в сети каждый уровень модели ВОС реагирует на свой заголовок. Иными словами, происходит взаимодействие между одноименными уровнями модели в различных абонентских ЭВМ. Такое взаимодействие должно выполняться по определенным правилам.

*Протокол* - набор правил, определяющий взаимодействие двух одноименных уровней модели взаимодействия открытых систем в различных абонентских ЭВМ.

Протокол - это не программа. Правила и последовательность выполнения действий при обмене информацией, определенные протоколом, должны быть реализованы в программе. Обычно функции протоколов различных уровней реализуются в драйверах для различных вычислительных сетей.

В соответствии с семиуровневой структурой модели можно говорить о необходимости существования протоколов для каждого уровня.

Концепция открытых систем предусматривает разработку стандартов для протоколов различных уровней. Легче всего поддаются стандартизации протоколы трех нижних уровней модели архитектуры открытых систем, так как они определяют действия и процедуры, свойственные для вычислительных сетей любого класса.

Труднее всего стандартизовать протоколы верхних уровней, особенно прикладного, из-за множественности прикладных задач и в ряде случаев их уникальности. Если по типам структур, методам доступа к физической передающей среде, используемым сетевым технологиям и некоторым другим особенностям можно насчитать примерно десяток различных моделей вычислительных сетей, то по их функциональному назначению пределов не существует.

**Основные типы протоколов**

Проще всего представить особенности сетевых протоколов на примере протоколов канального уровня, которые делятся на две основные группы: байт-ориентированные и 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. Протокол обеспечивает передачу двух типов кадров: управляющих и информационных. В *управляющих кадрах* передаются управляющие и служебные символы, в *информационных -* сообщения (отдельные пакеты, последовательность пакетов). Работа протокола BSC осуществляется в три фазы: установление соединения, поддержание сеанса передачи сообщений, разрыв соединения. Протокол требует на каждый переданный кадр посылки квитанции о результате его приема. Кадры, переданные с ошибкой, передаются повторно. Протокол определяет максимальное число повторных передач.

***Примечание.*** Квитанция представляет собой управляющий кадр, в котором содержится подтверждение приема сообщения (положительная квитанция) или отказ от приема из-за ошибки (отрицательная квитанция).

Передача последующего кадра возможна только тогда, когда получена положительная квитанция на прием предыдущего. Это существенно ограничивает быстродействие протокола и предъявляет высокие требования к качеству канала связи.

*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 в конце - флаг закрывающий.

Бит-ориентированный протокол удобен относительно коммуникационной среды, так как канал связи как раз и ориентирован на передачу последовательности битов. Для ЭВМ он не очень удобен, потому что из поступающей последовательности битов приходится выделять байты для последующей обработки сообщения. Впрочем, учитывая быстродействие ЭВМ, можно считать, что эта операция не окажет существенного влияния на ее производительность. Потенциально бит-ориентированные протоколы являются более скоростными по сравнению с байт-ориентированными, что обусловливает их широкое распространение в современных вычислительных сетях.

Типичным представителем группы бит-ориентированных протоколов являются протокол HDLC (High-level Data Link Control - высший уровень управления каналом связи) и его подмножества. Протокол HDLC управляет информационным каналом с помощью специальных управляющих кадров, в которых передаются команды. Информационные кадры нумеруются. Кроме того, протокол HDLC позволяет без получения положительной квитанции передавать в канал до трех - пяти кадров. Положительная квитанция, полученная, например, на третий кадр, показывает, что два предыдущих приняты без ошибок и необходимо повторить передачу только четвертого и пятого кадров. Такой алгоритм работы и обеспечивает высокое быстродействие протокола.

Из протоколов верхнего уровня модели ВОС следует отметить протокол Х.400 (электронная почта) и FTAM (File Transfer, Access and Management - передача файлов, доступ к файлам и управление файлами).

**Стандарты протоколов вычислительных сетей**

Для протоколов физического уровня стандарты определены рекомендациями МККТТ. Цифровая передача предусматривает использование протоколов Х.21 и Х.21- бис.

Канальный уровень определяют протокол HDLC и его подмножества, а также протокол Х.25/3. Широкое распространение локальных вычислительных сетей потребовало разработки стандартов для этой области. В настоящее время для ЛВС используются стандарты, разработанные Институтом инженеров по электротехнике и радиоэлектронике-ИИЭР(IЕЕЕ- 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);
* 802.3 - случайный метод доступа к среде (CSMA/CD - Carrier 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.

Взаимодействие двух узлов из различных сетей схематически показано на рис. 4. Обмен информацией между одноименными уровнями определяется протоколами выше.

***Примечание****.* Узлы соединены с помощью канала связи. Это та среда, по которой распространяются сообщения от одного узла сети до другого. Пакеты и кадры, о которых шел разговор, в виде последовательности электрических сигналов приходят из одного узла в другой. Взаимодействие одноименных уровней модели показано пунктирными стрелками.

**Рис. 4.** Взаимодействие узлов сети на базе эталонной модели

**Сетевые протоколы**

* Стеки протоколов
* Протоколы канального уровня
* Протоколы межсетевого уровня
* Транспортные протоколы
* Прикладные протоколы

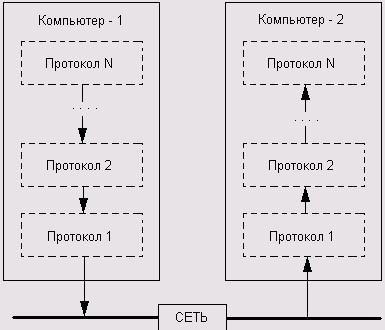
Как уже упоминалось ранее, в локальных сетях могут совместно работать компьютеры разных производителей, оснащенные различным набором устройств и обладающие несхожими техническими характеристиками. На практике это означает, что для обеспечения нормального взаимодействия этих компьютеров необходим некий единый унифицированный стандарт, строго определяющий алгоритм передачи данных в распределенной вычислительной системе. В современных локальных сетях, или, как их принято называть в англоязычных странах, LAN (Local Area Network), роль такого стандарта выполняют сетевые протоколы. Итак, сетевым протоколом, или протоколом передачи данных, называется согласованный и утвержденный стандарт, содержащий описание правил приема и передачи между несколькими компьютерами команд, файлов, иных данных, и служащий для синхронизации работы вычислительных машин в сети.  
Прежде всего следует понимать, что в локальных сетях передача информации осуществляется не только между компьютерами как физическими устройствами, но и между приложениями, обеспечивающими коммуникации на программном уровне. Причем под такими приложениями можно понимать как компоненты операционной системы, организующие взаимодействие с различными устройствами компьютера, так и клиентские приложения, обеспечивающие интерфейс с пользователем. Таким образом, мы постепенно приходим к пониманию многоуровневой структуры сетевых коммуникаций — как минимум, с одной стороны мы имеем дело с аппаратной конфигурацией сети, с другой стороны — с программной. Вместе с тем передача информации между несколькими сетевыми компьютерами — не такая уж простая задача, как это может показаться на первый взгляд. Для того чтобы понять это, достаточно представить себе тот круг проблем, который может возникнуть в процессе приема или трансляции каких-либо данных. В числе таких «неприятностей» можно перечислить аппаратный сбой либо выход из строя одного из обеспечивающих связь устройств, например, сетевой карты или концентратора, сбой прикладного или системного программного обеспечения, возникновение ошибки в самих передаваемых данных, потерю части транслируемой информации или ее искажение. Отсюда следует, что в локальной сети необходимо обеспечить жесткий контроль для отслеживания всех этих ошибок, и более того, организовать четкую работу как аппаратных, так и программных компонентов сети. Возложить все эти задачи на один-единственный протокол практически невозможно. Выход в разделении протоколов на ряд концептуальных уровней, каждый из которых обеспечивает интерфейс между различными модулями программного обеспечения, установленного на работающих в сети компьютерах. Таким образом, механизм передачи какого-либо пакета информации через сеть от клиентской программы, работающей на о/щом компьютере, клиентской программе, работающей на другом компьютере, можно условно представить в виде последовательной пересылки этого пакета сверху вниз от некоего протокола верхнего уровня, обеспечивающего взаимодействие с пользовательским приложением, протоколу нижнего уровня, организующему интерфейс с сетью, его трансляции на компьютер-получатель и обратной передачи протоколу верхнего уровня уже на удаленной машине (рис. 5).

Рис. 5. Концептуальная модель многоуровневой системы протоколов

Согласно такой схеме, каждый из уровней подобной системы обеспечивает собственный набор функций при передаче информации по локальной сети.

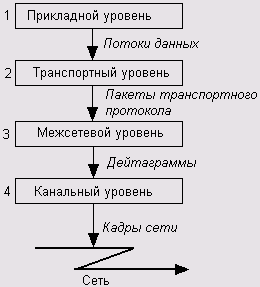
Например, можно предположить, что протокол верхнего уровня, осуществляющий непосредственное взаимодействие с клиентскими программами, транслирует данные протоколу более низкого уровня, «отвечающему» за работу с аппаратными устройствами сети, преобразовывая их в «понятную» для него форму. Тот, в свою очередь, передает их протоколу, осуществляющему непосредственно пересылку информации на другой компьютер. На удаленном компьютере прием данных осуществляет аналогичный протокол «нижнего» уровня и контролирует корректность принятых данных, то есть определяет, следует ли транслировать их протоколу, расположенному выше в иерархической структуре, либо запросить повторную передачу. В этом случае взаимодействие осуществляется только между протоколами нижнего уровня, верхние уровни иерархии в данном процессе не задействованы. В случае если информация была передана без искажений, она транслируется вверх через соседние уровни протоколов до тех пор, пока не достигнет программы-получателя. При этом каждый из уровней не только контролирует правильность трансляции данных на основе анализа содержимого пакета информации, но и определяет дальнейшие действия исходя из сведений о его назначении. Например, один из уровней «отвечает» за выбор устройства, с которого осуществляется получение и через которое передаются данные в сеть, другой «решает», передавать ли информацию дальше по сети, или она предназначена именно этому компьютеру, третий «выбирает» программу, которой адресована принятая информация. Подобный иерархический подход позволяет не только разделить функции между различными модулями сетевого программного обеспечения, что значительно облегчает контроль работы всей системы в целом, но и дает возможность производить коррекцию ошибок на том уровне иерархии, на котором они возникли. Каждую из подобных иерархических систем, включающих определенный набор протоколов различного уровня, принято называть стеком протоколов. Вполне очевидно, что между теорией и практикой, то есть между концептуальной моделью стека протоколов и его практической реализацией существует значительная разница. На практике принято несколько различных вариантов дробления стека протоколов на функциональные уровни, каждый из которых выполняет свой круг задач. Мы остановимся на одном из этих вариантов, который представляется наиболее универсальным. Данная схема включает четыре функциональных уровня, и так же, как и преды дущая диаграмма, описывает не конкретный механизм работы какого-либо стека протоколов, а общую модель, которая поможет лучше понять принцип действия подобных систем (рис. 6).

Рис. 6. Модель реализации стека протоколов

Самый верхний в иерархической системе, прикладной уровень стека протоколов обеспечивает интерфейс с программным обеспечением, организующим работу пользователя в сети. При запуске любой программы, для функционирования которой требуется диалог с сетью, эта программа вызывает соответствующий протокол прикладного уровня. Данный протокол передает программе информацию из сети в доступном для обработки формате, то есть в виде системных сообщений либо в виде потока байтов. В точности таким же образом пользовательские приложения могут получать потоки данных и управляющие сообщения — как от самой операционной системы, так и от других запущенных на компьютере программ. То есть, обобщая, можно сказать, что протокол прикладного уровня выступает в роли своего рода посредника между сетью и программным обеспечением, преобразуя транслируемую через сеть информацию в «понятную» программе-получателю форму. Основная задача протоколов транспортного уровня заключается в осуществлении контроля правильности передачи данных, а также в обеспечении взаимодействия между различными сетевыми приложениями. В частности, получая входящий поток данных, протокол транспортного уровня дробит его на отдельные фрагменты, называемые пакетами, записывает в каждый пакет некоторую дополнительную информацию, например идентификатор программы, для которой предназначены передаваемые данные, и контрольную сумму, необходимую для проверки целостности пакета, и направляет их на смежный уровень для дальнейшей обработки. Помимо этого протоколы транспортного уровня осуществляют управление передачей информации — например, могут запросить у получателя подтверждение доставки пакета и повторно выслать утерянные фрагменты транслируемой последовательности данных. Некоторое недоумение может вызвать то обстоятельство, что протоколы транспортного уровня так же, как и протоколы прикладного уровня, взаимодействуют с сетевыми программами и координируют передачу данных между ними. Эту ситуацию можно прояснить на следующем примере: предположим, на подключенном к сети компьютере запущен почтовый клиент, эксплуатирующий два различных протокола прикладного уровня — РОРЗ ( Post Office Protocol) и SMTP (Simple Mail Transfer Protocol) — и программа загрузки файлов на удаленный сервер — FTP-клиент, работающий с протоколом прикладного уровня FTP (File Transfer Protocol). Все эти протоколы прикладного уровня опираются на один и тот же протокол транспортного уровня — TCP/IP (Transmission Control Protocol/Internet Protocol), который, получая поток данных от вышеуказанных программ, преобразует их в пакеты данных, где присутствует указание на конечное приложение, использующее эту информацию. Из рассмотренного нами примера следует, что данные, приходящие из сети, могут иметь различное назначение, и, соответственно, они обрабатываются различными программами, либо различными модулями одного и того же приложения. Во избежание путаницы при приеме и обработке информации каждая взаимодействующая с сетью программа имеет собственный идентификатор, который позволяет транспортному протоколу направлять данные именно тому приложению, для которого они предназначены. Такие идентификаторы носят название программных портов. В частности, протокол прикладного уровня SMTP, предназначенный для отправки сообщений электронной почты, работает обычно с портом 25, протокол входящей почты РОРЗ — с портом 110, протокол Telnet — с портом 23. Задача перенаправления потоков данных между программными портами лежит па транспортных протоколах.  
На межсетевом уровне реализуется взаимодействие конкретных компьютеров распределенной вычислительной системы, другими словами, осуществляется процесс определения маршрута движения информации внутри локальной сети и выполняется отправка этой информации конкретному адресату. Данный процесс принято называть маршрутизацией. Получая пакет данных от протокола транспортного уровня вместе с запросом на его передачу и указанием получателя, протокол межсетевого уровня выясняет, на какой компьютер следует передать информацию, находится ли этот компьютер в пределах данного сегмента локальной сети или на пути к нему расположен шлюз, после чего трансформирует пакет в дейтаграмму — специальный фрагмент информации, передаваемый через сеть независимо от других аналогичных фрагментов, без образования виртуального канала (специально сконфигурированной среды для двустороннего обмена данными между несколькими устройствами) и подтверждения приема. В заголовок дейтаграммы записывается адрес компьютера-получателя пересылаемых данных и сведения о маршруте следования дейтаграммы. После чего она передается на канальный уровень.

*Шлюз* — это программа, при помощи которой можно передавать информацию между двумя сетевыми системами, использующими различные протоколы обмена данными.

Получая дейтаграмму, протокол межсетевого уровня определяет правильность ее приема, после чего выясняет, адресована ли она локальному компьютеру, или же ее следует направить по сети дальше. В случае, если дальнейшей пересылки не требуется, протокол межсетевого уровня удаляет заголовок дейтаграммы, вычисляет, какой из транспортных протоколов данного компьютера будет обрабатывать полученную информацию, трансформирует ее в соответствующий пакет и передает на транспортный уровень. Проиллюстрировать этот на первый взгляд сложный механизм можно простым примером. Предположим, на пеком компьютере одновременно используется два различных транспортных протокола: TCP/IP — для соединения с Интернетом и NetBEUI (NetBIOS Extended User Interface) для работы в локальной сети. В этом случае данные, обрабатываемые на транспортном уровне, будут для этих протоколов различны, однако на межсетевом уровне информация будет передаваться посредством дейтаграмм одного и того же формата. На канальном уровне осуществляется преобразование дейтаграмм в соответствующий сигнал, который через коммуникационное устройство транслируется по сети. В самом простом случае, когда компьютер напрямую подключен к локальной сети того или иного стандарта посредством сетевого адаптера, роль протокола канального уровня играет драйвер этого адаптера, непосредственно реализующий интерфейс с сетью. В более сложных ситуациях на канальном уровне могут работать сразу несколько специализированных протоколов, каждый из которых выполняет собственный набор функций.

# Протоколы канального уровня

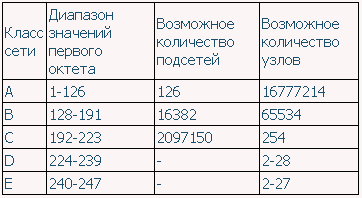
Протоколы, обеспечивающие взаимодействие компьютера с сетью на самом низком, аппаратном уровне, во многом определяют топологию локальной сети, а также ее внутреннюю архитектуру. В настоящее время на практике достаточно часто применяется несколько различных стандартов построения локальных сетей, наиболее распространенными среди которых являются технологии Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI) и ArcNet. На сегодняшний день локальные сети, построенные на основе стандарта Ethernet, являются наиболее популярными как в нашей стране, так и во всем мире. На долю сетей Ethernet приходится почти девяносто процентов всех малых и домашних локальных сетей, что не удивительно, поскольку именно эта технология позволяет строить простые и удобные в эксплуатации и настройке локальные сети с минимумом затрат. Именно поэтому в качестве основного рассматриваемого нами стандарта будет принята именно технология Ethernet. Протоколы канального уровня поддержки Ethernet, как правило, встроены в оборудование, обеспечивающее подключение компьютера к локальной сети на физическом уровне. Стандарт Ethernet является широковещательным, то есть каждый подключенный к сети компьютер принимает всю следующую через его сетевой сегмент информацию — как предназначенную именно для этого компьютера, так и данные, направляемые на другую машину. Во всех сетях Ethernet применяется один и тот же алгоритм разделения среды передачи информации — множественный доступ с контролем несущей и обнаружением конфликтов (Carrier Sense Multiple Access with Collision Detection, CSMA/CD).  
В рамках технологии Ethernet сегодня различается несколько стандартов организации сетевых коммуникаций, определяющих пропускную способность канала связи и максимально допустимую длину одного сегмента сети, то есть расстояние между двумя подключенными к сети устройствами. Об этих стандартах мы побеседуем в следующей главе, посвященной изучению сетевого оборудования, пока же необходимо отметить, что в рамках стандарта Ethernet применяется, как правило, одна из двух различных топологий: конфигурация сети с общей шиной или звездообразная архитектура.

# Протоколы межсетевого уровня

Протоколы уровня межсетевого взаимодействия, как уже упоминалось ранее, предназначены для определения маршрутов следования информации в локальной сети, приема и передачи дейтаграмм, а также для трансляции принятых данных протоколам более высокого уровня, если эти данные предназначены для обработки на локальном компьютере. К протоколам межсетевого уровня принято относить протоколы маршрутизации, такие как RIP (Routing Internet Protocol) и OSPF (Open Shortest Path First), а также протокол контроля и управления передачей данных ICMP (Internet Control Message Protocol). Но вместе с тем одним из самых известных протоколов межсетевого уровня является протокол IP.

## Протокол IP

Протокол IP (Internet Protocol) используется как в глобальных распределенных системах, например в сети Интернет, так и в локальных сетях. Впервые протокол IP применялся еще в сети ArpaNet, являвшейся предтечей современного Интернета, и с тех пор он уверенно удерживает позиции в качестве одного из наиболее распространенных и популярных протоколов межсетевого уровня. Поскольку межсетевой протокол IP является универсальным стандартом, он нередко применяется в так называемых составных сетях, то есть сетях, использующих различные технологии передачи данных и соединяемых между собой посредством шлюзов. Этот же протокол «отвечает» за адресацию при передаче информации в сети. Как осуществляется эта адресация? Каждый человек, живущий на Земле, имеет адрес, по которому его в случае необходимости можно разыскать. Думаю, ни у кого не вызовет удивления то, что каждая работающая в Интернете или локальной сети машина также имеет свой уникальный адрес. Адреса в компьютерных сетях разительно отличаются от привычных нам почтовых. Боюсь, совершенно бесполезно писать на отправляемом вами в Сеть пакете информации нечто вроде «Компьютеру Intel i7, админу, Набережная 114, Тамбов, Россия». Увидев такую надпись, ваша персоналка в лучшем случае фундаментально зависнет. Но если вы укажете компьютеру в качестве адреса нечто вроде 195.85.102.14, машина вас прекрасно поймет.Именно стандарт IP подразумевает подобную запись адресов подключенных к сети компьютеров. Такая запись носит название IP-адрес. Из приведенного примера видно, что IP-адрес состоит из четырех десятичных идентификаторов, или октетов, по одному байту каждый, разделенных точкой. Левый октет указывает тип локальной интрасети (под термином «интрасеть» (intranet) здесь понимается частная корпоративная или домашняя локальная сеть, имеющая подключение к Интернету), в которой находится искомый компьютер. В рамках данного стандарта различается несколько подвидов интрасетей, определяемых значением первого октета. Это значение характеризует максимально возможное количество подсетей и узлов, которые может включать такая сеть. В табл. 7 приведено соответствие классов сетей значению первого октета IP-адреса.

Таблица 7. Соответствие классов сетей значению первого октета IP-адреса

Адреса класса А используются в крупных сетях общего пользования, поскольку позволяют создавать системы с большим количеством узлов. Адреса класса В, как правило, применяют в корпоративных сетях средних размеров, адреса класса С — в локальных сетях небольших предприятий. Для обращения к группам машин предназначены широковещательные адреса класса D, адреса класса Е пока не используются: предполагается, что со временем они будут задействованы с целью расширения стандарта. Значение первого октета 127 зарезервировано для служебных целей, в основном для тестирования сетевого оборудования, поскольку IP-пакеты, направленные на такой адрес, не передаются в сеть, а ретранслируются обратно управляющей надстройке сетевого программного обеспечения как только что принятые. Кроме того, существует набор так называемых «выделенных» IP-адресов, имеющих особое значение. Эти адреса приведены в табл. 8.

Таблица 8. Значение выделенных IP-адресов

**Примечание:** Хостом принято называть любой подключенный к Интернету компьютер независимо от его назначения.

Небольшие локальные сети могут соединяться между собой, образуя более сложные и разветвленные структуры. Например, локальная сеть предприятия может состоять из сети административного корпуса и сети производственного отдела, сеть административного корпуса, в свою очередь, может включать в себя сеть бухгалтерии, планово-экономического отдела и отдела маркетинга. В приведенном выше примере сеть более низкого уровня является подсетью системы более высокого уровня, то есть локальная сеть бухгалтерии — подсеть для сети административного корпуса, а та, в свою очередь, — подсеть для сети всего предприятия в целом. Последний (правый) идентификатор IP-адреса обозначает номер компьютера в данной локальной сети. Все, что расположено между правым и левым октетами в такой записи, — номера подсетей более низкого уровня. Например, есть некий адрес в Интернете, на который хотим отправить пакет с набором свеженьких анекдотов. В качестве примера возьмем тот же IP-адрес— 195.85.102.14. Итак, мы отправляем пакет в 195-ю подсеть сети Интернет, которая, как видно из значения первого октета, относится к классу С. Допустим, 195-я сеть включает в себя еще 902 подсети, но наш пакет высылается в 85-ю. Она содержит 250 подсетей  
более низкого порядка, но нам нужна 102-я. Ну и, наконец, к 102-й сети подключено 40 компьютеров. Исходя из рассматриваемого нами адреса, подборку анекдотов получит машина, имеющая в этой сетевой системе номер 14. Становится очевидно, что IP-адрес каждого компьютера, работающего как в локальной сети, так и в глобальных вычислительных системах, должен быть уникален.  
Большинство админов небольших локальных сетей, насчитывающих 5—10 компьютеров, назначают IP-адреса подключенным к сети машинам самостоятельно, исходя из описанных выше правил адресации в IP-сетях. Тацой подход вполне имеет право на жизнь, но вместе с тем произвольное назначение IP-адресов может стать проблемой, если в будущем такая сеть будет соединена с другими локальными сетями или в ней будет организовано прямое подключение к Интернету. В данном случае случайное совпадение нескольких IP-адресов может привести к весьма неприятным последствиям, например к ошибкам в маршрутизации передаваемых по сети данных или отказу в работе всей сети в целом.  
Небольшие локальные сети, насчитывающие ограниченное количество компьютеров, должны запрашивать для регистрации адреса класса С. При этом каждой из таких сетей назначаются только два первых октета IP-адреса, например 197.112.Х.Х, на практике это означает, что администратор данной сети может создавать подсети и назначать номера узлов в рамках каждой из них произвольно, исходя из собственных потребностей.  
Большие локальные сети, использующие в качестве базового межсетевой протокол IP, нередко применяют чрезвычайно удобный способ структуризации всей сетевой системы путем разделения общей IP-сети на подсети. Например, если вся сеть предприятия состоит из ряда объединенных вместе локальных сетей Ethernet, то в ней может быть выделено несколько структурных составляющих, то есть подсетей, отличающихся значением третьего октета IP-адреса. Как правило, в качестве каждой из подсетей используется физическая сеть какого-либо отдела фирмы, скажем, сеть Ethernet, объединяющая все компьютеры бухгалтерии. Такой подход, во-первых, позволяет  
излишне не расходовать IP-адреса, а во-вторых, предоставляет определенные удобства с точки зрения администрирования: например, администратор может открыть доступ к Интернету только для одной из вверенных ему подсетей или на время отключить одну из подсетей от локальной сети предприятия. Кроме того, в случае если сетевой администратор решит, что третий октет IP-адреса описывает номер подсети, а четвертый — номер узла в ней, то такая информация записывается в локальных таблицах маршрутизации сети вашего предприятия и не видна извне. Другими словами, данный подход обеспечивает большую безопасность.  
Для того чтобы программное обеспечение могло автоматически выделять номера конкретных компьютеров из используемых в данной сетевой системе IP-адресов, применяются так называемые маски подсети. Принцип, по которому осуществляется распознавание номеров узлов в составе IP-адреса, достаточно прост: биты маски подсети, обозначающие номер самой IP-сети, должны быть равны единице, а биты, определяющие номер узла, — нулю. Именно поэтому в большинстве локальных IP-сетей класса С в качестве маски подсети принято значение 255.255.255.0: при такой конфигурации в состав общей сети может быть включено до 256 подсетей, в каждой из которых работает до 254 компьютеров. В ряде случаев это значение может изменяться, например, если возникла необходимость использовать в составе сети количество подсетей большее, чем 256, можно использовать маску подсети формата 255.255.255.195. В этой конфигурации сеть может включать до 1024 подсетей, максимальное число компьютеров в каждой из которых не должно превышать 60.  
В локальных сетях, работающих под управлением межсетевого протокола IP, помимо обозначения IP-адресов входящих в сеть узлов принято также символьное обозначение компьютеров: например, компьютер с адресом 192.112.85.7 может иметь сетевое имя Localhost. Таблица соответствий IP-адресов символьным именам узлов содержится в специальном файле hosts, хранящемся в одной из системных папок; в частности, в операционной системе Microsoft Windows XP, Vista этот файл можно отыскать в папке \Windows\system32\drivers\etc\. Синтаксис записи таблицы сопоставлений имен узлов локальной сети IP-адресам достаточно прост: каждый элемент таблицы должен быть расположен в новой строке, IP-адрес располагается в первом столбце, а за ним следует имя компьютера, при этом IP-адрес и имя должны быть разделены как минимум одним пробелом. Каждая из строк таблицы может включать произвольный комментарий, обозначаемый символом #. Пример файла hosts приведен ниже:

192.112.85.7 localhost # этот компьютер  
192.112.85.1 server # сервер сети  
192.112.85.2 director # компьютер приемной директора   
192.112.85.5 admin # компьютер системного администратора

Как правило, файл hosts создается для какой-либо конкретной локальной сети, и его копия хранится на каждом из подключенных к ней компьютеров. В случае, если один из узлов сети имеет несколько IP-адресов, то в таблице соответствий обычно указывается лишь один из них, вне зависимости от того, какой из адресов реально используется. При получении из сети IP-пакета, предназначенного для данного компьютера, протокол IP сверится с таблицей маршрутизации и на основе анализа заголовка IP-пакета автоматически опознает любой из IP-адресов, назначенных данному узлу.  
Помимо отдельных узлов сети собственные символьные имена могут иметь также входящие в локальную сеть подсети. Таблица соответствий IP-адресов именам подсетей содержится в файле networks, хранящемся в той же папке, что и файл hosts. Синтаксис записи данной таблицы сопоставлений несколько отличается от предыдущего, и в общем виде выглядит следующим образом: <сетевяе имя> <номер сети> [псевдонимы...] [#<конментарий>]  
где сетевое имя — имя, назначенное каждой подсети, номер сети — часть IP-адреса подсети (за исключением номеров более мелких подсетей, входящих в данную подсеть, и номеров узлов), псевдонимы — необязательный параметр, указывающий на возможные синонимы имен подсетей: они используется в случае, если какая-либо подсеть имеет несколько различных символьных имен; и, наконец, комментарий — произвольный комментарий, поясняющий смысл каждой записи. Пример файла networks приведен ниже:

loopback 127  
marketing 192.112.85 # отдел маркетинга  
buhgalteria 192.112.81 # бухгалтерия  
workshop 192.112.80 # сеть производственного цеха  
workgroup 192.112.10 localnetwork # основная рабочая группа

Обратите внимание на то обстоятельство, что адреса, начинающиеся на 127, являются зарезервированными для протокола IP, а подсеть с адресом 192.112.10 в нашем примере имеет два символьных имени, используемых совместно.  
Файлы hosts и networks не оказывают непосредственного влияния на принципиальный механизм работы протокола IP и используются в основном прикладными программами, однако они существенно облегчают настройку и администрирование локальной сети.

## Протокол IPX

Протокол IPX (Internet Packet Exchange) является межсетевым протоколом, используемым в локальных сетях, узлы которых работают под управлением операционных систем семейства Nowell Netware. Данный протокол обеспечивает передачу дейтаграмм в таких сетях без организации логического соединения — постоянного двустороннего обмена данными между двумя узлами сети, которое организуется протоколом транспортного уровня. Разработанный на основе технологий Nowell, этот некогда популярный протокол в силу несовместимости с чрезвычайно распространенным стеком протоколов TCP/IP в настоящее время медленно, но верно утрачивает свои позиции.  
Как и межсетевой протокол IP, IPX способен поддерживать широковещательную передачу данных посредством дейтаграмм длиной до 576 байт, 30 из которых занимает заголовок пакета. В сетях IPX используются составные адреса узлов, состоящих из номера сети, адреса узла и адреса прикладной программы, для которой предназначен передаваемый пакет информации, который также носит наименование гнезда или сокета. Для обеспечения обмена данными между несколькими сетевыми приложениями в многозадачной среде на узле, работающем под управлением протокола IPX, должно быть одновременно открыто несколько сокетов.  
Поскольку в процессе трансляции данных протокол IPX не запрашивает подтверждения получения дейтаграмм, доставка данных в таких сетях не гарантируется, и потому функции контроля над передачей информации возлагаются на сетевое программное обеспечение. Фактически IPX обеспечивает только инкапсуляцию транслируемых по сети потоков данных в дейтаграммы, их маршрутизацию и передачу пакетов протоколам более высокого уровня.  
Протоколам канального уровня IPX передает пакеты данных, имеющие следующую логическую структуру:

* контрольная сумма, предназначенная для определения целостности передаваемого пакета (2 байта);
* указание на длину пакета (2 байта);
* данные управления транспортом (1 байт);
* адрес сети назначения (4 байта);
* адрес узла назначения (6 байт);
* номер сокета назначения (2 байта);
* адрес сети-отправителя (4 байта);
* адрес узла-отправителя (6 байт);
* номер сокета-отправителя (2 байта);
* передаваемая информация (0-546 байт).

Протоколы канального уровня размещают этот пакет внутри кадра сети и передают его в распределенную вычислительную систему.

# Транспортные протоколы

Как уже упоминалось ранее, протоколы транспортного уровня обеспечивают контроль над передачей данных между межсетевыми протоколами и приложениями уровня операционной системы. В настоящее время в локальных сетях наиболее распространено несколько разновидностей транспортных протоколов.

## Протокол TCP

Протокол IP позволяет только транслировать данные. Для того чтобы управлять этим процессом, служит протокол TCP (Transmission Control Protocol), опирающийся на возможности протокола IP. Как же контролируется передача информации?  
Положим, вы хотите переслать по почте вашему другу толстый журнал, не потратив при этом денег на отправку бандероли. Как решить эту проблему, если почта отказывается принимать письма, содержащие больше нескольких бумажных листов? Выход простой: разделить журнал на страницы и отправлять их отдельными письмами. По номерам страниц ваш друг сможет собрать журнал целиком. Приблизительно таким же способом работает протокол TCP. Он дробит информацию на несколько частей, присваивает каждой части номер, по которому данные впоследствии можно будет соединить воедино, добавляет к ней «служебную» информацию и укладывает все это в отдельный «IP-конверт». Далее этот «конверт» отправляется по сети — ведь протокол межсетевого уровня умеет обрабатывать подобную информацию. Поскольку в такой схеме протоколы TCP и IP тесно связаны, их часто объединяют в одно понятие: TCP/IP. Размер передаваемых в Интернете TCP/IP-пакетов составляет, как правило, от 1 до 1500 байт, что связано с техническими характеристиками сети.  
Наверняка, пользуясь услугами обычной почтовой связи, вы сталкивались с тем, что обычные письма, посылки и иные почтовые отправления теряются и приходят совсем не туда, куда нужно. Те же проблемы характерны и для локальных сетей. На почте такие неприятные ситуации решают руководители почтовых отделений, а в сетевых системах этим занимается протокол TCP. Если какой-либо пакет данных не был доставлен получателю вовремя, TCP повторяет пересылку до тех пор, пока информация не будет принята корректно и в полном объеме.  
В действительности данные, передаваемые по электронным сетям, не только теряются, но зачастую искажаются из-за помех на линиях связи. Встроенные в TCP алгоритмы контроля корректности передачи данных решают и эту проблему. Одним из самых известных механизмов контроля правильности пересылки информации является метод, согласно которому в заголовок каждого передаваемого пакета записывается некая контрольная сумма, вычисленная компьютером-отправителем. Компьютер-получатель по аналогичной системе вычисляет контрольную сумму и сравнивает ее с числом, имеющимся в заголовке пакета. Если цифры не совпадают, TCP пытается повторить передачу.  
Следует отметить также, что при отправке информационных пакетов протокол TCP требует от компьютера-получателя подтверждения приема информации. Это организуется путем создания временных задержек при приеме-передаче — тайм-аутов, или ожиданий. Тем временем отправитель продолжает пересылать данные. Образуется некий объем уже переданных, но еще не подтвержденных данных. Иными словами, TCP организует двунаправленный обмен информацией, что обеспечивает более высокую скорость ее трансляции.  
При соединении двух компьютеров их модули TCP следят за состоянием связи. При этом само соединение, посредством которого осуществляется обмен данными, носит название виртуального или логического канала.  
Фактически протокол TCP является неотъемлемой частью стека протоколов TCP/IP, и именно с его помощью реализуются все функции контроля над передачей информации по сети, а также задачи ее распределения между клиентскими приложениями.

## Протокол SPX

В точности так же, как протокол TCP для IP-сетей, для сетей, построенных на базе межсетевого протокола IPX, транспортным протоколом служит специальный протокол SPX (Sequenced Pocket eXchange). В таких локальных сетях протокол SPX выполняет следующий набор функций:

* инициализация соединения;
* организация виртуального канала связи (логического соединения);
* проверка состояния канала;
* контроль передачи данных;
* разрыв соединения.

Поскольку транспортный протокол SPX и межсетевой протокол IPX тесно связаны между собой, их нередко объединяют в общее понятие — семейство протоколов IPX/SPX. Поддержка данного семейства протоколов реализована не только в операционных системах семейства Nowell Netware, но и в ОС Microsoft Windows 2000/XP/Vista, Unix/Linux и OS/2.

**Протоколы NetBIOS/NetBEUI**

Разработанный компанией IBM транспортный протокол NetBIOS (Network Basic Input/Output System) является базовым протоколом для локальных  
сетей, работающих под управлением операционных систем семейств Nowell Netware и OS/2, однако его поддержка реализована также и в ОС Microsoft Windows, и в некоторых реализациях Unix-совместимых операционных систем. Фактически можно сказать, что данный протокол работает сразу на нескольких логических уровнях стека протоколов: на транспортном уровне он организует интерфейс между сетевыми приложениями в качестве надстройки над протоколами IPX/SPX, на межсетевом — управляет маршрутизацией дейтаграмм, на канальном уровне — организует обмен сообщениями между различными узлами сети.  
В отличие от других протоколов, NetBIOS осуществляет адресацию в локальных сетях на основе уникальных имен узлов и практически не требует настройки, благодаря чему остается весьма привлекательным для системных администраторов, управляющих сетями с небольшим числом компьютеров. В качестве имен хостов протоколом NetBIOS используются значащие последовательности длиной в 16 байт, то есть каждый узел сети имеет собственное уникальное имя (permanent name), которое образуется из сетевого адреса машины с добавлением десяти служебных байтов. Кроме этого, каждый компьютер в сетях NetBIOS имеет произвольное символьное имя, равно как произвольные имена могут иметь логические рабочие группы, объединяющие несколько работающих совместно узлов — такие имена могут назначаться и удаляться по желанию системного администратора. Имена узлов служат для идентификации компьютера в сети, имена рабочих групп могут служить, в частности, для отправки данных нескольким компьютерам группы или для обращения к целому ряду сетевых узлов одновременно.  
При каждом подключении к распределенной вычислительной системе протокол NetBIOS осуществляет опрос локальной сети для проверки уникальности имени узла; поскольку несколько узлов сети могут иметь идентичные групповые имена, определение уникальности группового имени не производится.  
Специально для локальных сетей, работающих на базе стандарта NetBIOS, корпорацией IBM был разработан расширенный интерфейс для этого протокола, который получил название NetBEUI (NetBIOS Extended User Interface). Этот протокол рассчитан на поддержку небольших локальных сетей, включающих не более 150-200 машин, и по причине того, что данный протокол может использоваться только в отдельных сегментах локальных сетей (пакеты NetBEUI не могут транслироваться через мосты — устройства, соединяющие несколько локальных сетей, нередко использующих различную среду передачи данных или различную топологию), этот стандарт считается устаревшим и более не поддерживается операционной системой Microsoft Windows XP и последующими версиями.

# Прикладные протоколы

Протоколы прикладного уровня служат для передачи информации конкретным клиентским приложениям, запущенным на сетевом компьютере. В IP-сетях протоколы прикладного уровня опираются на стандарт TCP и выполняют ряд специализированных функций, предоставляя пользовательским программам данные строго определенного назначения. Ниже мы кратко рассмотрим несколько прикладных протоколов стека TCP/IP.

## Протокол FTP

Как следует из названия, протокол FTP (File Transfer Protocol) предназначен для передачи файлов через Интернет. Именно на базе этого протокола реализованы процедуры загрузки и выгрузки файлов на удаленных узлах Всемирной Сети. FTP позволяет переносить с машины па машину не только файлы, но и целые папки, включающие поддиректории на любую глубину вложений. Осуществляется это путем обращения к системе команд FTP, описывающих ряд встроенных функций данного протокола.

## Протоколы РОРЗ и SMTP

Прикладные протоколы, используемые при работе с электронной почтой, называются SMTP (Simple Mail Transfer Protocol) и РОРЗ (Post Office Protocol), первый «отвечает» за отправку исходящей корреспонденции, второй — за доставку входящей.  
В функции этих протоколов входит организация доставки сообщений e-mail и передача их почтовому клиенту. Помимо этого, протокол SMTP позволяет отправлять несколько сообщений в адрес одного получателя, организовывать промежуточное хранение сообщений, копировать одно сообщение для отправки нескольким адресатам. И РОРЗ, и SMTP обладают встроенными механизмами распознавания адресов электронной почты, а также специальными модулями повышения надежности доставки сообщений.

## Протокол HTTP

Протокол HTTP (Hyper Text Transfer Protocol) обеспечивает передачу с удаленных серверов на локальный компьютер документов, содержащих код разметки гипертекста, написанный на языке HTML или XML, то есть веб-страниц. Данный прикладной протокол ориентирован прежде всего на предоставление информации программам просмотра веб-страниц, веб-браузерам, наиболее известными из которых являются такие приложения, как Microsoft Internet Explorer и Netscape Communicator.  
Именно с использованием протокола HTTP организуется отправка запросов удаленным http-серверам сети Интернет и обработка их откликов; помимо  
этого HTTP позволяет использовать для вызова ресурсов Всемирной сети адреса стандарта доменной системы имен (DNS, Domain Name System), то есть обозначения, называемые URL (Uniform Resource Locator) вида http:/ /www.domain.zone/page (l).

## Протокол TELNET

Протокол TELNET предназначен для организации терминального доступа к удаленному узлу посредством обмена командами в символьном формате ASCII. Как правило, для работы с сервером по протоколу TELNET на стороне клиента должна быть установлена специальная программа, называемая telnet-клиентом, которая, установив связь с удаленным узлом, открывает в своем окне системную консоль операционной оболочки сервера. После этого вы можете управлять серверным компьютером в режиме терминала, как своим собственным (естественно, в очерченных администратором рамках). Например, вы получите возможность изменять, удалять, создавать, редактировать файлы и папки, а также запускать на исполнение программы на диске серверной машины, сможете просматривать содержимое папок других пользователей. Какую бы операционную систему вы ни использовали, протокол Telnet позволит вам общаться с удаленной машиной «на равных». Например, вы без труда сможете открыть сеанс UNIX на компьютере, работающем под управлением MS Windows.

## Протокол UDP

Прикладной протокол передачи данных UDP (User Datagram Protocol) используется на медленных линиях для трансляции информации как дейтаграмм.  
Дейтаграмма содержит полный комплекс данных, необходимых для ее отсылки и получения. При передаче дейтаграмм компьютеры не занимаются обеспечением стабильности связи, поэтому следует принимать особые меры для обеспечения надежности.  
Схема обработки информации протоколом UDP, в принципе, такая же, как и в случае с TCP, но с одним отличием: UDP всегда дробит информацию по одному и тому же алгоритму, строго определенным образом. Для осуществления связи с использованием протокола UDP применяется система отклика: получив UDP-пакет, компьютер отсылает отправителю заранее обусловленный сигнал. Если отправитель ожидает сигнал слишком долго, он просто повторяет передачу.  
На первый взгляд может показаться, что протокол UDP состоит сплошь из одних недостатков, однако есть в нем и одно существенное достоинство: прикладные интернет-программы работают с UDP в два раза быстрее, чем с его более высокотехнологичным собратом TCP.

# Сквозные протоколы и шлюзы

Интернет — это единая глобальная структура, объединяющая на сегодня около 13 000 различных локальных сетей, не считая отдельных пользователей.

Раньше все сети, входившие в состав Интернета, использовали сетевой протокол IP. Однако настал момент, когда пользователи локальных систем, не использующих IP, тоже попросились в лоно Интернета. Так появились шлюзы.  
Поначалу через шлюзы транслировалась только электронная почта, но вскоре пользователям и этого стало мало.

Теперь посредством шлюзов можно передавать любую информацию — и графику, и гипертекст, и музыку, и даже видео.

Информация, пересылаемая через такие сети другим сетевым системам, транслируется с помощью сквозного протокола, обеспечивающего беспрепятственное прохождение IP-пакетов через не IP-сеть.

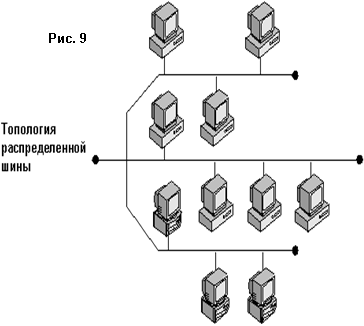
**2. Понятие топологии сети.**

Термин "сетевая топология" описывает возможные конфигурации компьютерных сетей. Специфика сетевых технологий состоит в необходимости строгого согласования всех характеристик аппаратных и программных сетевых средств для успешного обмена данными. При этом существующие аппаратные средства способны обеспечивать различные возможности (скорость, надежность и т.п.) по передаче данных в зависимости от способа использования этих устройств. Для учета всех этих особенностей режимов работы оборудования и было введено понятие "сетевая топология". В настоящее время для описания конфигурации сети используют два вида топологий: физическую и логическую.

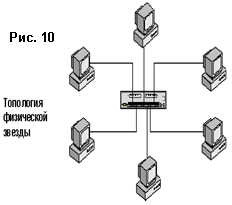
**3. Физическая и логическая топологии сетей.**

 Физическая топология описывает реально использующиеся способы организации физических соединений различного сетевого оборудования (использующиеся кабели, разъемы и способы подключения сетевого оборудования). Физические топологии различаются по стоимости и функциональности. Ниже мы приведем описание трех наиболее часто использующихся физических топологий с указанием их преимуществ и недостатков.

**Физическая шина (Physical Bus)**

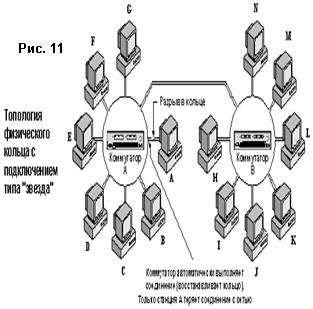
Самая простая форма топологии физической шины представляет собой один основной кабель, оконцованный с обеих сторон специальными типами разъемов – терминаторами. При создании такой сети основной кабель прокладывают последовательно от одного сетевого устройства к другому. Сами устройства подключаются к основному кабелю с использованием подводящих кабелей и T-образных разъемов. Более сложной формой топологии физической шины является "распределенная шина" (чаще называется "древовидная топология"). В такой топологии основной кабель, начинаясь из одной точки, называемой "корнем" (root), разветвляется в различных направлениях определяемых реальным физическим местоположением сетевых устройств. В отличие от описанной выше топологии, в топологии "распределенная шина" основной кабель имеет более двух окончаний. Разветвление кабеля осуществляется с использованием специальных разъемов. Пример такой топологии приведен на рисунке 9.

**Физическая звезда (Physical Star)**

Самая простая форма топологии "физическая звезда" состоит из множества кабелей (по одному на каждое подключаемое сетевое устройство) подключенных к одному, центральному устройству. Это центральное устройство называют концентратором. Примером топологии физической звезды является технология Ethernet 10Base-T или Ethernet 100Base-T. В таких сетях каждое сетевое устройство подключается к концентратору с использованием кабеля типа "витая пара". В случае использования простой топологии "физическая звезда" реальные пути движения сигналов могут не соответствовать форме звезды. Единственная характеристика, описываемая топологией "физическая звезда" – это способ физического соединения сетевых устройств. Пример самой простой топологии "физическая звезда" приведен на рисунке 10.

**Физическое кольцо с подключением типа "звезда" (Physical Star-Wired Ring)**

В этой топологии все сетевые устройства подключаются к центральному концентратору так же, как это происходит при использовании топологии "физическая звезда". Но каждый из концентраторов внутри себя организовывает физические соединения, обеспечивающие построение единого физического кольца. При использовании нескольких концентраторов, кольцо в каждом из концентраторов размыкается, а сами концентраторы подключаются друг к другу с использованием двух кабелей, организуя физическое замыкание кольца. Топология физического кольца используется в сетях IBM Token-Ring. Пример описанной топологии приведен на рис. В этой топологии все концентраторы являются "интеллектуальными" устройствами. При возникновении разрыва физического кольца в любой точке сети концентратор автоматически обнаруживает разрыв и восстанавливает кольцо путем замыкания внутри себя соответствующих портов. На рисунке 11 показан пример такого восстановления кольца (концентратор А).

В настоящее время наибольшей популярностью пользуется звездообразная топология, поскольку она обеспечивает самый простой способ подключения новых устройств в сеть. В большинстве случаев включение нового устройства в сеть заключается лишь в прокладке отрезка кабеля, соединяющего подключаемое сетевое устройство с концентратором.

**Логические топологии**

Логическая топология определяет реальные пути движения сигналов при передаче данных по используемой физической топологии. Таким образом, логическая топология описывает пути передачи потоков данных между сетевыми устройствами. Она определяет правила передачи данных в существующей среде передачи с гарантированием отсутствия помех влияющих на корректность передачи данных. Поскольку логическая топология описывает путь и направление передачи данных, то она тесно связана с уровнем MAC (Media Access Control) модели OSI (подуровень канального уровня). Для каждой из существующих логических топологий существуют методы контроля доступа к среде передачи данных (MAC) позволяющие осуществлять мониторинг и контроль процесса передачи данных. Эти методы будут обсуждаться вместе с соответствующей им топологией. В настоящее время существует три базовые логические топологии: "логическая шина", "логическое кольцо" и "логическая звезда" (коммутация). Каждая из этих топологий обеспечивает преимущества в зависимости от способов использования. Используя рассмотренные ранее рисунки, посвященные физическим топологиям, всегда помните, что логическая топология определяет направление и способ передачи, а не схему соединения физических проводников и устройств.

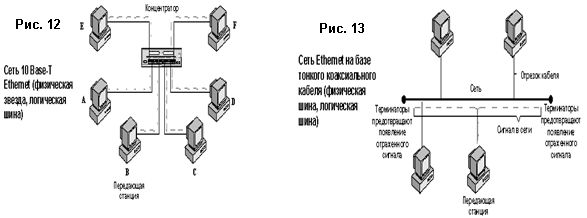
**Логическая шина**

В топологии "логическая шина" последовательности данных, называемые "кадрами" (frames), в виде сигналов распространяются одновременно во всех направлениях по существующей среде передачи. Каждая станция в сети проверяет каждый кадр данных для определения того, кому адресованы эти данные. Когда сигнал достигает конца среды передачи, он автоматически гасится (удаляется из среды передачи) соответствующими устройствами, называемыми "терминаторами" (terminators). Такое уничтожение сигнала на концах среды передачи данных предотвращает отражение сигнала и его обратное поступление в среду передачи. Если бы терминаторов не существовало, то отраженный сигнал накладывался бы на полезный и искажал его.

В топологии "логическая шина" среда передачи совместно и одновременно используется всеми устройствами передачи данных. Для предотвращения помех при попытках одновременной передачи данных несколькими станциями, только одна станция в любой момент времени имеет право передавать данные. Таким образом, должен существовать метод определения того, какая станция имеет право передавать данные в каждый конкретный момент времени. В соответствии с этими требованиями были созданы методы контроля доступа к среде передачи, которые мы обзорно рассмотрели в разделе "Процесс обмена данными".

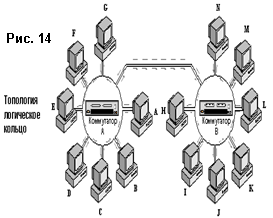
Наиболее часто используемым при организации топологии логической шины методом контроля доступа к среде передачи является CSMA/CD – “метод прослушивания несущей, с организацией множественного доступа и обнаружением коллизий” (Carrier Sense Multiple Access/ Collision Detection). Этот метод доступа очень похож на разговор нескольких людей в одной комнате. Для того, чтобы не мешать друг другу, в любой момент времени говорит только один человек, а все остальные слушают. А начинать говорить кто-либо может только, убедившись в том, что в комнате воцарилось молчание. Точно таким же образом работает и сеть. Когда какая-либо станция собирается передавать данные, сначала она "прослушивает" (carrier sense) среду передачи данных в целях обнаружения какой-либо уже передающей данные станции. Если какая-либо станция в данный момент выполняет передачу, то станция ждет окончания процесса передачи. Когда среда передачи освобождается, ожидавшая станция начинает передачу своих данных. Если в этот момент начинается передача еще одной или несколькими станциями тоже ожидавшими освобождения среды передачи, то возникает "коллизия" (collision). Все передающие станции обнаруживают коллизию и посылают специальный сигнал информирующий все станции сети о возникновении коллизии. После этого все станции замолкают на случайный промежуток времени перед повторной попыткой передачи данных. После этого алгоритм работы начинается сначала.

Сеть, базирующаяся на топологии логической шины, может также использовать и технологию "передачи маркера" (token passing) для контроля доступа к среде передачи данных. При использовании этого метода контроля каждой станции назначается порядковый номер указывающий очередность в передаче данных. После передачи данных станцией с максимальным номером, очередь возвращается к первой станции. Порядковые номера, назначаемые станциям, могут не соответствовать реальной последовательности физического подключения станций к среде передачи данных. Для контроля того, какая станция в текущий момент времени имеет право передать данные, используется контрольный кадр данных, называемый "маркером доступа". Этот маркер передается от станции к станции в последовательности, соответствующей их порядковым номерам. Станция, получившая маркер, имеет право передать свои данные. Однако, каждая передающая станция ограничена временем, в течение которого ей разрешается передавать данные. По окончании этого времени станция обязана передать маркер следующей станции.

Работа такой сети начинается с того, что первая станция, имеющая маркер доступа, передает свои данные и получает на них ответы в течение ограниченного промежутка времени (time slot). Если станция завершает обмен данными ранее окончания выделенного ей времени, она просто передает маркер станции со следующим порядковым номером. Далее процесс повторяется. Такой последовательный процесс передачи маркера продолжается непрерывно, предоставляя возможность каждой станции через строго определенный промежуток времени получить возможность передать данные. 

Топология "логической шины" базируется на использовании топологий "физическая шина" и "физическая звезда". Метод контроля доступа и типы физических топологий выбираются в зависимости от требований к проектируемой сети. Например, каждая из сетей: Ethernet, 10Base-T Ethernet и ARCnet® используют топологию "логическая шина". Кабели в сетях Ethernet (тонкий коаксиальный кабель) подключаются с использованием топологии "физическая шина", а сети 10Base-T Ethernet и ARCnet базируются на топологии "физическая звезда". Вместе с тем, сети Ethernet (физическая шина) и 10Base-T Ethernet (физическая звезда) используют CSMA/CD в качестве метода контроля доступа к среде передачи данных, а в ARCnet (физическая звезда) применяется маркер доступа. На рисунке 13 показана сеть Ethernet (физическая шина, логическая шина), а на рисунке 12 – проиллюстрирована сеть 10Base-T Ethernet (физическая звезда, логическая шина). На обоих рисунках 12-13 обратите внимание на то, что сигнал (показан стрелками) исходит от одной (передающей в данный момент) станции и распространяется во всех направлениях существующей среды передачи.

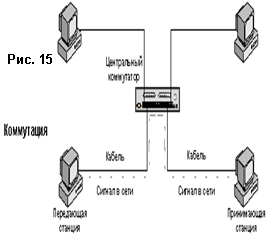
**Логическое кольцо**

В топологии "логическое кольцо" кадры данных передаются по физическому кольцу до тех пор, пока не пройдут через всю среду передачи данных. Топология "логическое кольцо" базируется на топологии "физическое кольцо с подключением типа "звезда"". Каждая станция, подключенная к физическому кольцу, получает данные от предыдущей станции и повторяет этот же сигнал для следующей станции. Таким образом, данные, повторяясь, следуют от одной станции к другой до тех пор, пока не достигнут станции, которой они были адресованы. Получающая станция, копирует данные из среды передачи и добавляет к кадру атрибут, указывающий на успешное получение данных. Далее кадр с установленным "атрибутом доставки" продолжает путешествие по кольцу до тех пор, пока не достигнет станции, изначально отправившей эти данные. Станция, проанализировав "атрибут доставки" и убедившись в успешности передачи данных, удаляет свой кадр из сети. Рисунок 14 демонстрирует процесс передачи данных в виде "логического кольца" в сети, базирующейся на топологии "физическое кольцо с подключением типа "звезда"".

Метод контроля доступа к среде передачи в таких сетях всегда базируется на технологии "маркеров доступа". Однако последовательность получения права на передачу данных (путь следования маркера), не всегда может соответствовать реальной последовательности подключения станций к физическому кольцу. IBM's Token-Ring является примером сети, использующей топологию "логического кольца", базирующегося на "физическом кольце с подключением типа "звезда""

**Логическая звезда (коммутация)**

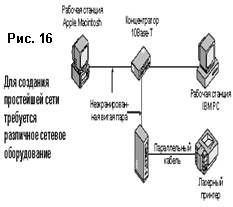
В топологии "логическая звезда" используется метод коммутации, обеспечивающий ограничение распространения сигнала в среде передачи в пределах некоторой ее части. Механизм такого ограничения является основополагающим в топологии "логическая звезда". В чистом виде, коммутация предоставляет выделенную линию передачи данных каждой станции. Когда одна станция передает сигнал другой станции подключенной к тому же самому коммутатору, то коммутатор передает сигнал только по среде передачи данных, соединяющей эти две станции. Рисунок 15 показывает способ передачи данных между двумя станциями, подключенными к одному и тому же коммутатору. При таком подходе возможна одновременная передача данных между несколькими парами машин, так как данные, передающиеся между любыми двумя станциями, остаются "невидимыми" для других пар станций.

Большинство технологий коммутации создаются на базе существующих сетевых стандартов, привнося в них новый уровень функциональности. Например, рассмотренный ранее стандарт сети 10Base-T (метод контроля CSMA/CD), позволяет применять коммутацию. Некоторые коммутаторы разрабатываются для поддержки возможностей одновременного использования нескольких сетевых стандартов. Например, один коммутатор может иметь порты для подключения станций как по стандарту 10Base-T Ethernet, так и FDDI (Fiber Distributed Data Interface).

Коммутаторы имеют встроенную логику, позволяющую им интеллектуально управлять процессом передачи данных между машинами. Внутренней логике коммутаторов свойственно высокое быстродействие, т.к. они должны обеспечивать возможность одновременной передачи данных с максимальной скоростью между каждой парой портов. Таким образом, использование коммутаторов позволяет существенно увеличить производительность сети. Коммутация иллюстрирует то, что логическая топология определяется не только методом контроля доступа к среде передачи, но и множеством других аспектов схем электронных соединений (коммутатор является достаточно сложным и дорогим электронным устройством).

Комбинируя новые технологии коммутации с существующими логическими схемами соединения, инженеры получают возможность создания новых логических топологий. Несколько коммутаторов могут быть соединены между собой с использованием одной или нескольких физических топологий. Коммутаторы могут быть использованы не только для соединения индивидуальных станций, но и целых групп станций. Такие группы носят название "сегментов сети". Таким образом, по множеству причин коммутация может значительно повысить производительность Вашей сети.

**Подключение к простейшей сети**

Теперь, когда мы обсудили вопросы связанные с аппаратной реализацией различных компонентов сети и уяснили различия между логическими и физическими топологиями, рассмотрим способы подключения оборудования в простейшей сети. На рисунке 16 показаны некоторые ранее рассмотренные сетевые устройства, подключенные к простейшей компьютерной сети.

Изображенная сеть состоит из следующих компонентов: три компьютера подключены к одному концентратору 10Base-T с использованием неэкранированной витой пары. На каждый компьютер установлены сетевые карты 10Base-T Ethernet. К одному из компьютеров также подключен лазерный принтер. Компьютер в центральной нижней части рисунка является сервером и осуществляет контроль над всей сетью. Два оставшиеся компьютера – это рабочие станции. Рабочие станции используют сеть, контролируемую сервером. Одна рабочая станция – это персональный компьютер типа IBM PC, другая –компьютер Apple® Macintosh. Концентратор 10Base-T обеспечивает физическое соединение всех трех компьютеров. Он также несет функции повторителя сигналов. Линии между различными компонентами сети обозначают среду передачи: витую пару. Эта сеть использует топологию "физическая звезда", но базируется на логической топологии "логическая шина". Принтер в этой сети подключен непосредственно к серверу с использованием параллельного порта этого компьютера. Такое подключение является стандартным для большинства принтеров. Сервер принимает задания на печать документов поступающих от каждой из рабочих станций. Поступившие задания на печать далее поступают к принтеру через параллельный порт сервера по соответствующему кабелю. Несмотря на то, что такой способ является наиболее простым для предоставления возможности нескольким станциям печатать документы на одном принтере, тем не менее существуют и другие способы подключения принтеров к сети. Вы можете, например, подключить принтер к специальному серверу печати или компьютеру со специальным программным обеспечением, предоставляющим возможность одновременно выполнять функции рабочей станции и сервера печати. Сейчас множество принтеров выпускается со встроенной в него сетевой картой, таким образом, принтер может подключаться непосредственно к среде передачи в любой точке сети.

**4. Среда передачи.**

**Среда передачи данных внутри сети**

В настоящее время используются различные виды сред для передачи данных внутри локальной сети.

Для построения локальных связей в вычислительных сетях используются различные виды кабелей - *коаксиальный* кабель, кабель на основе *экранированной* и *неэкранированной витой пары* и *оптоволоконный кабель*. Наиболее популярным видом среды передачи данных на небольшие расстояния (до 100 м) становится неэкранированная витая пара, которая включена практически во все современные стандарты и технологии локальных сетей и обеспечивает пропускную способность до 100 Мб/с (на кабелях категории 5). Оптоволоконный кабель широко применяется как для построения локальных связей, так и для образования магистралей глобальных сетей. Оптоволоконный кабель может обеспечить очень высокую пропускную способность канала (до нескольких Гб/с) и передачу на значительные расстояния (до нескольких десятков километров без промежуточного усиления сигнала).

Согласно зарубежным исследованиям (журнал LAN Technologies), 70% времени простоев обусловлено проблемами, возникшими вследствие низкого качества применяемых кабельных систем. Поэтому так важно правильно построить фундамент сети - кабельную систему. В последнее время в качестве такой надежной основы все чаще используется структурированная кабельная система.

*Структурированная кабельная система* (Structured Cabling System, *SCS*) - это набор коммутационных элементов (кабелей, разъемов, коннекторов, кроссовых панелей и шкафов), а также методика их совместного использования, которая позволяет создавать регулярные, легко расширяемые структуры связей в вычислительных сетях.

Преимущества структурированной кабельной системы:

* Универсальность. Структурированная кабельная система при продуманной организации может стать единой средой для передачи компьютерных данных в локальной вычислительной сети, организации локальной телефонной сети, передачи видеоинформации и даже передачи сигналов от датчиков пожарной безопасности или охранных систем. Это позволяет автоматизировать многие процессы по контролю, мониторингу и управлению хозяйственными службами и системами жизнеобеспечения.
* Увеличение срока службы. Срок старения хорошо структурированной кабельной системы может составлять 8-10 лет.
* Уменьшение стоимости добавления новых пользователей и изменения их мест размещения. Стоимость кабельной системы в основном определяется не стоимостью кабеля, а стоимостью работ по его прокладке. Поэтому более выгодно провести однократную работу по прокладке кабеля, возможно с большим запасом по длине, чем несколько раз выполнять прокладку, наращивая длину кабеля. Это помогает быстро и дешево изменять структуру кабельной системы при перемещениях персонала или смене приложений.
* Возможность легкого расширения сети. Структурированная кабельная система является модульной, поэтому ее легко наращивать, позволяя легко и ценой малых затрат переходить на более совершенное оборудование, удовлетворяющее растущим требованиям к системам коммуникаций.
* Обеспечение более эффективного обслуживания. Структурированная кабельная система облегчает обслуживание и поиск неисправностей по сравнению с шинной кабельной системой.
* Надежность. Структурированная кабельная система имеет повышенную надежность поскольку обычно производство всех ее компонентов и техническое сопровождение осуществляется одной фирмой-производителем.

**ЗАКЛЮЧЕНИЕ**

В компьютерных сетях идеологической основой стандартизации является многоуровневый подход к разработке средств сетевого взаимодействия.  
Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются протоколом.  
Формализованные правила, определяющие взаимодействие сетевых компонентов соседних уровней одного узла, называются интерфейсом. Интерфейс определяет набор сервисов, предоставляемых данным уровнем соседнему уровню.  
Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов сети, называется стеком коммуникационных протоколов.  
Открытой системой может быть названа любая система, которая построена в соответствии с общедоступными спецификациями, соответствующими стандартами и принятыми в результате публичного обсуждения всеми заинтересованными сторонами.  
Модель OSI стандартизирует взаимодействие открытых систем. Она определяет семь уровней взаимодействия: прикладной, представительный, сеансовый, транспортный, сетевой, канальный и физический.  
Важнейшим направлением стандартизации в области вычислительных сетей является стандартизация коммуникационных протоколов. Наиболее популярными являются наборы протоколов: TCP/IP, IPX/SPX, NetBIOS/SMB, DECnet и SNA.

**Список литературы**

1. Иллюстрированный самоучитель по созданию компьютерной сети, URL: http://adminbook.ru/index.php?men1=5 (дата обращения 17.05.14).
2. О локальных сетях, URL: http://freedomurban.at.ua/forum/20-59-1 (дата обращения 05.05.14).
3. Локальная вычислительная сеть, URL: ru.wikipedia.org/wiki/Локальная\_сеть (дата обращения 05.05.14).