

# **Лабораторная работа №3**

## **Организация взаимодействия открытых систем**

### **1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ**

#### *Цель работы*

Изучение процесса эволюции открытых сетей.

#### *Задачи работы*

1. Изучить многоуровневую структуру коммуникаций.
2. Изучить эталонную модель OSI.
3. Изучить эталонную модель TCP/IP.
4. Изучить стек
5. Выполнить упражнения для закрепления изученного материала.

### **2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

#### **ЭВОЛЮЦИЯ ОТКРЫТЫХ СЕТЕЙ**

В своем первоначальном варианте компьютерные сети представляли собой системы, позволявшие соединять разнородные компоненты и обладавшие закрытой архитектурой. Если в доисторические времена до появления персональных компьютеров какая-либо компания пыталась автоматизировать обработку данных или бухгалтерский учет, она обращалась к одному поставщику и приобретала целый аппаратно-программный комплекс, готовый к эксплуатации.

В такой закрытой среде, ориентированной на определенную фирму-поставщика, прикладные программы работали только в окружении, поддерживаемом определенной операционной системой. Операционная система работала только в контексте безопасности оборудования, предоставленного тем же поставщиком. Даже терминалы конечных пользователей и средства подключения к центральному компьютеру были частью того же интегрированного решения, разработанного конкретным поставщиком.

В эпоху безраздельного царствования интегрированных решений Министерство обороны осознало необходимость надежной и защищенной от ошибок коммуникационной сети, которая могла бы объединить все входящие в нее компьютеры, а также компьютеры университетов, научных центров и фирм-подрядчиков. На первый взгляд задача кажется не такой уж грандиозной, но это впечатление обманчиво. На заре компьютерных технологий фирмы-поставщики разрабатывали комплексы из программных, аппаратных и сетевых компонентов с очень высокой степенью интеграции. Пользователю, пожелавшему организовать обмен данными с другой платформой, пришлось бы

нелегко. Объяснялось это очень просто: поставщики Министерства обороны хотели привязать к себе заказчиков.

Заставить всех субподрядчиков Министерства обороны и связанные с ними научные организации перейти на единый стандарт оборудования было абсолютно нереально. Соответственно, возникла необходимость в средствах для соединения разнородных платформ. Так появился первый протокол открытых коммуникаций: *IP (Internet Protocol)*.

Таким образом, *открытой* следует считать сеть, обеспечивающую взаимодействие и совместное использование ресурсов на разнородных компьютерах. Открытость достигается посредством объединенной разработки и сопровождения технических спецификаций. Эти спецификации, также называемые *открытыми стандартами*, свободно распространяются и доступны для всех желающих.

## МНОГОУРОВНЕВАЯ СТРУКТУРА КОММУНИКАЦИЙ

Открытые коммуникации возможны только при четком согласовании всех функций, необходимых для взаимодействия и обмена данными между двумя конечными системами. Идентификация этих важнейших функций и установление порядка их выполнения закладывают основу для открытых коммуникаций. Две системы смогут взаимодействовать только в том случае, если они договорятся о том, как будет организовано это взаимодействие. Иначе говоря, они должны следовать общим правилам получения данных от приложения и их упаковки для передачи по сети. Никакие подробности, даже второстепенные, не могут считаться сами собой разумеющимися или зависеть от случайностей.

К счастью, существует более или менее логичная последовательность событий, необходимых для проведения сеанса связи. В минимальном варианте необходимо выполнить следующие операции:

- \* Данные передаются от приложения коммуникационному процессу (также называемому *протоколом*). О Коммуникационный протокол должен подготовить данные приложения для передачи по сети. Обычно при этом данные разбиваются на сегменты более удобного размера.

- \* Сегментированные данные заключаются в служебную структуру данных (*конверт*) для передачи по сети заданному устройству (или устройствам). Дополнительная информация, содержащаяся в этой структуре, должна позволить любому сетевому вычислительному устройству определить, откуда поступил конверт и куда он передается. В зависимости от используемого протокола эта структура может представлять собой кадр, пакет или ячейку.

- \* Кадры (или пакеты) преобразуются в физическое битовое представление для передачи. Биты передаются в виде световых импульсов по оптоволоконным сетям (например, FDDI) или в электронном состоянии (наличие/отсутствие сигнала) по электронным сетям — например, Ethernet или любой другой сети, в которой данные передаются в электрическом виде по металлическим проводам.

В точке приема, то есть на компьютере-получателе, эти действия выполняются в обратном порядке.

Во время сеанса связи также может возникнуть необходимость и в других операциях, координирующих действия отправителя и получателя и обеспечивающих благополучное прибытие данных. К числу таких операций относятся следующие:

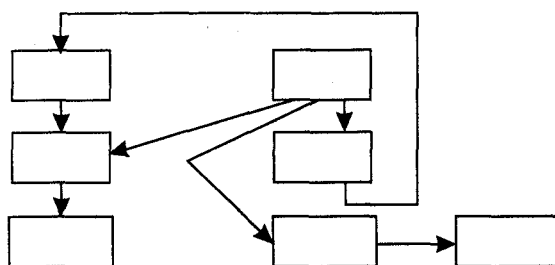
- \* Регулировка объема передаваемых данных, чтобы предотвратить перегрузку получателя и/или сети.

- \* Проверка целостности полученных данных, в ходе которой получатель убеждается в том, что данные не были повреждены в процессе передачи. Проверка осуществляется с использованием контрольных сумм — например, по алгоритму CRC (Cyclic Redundancy Checksum). Полиномиальный математический алгоритм вычисления контрольной суммы CRC позволяет обнаруживать ошибки в передаваемых данных.

- \* Координация повторной передачи данных, которые либо не прибыли по месту назначения, либо были получены в поврежденном виде.

- \* Наконец, получатель данных должен заново собрать сегменты в форму, распознаваемую приложением-получателем. С точки зрения получателя, принятые данные должны точно совпадать с данными, отправленными приложением-отправителем. Иначе говоря, результат должен выглядеть так, словно данные передаются напрямую между двумя приложениями. Данное свойство называется *логической смежностью*.

Организация взаимодействия между устройствами в сети является сложной задачей. Как известно, для решения сложных задач используется универсальный прием — декомпозиция, то есть разбиение одной сложной задачи на несколько более простых задач-модулей (рис. 1.20). Процедура декомпозиции включает в себя четкое определение функций каждого модуля, решающего отдельную задачу, и интерфейсов между ними. В результате достигается логическое упрощение задачи, а кроме того, появляется возможность модификации отдельных модулей без изменения остальной части системы.



*Рис. 1.20. Пример декомпозиции задачи*

При декомпозиции часто используют многоуровневый подход: все множество модулей разбивают на уровни. Уровни образуют иерархию, то есть имеются вышележащие и нижележащие уровни (рис. 1.21). Множество

модулей, составляющих каждый уровень, сформировано таким образом, что для выполнения своих задач они обращаются с запросами только к модулям непосредственно примыкающего нижележащего уровня. С другой стороны, результаты работы всех модулей, принадлежащих некоторому уровню, могут быть переданы только модулям соседнего вышележащего уровня. Такая иерархическая декомпозиция задачи предполагает четкое определение функции каждого уровня и интерфейсов между уровнями. Интерфейс определяет набор функций, которые нижележащий уровень предоставляет вышележащему. В результате иерархической декомпозиции достигается относительная независимость уровней, а значит, и возможность их легкой замены.

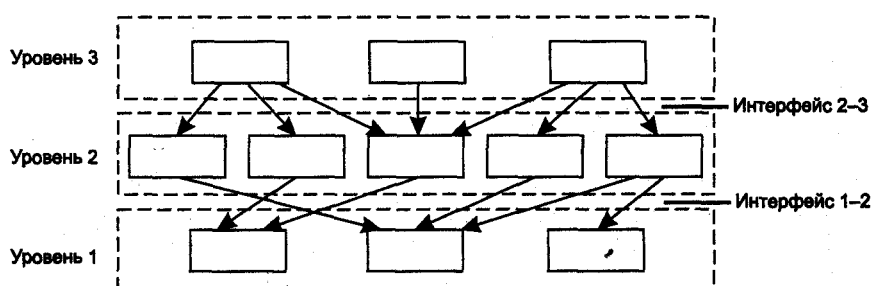


Рис. 1.21. Многоуровневый подход - создание иерархии задач

Средства сетевого взаимодействия, конечно, тоже могут быть представлены в виде иерархически организованного множества модулей.

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют две машины, то есть в данном случае необходимо организовать согласованную работу двух «иерархий». При передаче сообщений оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения длины сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты для всех уровней, начиная от самого низкого — уровня передачи битов — до самого высокого, реализующего сервис для пользователей сети.

На рис. 1.23 показана модель взаимодействия двух узлов. С каждой стороны средства взаимодействия представлены четырьмя уровнями. Процедура взаимодействия этих двух узлов может быть описана в виде набора правил взаимодействия каждой пары 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 *протоколом*.

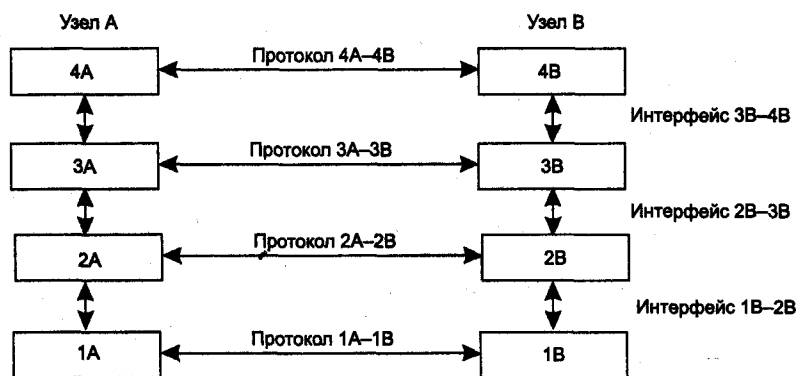


Рис. 1.23. Взаимодействие двух узлов

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 принято называть *интерфейсом*. Интерфейс определяет набор сервисов, предоставляемый данным уровнем соседнему уровню. 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, а интерфейсы — модулей соседних уровней в одном узле.

Средства каждого уровня должны отрабатывать, во-первых, свой собственный протокол, а во-вторых, интерфейсы с соседними уровнями.

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

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней — как правило, чисто программными средствами.

Программный модуль, реализующий некоторый протокол, часто для краткости также называют «протоколом». При этом соотношение между протоколом — формально определенной процедурой и протоколом — программным модулем, реализующим эту процедуру, аналогично соотношению между алгоритмом решения некоторой задачи и программой, решающей эту задачу.

Протокол может иметь несколько программных реализаций с разной степенью эффективности. Именно поэтому при сравнении протоколов следует учитывать не только логику их работы, но и качество программных решений. Более того, на эффективность взаимодействия устройств в сети влияет качество всей совокупности протоколов, составляющих стек, в частности, насколько рационально распределены функции между протоколами разных уровней и насколько хорошо определены интерфейсы между ними.

Протоколы реализуются не только компьютерами, но и другими сетевыми устройствами — концентраторами, мостами, коммутаторами, маршрутизаторами и т. д. Действительно, в общем случае связь компьютеров в сети осуществляется не напрямую, а через различные коммуникационные устройства. В зависимости от типа устройства в нем должны быть встроенные средства, реализующие тот или иной набор протоколов.

Вероятно, основные принципы многоуровневых коммуникаций, включая логическую смежность, лучше всего объяснять на примере *эталонной модели OSI*.

## ЭТАЛОННАЯ МОДЕЛЬ OSI

Международная организация по стандартизации (*ISO*, International Organization for Stanartization) разработала эталонную *модель взаимодействия открытых систем* (*OSI*, Open Systems Interconnection) в 1978/1979 годах для упрощения *открытого взаимодействия* компьютерных систем. Открытым называется взаимодействие, которое может поддерживаться в неоднородных средах, содержащих системы разных поставщиков. Модель *OSI* устанавливает глобальный стандарт, определяющий состав функциональных уровней при открытом взаимодействии между компьютерами.

Эталонная модель *OSI* разрабатывалась почти 20 лет назад, и лишь немногие коммерческие системы следовали ее спецификации. В то время производители компьютерного оборудования стремились привязать клиентов к запатентованным архитектурам, поддерживаемым одним поставщиком. С точки зрения производителей конкуренция была нежелательна. Соответственно, все функции как можно плотнее интегрировались в готовых решениях. Концепция функциональной модульности или многоуровневой структуры казалась прямо противоречащей намерениям любого производителя.

Следует заметить, что модель настолько успешно справилась со своими исходными целями, что в настоящее время ее достоинства уже практически не обсуждаются. Существовавший ранее закрытый, интегрированный подход уже не применяется на практике, в наше время открытость коммуникаций считается обязательной. Как ни странно, очень немногие продукты полностью соответствуют стандарту *OSI*. Вместо этого базовая многоуровневая структура часто адаптируется к новым стандартам. Тем не менее эталонная модель *OSI* остается ценным средством для демонстрации принципов работы сети.

Несмотря на все успехи, с эталонной моделью *OSI* связан ряд устойчивых заблуждений, поэтому в этом разделе придется привести ее очередной обзор с указанием и исправлением этих заблуждений.

Первое заблуждение заключается в том, что эталонная модель *OSI* была разработана *Международной организацией стандартов* (International Standards Organization). Это не так. Эталонная модель *OSI* разрабатывалась Международной организацией по стандартизации (International Organization for

Standartization), а эта организация предпочитает использовать мнемоническое сокращение вместо акронима. Мнемоника основана на греческом слове *isos*, которое означает «равный» или «стандартный».

Модель OSI разделяет процессы, участвующие в сеансе связи, на семь четко различающихся функциональных уровней. Структура уровней соответствует естественной последовательности событий, происходящих во время сеанса связи.

На рис. 1.1 изображена схема эталонной модели OSI. Уровни 1-3 обеспечивают доступ к сети, а уровни 4-7 посвящены логистике поддержки коммуникаций сквозной передачи.

Уровень OSI	Номер OSI
Прикладной уровень	7
Представительский уровень	6
Сеансовый уровень	5
Транспортный уровень	4
Сетевой уровень	3
Канальный уровень	2
Физический уровень	1

*Рис. 1.1. Эталонная модель OSI*

## 1. ФИЗИЧЕСКИЙ УРОВЕНЬ

Нижний уровень модели называется *физическим уровнем* (physical layer) и отвечает за передачу битового потока. Физический уровень получает кадры данных от уровня 2 (канального уровня) и передает их структуру и содержимое; обычно передача осуществляется последовательно по одному биту. Физический уровень также отвечает за побитовый прием входящих потоков данных. Затем этот поток передается канальному уровню для повторного формирования кадров.

В сущности, физический уровень имеет дело только с нулями и единицами. Он не обладает механизмом интерпретации передаваемых или получаемых битов. Все, что его интересует, — это физические характеристики электрических и/или оптических средств передачи сигнала. В частности, сюда относится напряжение электрического тока, используемого для передачи сигнала, тип носителя, характеристики волнового сопротивления и даже физическая форма коннектора, подключенного к передающей среде.

Основное заблуждение состоит в том, будто уровень 1 модели OSI включает компоненты, способные генерировать или передавать коммуникационные сигналы. На самом деле это не так, поскольку уровень 1 является всего лишь *функциональной моделью*. Физический уровень ограничивается процессами и механизмами, необходимыми для подачи сигнала в передающую среду и для приема сигналов от этой среды. Его нижней границей является физический коннектор, подключенный к передающей среде.

Сама среда в этот уровень *не входит*, хотя различные стандарты локальных сетей (LAN, Local Area Network), такие как 10BASET и 100BASET, указывают тип передающей среды.

Все средства непосредственной передачи сигнала, сгенерированного механизмами уровня 1 модели OSI, относятся к передающей среде. Среди примеров передающей среды можно назвать коаксиальные кабели, оптоволоконные кабели и витую пару. Вероятно, недоразумение обусловлено тем фактом, что физический уровень определяет некоторые обязательные характеристики, которыми Должна обладать передающая среда для работы процессов и механизмов, определенных на физическом уровне.

Соответственно, передающая среда остается за пределами физического уровня. Иногда ее называют *нулевым уровнем* эталонной модели OSI.

## 2. КАНАЛЬНЫЙ УРОВЕНЬ

Второй уровень эталонной модели OSI называется *канальным уровнем* (data link layer). Его задачи, как и задачи всех остальных уровней, делятся на две категории: прием и передача данных. Он отвечает за достоверность переданных данных, обычно — по физическому каналу

На передающей стороне канальный уровень отвечает за упаковку инструкций, данных и т. д. в кадры. *Кадр* (frame) представляет собой структуру данных, специфическую для канального уровня; эта структура содержит информацию, достаточную для успешной передачи данных по физическому каналу (например, по локальной сети) к точке приема.

Успешная доставка означает, что кадр достигает положенного места назначения без изменений. Следовательно, кадры также должны содержать информацию для проверки целостности содержимого после доставки.

Гарантированная доставка происходит при выполнении двух условий:

- \* Узел-отправитель должен получить подтверждение того, что каждый кадр был принят узлом-получателем без изменений.

- \* Перед тем как подтверждать прием кадра, узел-получатель должен проверить целостность его содержимого.

По некоторым причинам переданные кадры могут не достигнуть места назначения или их содержимое будет искажено и станет непригодным в процессе передачи. Канальный уровень отвечает за обнаружение и исправление всех подобных ошибок. Также он отвечает за повторную сборку двоичного потока, полученного от физического уровня, и формирование кадров. Впрочем, если учесть, что в кадрах передается как управляющая информация, так и содержимое, канальный уровень на самом деле не восстанавливает кадры, а просто накапливает поступающие биты до получения полного кадра.

Уровни 1 и 2 обязательны для всех типов коммуникаций, как в локальных, так и в глобальных сетях. Как правило, функциональность уровней 1 и 2 реализуется сетевым адаптером, установленным внутри компьютера.

## 3. СЕТЕВОЙ УРОВЕНЬ

*Сетевой уровень* (network layer) отвечает за установление маршрута между отправителем и получателем. Этот уровень не располагает



собственными средствами обнаружения/исправления ошибок передачи, поэтому он использует средства надежной пересылки данных уровня 2. Надежность сквозной передачи данных по нескольким физическим каналам в большей степени относится к функциям транспортного уровня (уровень 4).

Сетевой уровень используется для установления связи с компьютерными системами, не входящими в местный сегмент локальной сети. Это возможно благодаря его собственной архитектуре маршрутной адресации, независимой от машинной адресации уровня 2. Подобные протоколы называются *маршрутизируемыми*. К этой категории относятся протоколы IP, IPX (компания Novell) и AppleTalk, которые будут рассматриваться позже, а также связанные с ним протоколы и приложения.

Использование сетевого уровня при коммуникациях не обязательно. Сетевой уровень необходим лишь в том случае, если компьютеры находятся в разных сегментах сети, разделенных маршрутизатором, или взаимодействующие приложения должны использовать возможности сетевого или транспортного уровня. Например, два хоста, напрямую подключенных к одной локальной сети, могут взаимодействовать, ограничиваясь только механизмами локальной сети (уровни 1 и 2 эталонной модели OSI).

**ПРИМЕЧАНИЕ** В наши дни необходимость в соединении сетей возникает так часто, что сетевой уровень уже не считается необязательным. Работа старых протоколов, не обладающих собственной реализацией сетевого уровня (например, NetBEUI), в современных сетевых архитектурах обеспечивается как особый случай.

#### 4. ТРАНСПОРТНЫЙ УРОВЕНЬ

*Транспортный уровень* (transport layer), как и канальный уровень, отвечает за целостность передаваемых данных. Тем не менее транспортный уровень способен выполнять эту функцию за пределами местного сегмента локальной сети. Он обнаруживает пакеты, отвергнутые маршрутизатором, и автоматически генерирует запрос на повторную передачу. Другими словами, транспортный уровень обеспечивает настоящую надежность сквозной передачи.

Другая важная функция транспортного уровня — переупорядочивание пакетов, поступивших с нарушением исходной последовательности. Нарушение может происходить по разным причинам — например, из-за того, что пакеты перемещались в сети по разным маршрутам или были повреждены в процессе пересылки. Так или иначе, транспортный уровень способен определить исходную последовательность пакетов и расположить их в этой последовательности перед тем, как передавать их содержимое сеансовому уровню.

#### 5. СЕАНСОВЫЙ УРОВЕНЬ

Пятое место в модели OSI занимает *сеансовый уровень* (session layer). Он используется относительно редко, многие протоколы реализуют его функциональные возможности на своих транспортных уровнях.

Основная функция сеансового уровня OSI — организация *сеанса*, то есть передачи управления во время связи двух компьютерных систем. Сеанс определяет направленность передачи данных (одно- или двусторонняя), а также гарантирует завершение обработки одного запроса до принятия следующего.

Сеансовый уровень также может поддерживать некоторые из следующих дополнений:

- \* Управление диалогом.
- \* Управление маркерами.
- \* Управление операциями.

В общем случае сеанс поддерживает двусторонний (*дуплексный*) режим обмена данными. В некоторых приложениях вместо него может понадобиться односторонний (*полудуплексный*) режим обмена. Сеансовый уровень позволяет выбрать нужный режим, и эта возможность называется *управлением диалогом*.

Для работы некоторых протоколов очень важно, чтобы в любой момент времени попытки выполнения критических операций могли выполняться только одной из сторон. Чтобы обе стороны не пытались одновременно выполнить одну и ту же операцию, необходимо реализовать специальный механизм управления — например, основанный на использовании *маркеров* (tokens). В этом случае выполнение операции разрешается только той стороне, которую в настоящий момент удерживает маркер. Определение того, на какой стороне находится маркер и как он передается между двумя сторонами, называется *управлением маркером*. Функции управления маркером используются в протоколах сеансового уровня ISO.

#### ПРИМЕЧАНИЕ

Не путайте термин «маркер» с похожим термином, существующим в архитектуре Token Ring. Управление маркером на уровне 5 модели OSI происходит на значительно более высоком уровне. Архитектура IBM Token Ring работает на 1 и 2 уровнях модели OSI.

Если передача файла между двумя компьютерами занимает один час, а сетевые сбои происходят примерно каждые 30 минут, скорее всего, переслать файл вообще никогда не удастся. После каждого аварийного завершения пересылку приходится начинать заново. Для решения этой проблемы вся пересылка файла рассматривается как одна операция, а в поток данных вставляются контрольные точки. В этом случае при возникновении сбоя сеансовый уровень может продолжить пересылку данных с предыдущей контрольной точки. Эти контрольные точки называются *точками синхронизации*.

Точки синхронизации делятся на первичные и вторичные. Первичные точки синхронизации, вставленные в поток данных любой из взаимодействующих сторон, должны подтверждаться другой стороной, тогда как вторичные точки синхронизации подтверждения не требуют. Часть сеанса между двумя первичными точками синхронизации называется *диалоговым блоком*, а управление всем процессом называется *управлением операциями*. Операция состоит из одного или нескольких диалоговых блоков.

В сетях TCP/IP отсутствует общий протокол сеансового уровня. Это объясняется тем, что некоторые характеристики сеансового уровня обеспечиваются протоколом TCP. Если приложениям TCP/IP потребуются специальные возможности сеансового уровня, они должны реализовать их самостоятельно. Например, файловая система NFS (Network File System) самостоятельно реализует сервис сеансового уровня — протокол RPC (Remote Procedure Call).

**ПРИМЕЧАНИЕ** Протокол NFS также использует собственный сервис сеансового уровня — протокол XDR (External Data Representation).

## 6. ПРЕДСТАВИТЕЛЬСКИЙ УРОВЕНЬ

*Представительский уровень* (Presentation Layer) отвечает за управление кодировкой данных. Во многих компьютерных системах используются разные схемы кодировки, и представительский уровень должен обеспечить преобразование между несовместимыми кодировками — такими, как *ASCII* (American Standard Code for Information Interchange) и *EBCDIC* (Extended Binary Coded Decimal Interchange Code).

Представительский уровень также может использоваться в качестве промежуточного звена, обеспечивающего преобразование между форматами представления вещественных чисел, числовыми форматами (например, поразрядным дополнением до единицы или двойки), порядком следования байтов, а также предоставляющего возможности шифрования/дешифрования данных.

Представительский уровень обеспечивает представление данных с единым синтаксисом и семантикой. Если все узлы будут использовать этот общий язык и понимать его, это предотвратит возможную неправильную интерпретацию представления данных. Примером такого общего языка является рекомендуемый OSI язык ASN. 1 (Abstract Syntax Representation, Rev. 1). Кстати говоря, ASN. 1 используется протоколом SNMP (Simple Network Management Protocol) для кодирования данных высокого уровня.

## 7. ПРИКЛАДНОЙ УРОВЕНЬ

На самом верху эталонной модели OSI находится *прикладной уровень* (application level). Несмотря на свое название, этот уровень не включает в себя пользовательские приложения, а лишь обеспечивает взаимодействие этих приложений с сетевым сервисом.

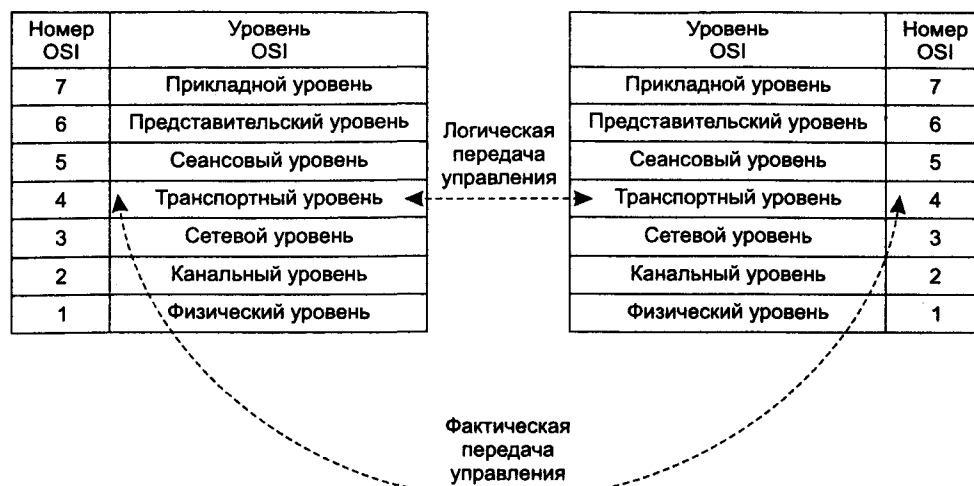
Этот уровень может рассматриваться как сторона, непосредственно иницилирующая сеанс связи. Предположим, почтовый клиент должен запросить новые сообщения с почтового сервера. Клиентское приложение автоматически генерирует запрос к соответствующему протоколу (или протоколам) 7 уровня и иницирует сеанс связи для получения нужных файлов.

## ИСПОЛЬЗОВАНИЕ МОДЕЛИ

Вертикальное расположение уровней отражает функциональную структуру процессов и данных, участвующих в передаче. Каждый уровень связывается со смежными уровнями через интерфейсы. Связь между системами обеспечивается передачей данных, инструкций, адресов и т. д. между уровнями.

Различия между логической и фактической последовательностью передач управления во время сеанса связи показаны на рис. 1.2.

**ПРИМЕЧАНИЕ** Хотя эталонная модель состоит из семи уровней, это не означает, что все уровни задействованы в каждом сеансе связи. Например, обмен данными через один сегмент локальной сети может производиться исключительно на уровнях 1 и 2, без участия других уровней.



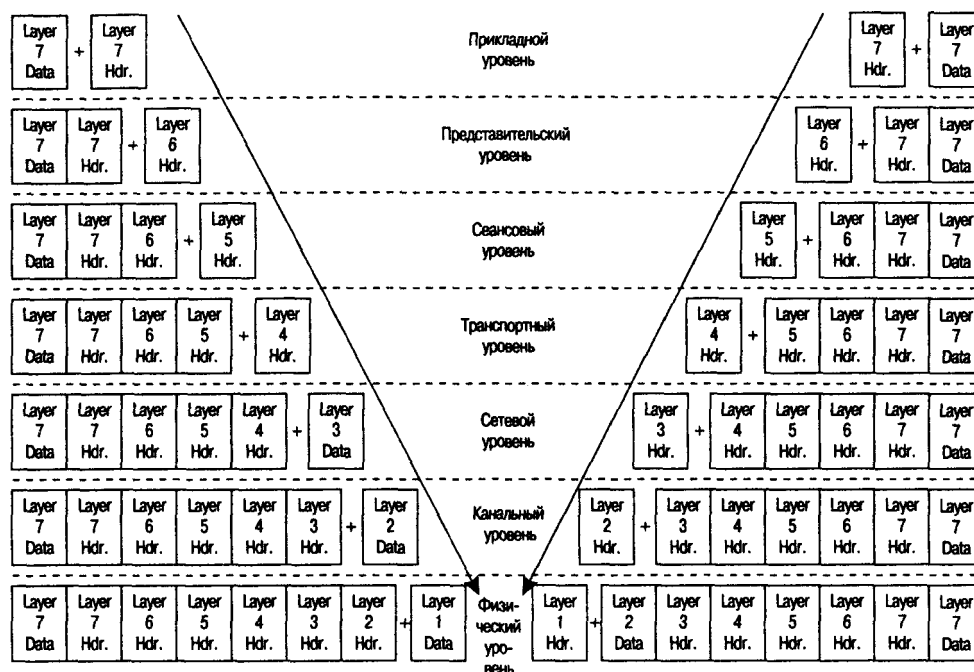
**Рис. 1.2.** Логическая и фактическая последовательность передачи управления в многоуровневой коммуникационной модели

Хотя управление передается по вертикали, каждый уровень воспринимает это так, словно он может напрямую взаимодействовать со своим аналогом на удаленном компьютере. Чтобы создать логическую смежность уровней, каждый уровень стека протоколов компьютера-отправителя добавляет свой заголовок, который распознается и используется только этим уровнем или его аналогами на других компьютерах. Стек протоколов компьютера-получателя последовательно извлекает заголовки по уровням в процессе передачи данных приложению. Этот процесс проиллюстрирован на рис. 1.3.

Например, сегменты данных упаковываются уровнем 4 компьютера-отправителя для передачи уровню 3. Уровень 3 преобразует данные, полученные от уровня 4, в пакеты, снабжает адресами и отправляет протоколу уровня 3 компьютера-получателя через свой уровень 2. Уровень 2 преобразует пакеты в кадры, снабженные адресами, распознаваемыми локальной сетью. Кадры передаются уровню 1 для преобразования в поток двоичных цифр (битов), передаваемых уровню 1 получателя.

На компьютере-получателе эта цепочка повторяется в обратном порядке, при этом каждый уровень удаляет заголовки, добавленные аналогичными уровнями на компьютере-отправителе. К тому моменту, когда данные достигнут уровня 4, они имеют ту же форму, в которой они были представлены на уровне 4 отправителя. Соответственно, результат выглядит так, словно два протокола 4 уровня являются физически смежными и обмениваются данными напрямую.

**ПРИМЕЧАНИЕ** Следует учитывать, что многие современные сетевые протоколы используют собственные многоуровневые модели. Эти модели различаются по степени соответствия стандартам разделения функций, заявленным в эталонной модели OSI. Довольно часто эти модели объединяют семь уровней OSI в пять и менее уровней. Кроме того, верхние уровни иерархий часто не соответствуют в полной мере своим аналогам в модели OSI.



*Рис. 1.3. Применение многоуровневых заголовков для поддержки логической смежности уровней*

На самом деле уровень 3 передает данные «вниз» уровню 2, который, в свою очередь, преобразует кадры в битовый поток. После того как битовый поток поступает устройству уровня 1 получателя, он передается уровню 2 для повторного формирования кадров. После успешного завершения приема кадра служебные данные удаляются, а хранящийся в кадре пакет передается уровню 3 получателя. Информация поступает точно в таком же виде, в котором она была отправлена. С точки зрения уровня 3 передача данных производилась фактически напрямую.

Возможность обмена данными между смежными уровнями (с точки зрения этих уровней) стала одним из факторов, обусловивших успех модели.

Хотя эталонная модель OSI изначально предназначалась для определения архитектуры протоколов открытых коммуникаций, в этом качестве ее ждал бесславный провал. Существует шутка, что к уровням модели OSI нужно добавить ещё два – экономический и политический.

В сущности, модель почти полностью выродилась в академическую абстракцию. Но зато эта модель великолепно подходит для объяснения концепций открытых коммуникаций и логической последовательности

выполнения необходимых действий в сеансе связи. Для практических целей существует другая, гораздо более содержательная модель — эталонная модель TCP/IP. Она описывает архитектуру семейства протоколов IP.

## ЭТАЛОННАЯ МОДЕЛЬ TCP/IP

В отличие от эталонной модели OSI, модель TCP/IP в большей степени ориентируется на обеспечение сетевых взаимодействий, нежели на жесткое разделение функциональных уровней. Для этой цели она признает важность иерархической структуры функций, но предоставляет проектировщикам протоколов достаточную гибкость в реализации. Соответственно, эталонная модель OSI гораздо лучше подходит для объяснения механики межкомпьютерных взаимодействий, но протокол TCP/IP стал основным межсетевым протоколом.

Гибкость эталонной модели TCP/IP по сравнению с эталонной моделью OSI продемонстрирована на рис. 1.4.

Уровень OSI	Номер OSI	Эквивалентный уровень TCP/IP
Прикладной уровень	7	Прикладной уровень
Представительский уровень	6	
Сеансовый уровень	5	
Транспортный уровень	4	Межхостовой уровень
Сетевой уровень	3	Межсетевой уровень
Канальный уровень	2	Уровень сетевого доступа
Физический уровень	1	

*Рис. 1.4. Сравнение эталонных моделей TCP/IP и OSI*

Эталонная модель TCP/IP разрабатывалась значительно позже описываемого ей протокола и отличается значительно большей гибкостью, чем модель OSI, поскольку она в большей степени подчеркивает иерархическое расположение функций вместо жесткого определения функциональных уровней.

## АНАТОМИЯ МОДЕЛИ TCP/IP

Стек протоколов TCP/IP состоит из четырех функциональных уровней: прикладного, межхостового, межсетевого и уровня сетевого доступа. Эти четыре уровня приблизительно соответствуют семи уровням эталонной модели OSI с сохранением общей функциональности.

### ПРИКЛАДНОЙ УРОВЕНЬ

Прикладной уровень содержит протоколы удаленного доступа и совместного использования ресурсов. Хорошо знакомые нам приложения — такие, как Telnet, FTP, SMTP, HTTP и многие другие — работают на этом

уровне и зависят от функциональности уровней, расположенных ниже в иерархии. Любые приложения, использующие взаимодействие в сетях IP (включая любительские и коммерческие программы), относятся к этому уровню модели.

### МЕЖХОСТОВОЙ УРОВЕНЬ

Межхостовой уровень IP приблизительно соответствует сеансовому и транспортному уровням эталонной модели OSI. К функциям этого уровня относится сегментирование данных в приложениях для пересылки по сети, выполнение математических проверок целостности принятых данных и мультиплексирование потоков данных (как передаваемых, так и принимаемых) для нескольких приложений одновременно. Отсюда следует, что межхостовой уровень располагает средствами идентификации приложений и умеет переупорядочивать данные, принятые не в том порядке.

В настоящее время межхостовой уровень состоит из двух протоколов: протокола управления передачей *TCP* (Transmission Control Protocol) и протокола пользовательских дейтаграмм *UDP* (User Datagram Protocol). С учетом того, что Интернет становится все более транзакционно-ориентированным, был определен третий протокол, условно названный протоколом управления транзакциями/передачей *T/TCP* (Transaction/Transmission Control Protocol). Тем не менее в большинстве прикладных сервисов Интернета на межхостовом уровне используются протоколы TCP и UDP.

### МЕЖСЕТЕВОЙ УРОВЕНЬ

Межсетевой уровень IPv4 состоит из всех протоколов и процедур, позволяющих потоку данных между хостами проходить по нескольким сетям. Следовательно, пакеты, в которых передаются данные, должны быть маршрутизируемыми. За маршрутизируемость пакетов отвечает протокол IP (Internet Protocol).

Впрочем, функциональность межсетевого уровня не ограничивается формированием пакетов. Уровень должен обладать средствами преобразования адресов уровня 2 в адреса уровня 3, и наоборот. Эти функции обеспечиваются одноранговыми (peer-to-peer) протоколами.

Кроме того, межсетевой уровень должен поддерживать маршрутизацию и функции управления маршрутами. Эти функции предоставляются внешними протоколами, которые называются *протоколами маршрутизации*. К их числу относятся протоколы IGP (Interior Gateway Protocols) и EGP (Exterior Gateway Protocols). Хотя эти протоколы также находятся на межсетевом уровне, они не являются «родными» протоколами семейства IP. Следует отметить, что многие протоколы маршрутизации позволяют обнаруживать и строить маршруты в архитектурах, использующих адресацию с несколькими протоколами маршрутизации. Другими примерами протоколов маршрутизации с разными архитектурами адресации являются IPX и AppleTalk.

### УРОВЕНЬ СЕТЕВОГО ДОСТУПА

Уровень сетевого доступа состоит из всех функций, необходимых для физического подключения и передачи данных по сети. В эталонной модели OSI этот набор функций разбит на два уровня: физический и канальный. Эталонная модель TCP/IP создавалась после протоколов, присутствующих в ее названии, и в ней эти два уровня были слиты воедино, поскольку различные протоколы

IP останавливаются на межсетевом уровне. Протокол IP предполагает, что все низкоуровневые функции предоставляются либо локальной сетью, либо подключением через последовательный интерфейс.

## СТАНДАРТНЫЕ СТЕКИ КОММУНИКАЦИОННЫХ ПРОТОКОЛОВ

Важнейшим направлением стандартизации в области вычислительных сетей является стандартизация коммуникационных протоколов. В настоящее время в сетях используется большое количество стеков коммуникационных протоколов. Наиболее популярными являются стеки: TCP/IP, IPX/SPX, NetBIOS/SMB, DECnet, SNA и OSI. Все эти стеки, кроме SNA на нижних уровнях — физическом и канальном, — используют одни и те же хорошо стандартизованные протоколы Ethernet, Token Ring, FDDI и некоторые другие, которые позволяют использовать во всех сетях одну и ту же аппаратуру. Зато на верхних уровнях все стеки работают по своим собственным протоколам. Эти протоколы часто не соответствуют рекомендуемому моделью OSI разбиению на уровни. В частности, функции сеансового и представительного уровня, как правило, объединены с прикладным уровнем. Такое несоответствие связано с тем, что модель OSI появилась как результат обобщения уже существующих и реально используемых стеков, а не наоборот.

### СТЕК OSI

Следует четко различать модель OSI и стек OSI. В то время как модель OSI является концептуальной схемой взаимодействия открытых систем, стек OSI представляет собой набор вполне конкретных спецификаций протоколов. В отличие от других стеков протоколов стек OSI полностью соответствует модели OSI, он включает спецификации протоколов для всех семи уровней взаимодействия, определенных в этой модели. На нижних уровнях стек OSI поддерживает Ethernet, Token Ring, FDDI, протоколы глобальных сетей, X.25 и ISDN, — то есть использует разработанные вне стека протоколы нижних уровней, как и все другие стеки. Протоколы сетевого, транспортного и сеансового уровней стека OSI специфицированы и реализованы различными производителями, но распространены пока мало. Наиболее популярными протоколами стека OSI являются прикладные протоколы. К ним относятся: протокол передачи файлов FTAM, протокол эмуляции терминала VTP, протоколы справочной службы X.500, электронной почты X.400 и ряд других.

Протоколы стека OSI отличает большая сложность и неоднозначность спецификаций. Эти свойства явились результатом общей политики разработчиков стека, стремившихся учесть в своих протоколах все случаи жизни и все существующие и появляющиеся технологии. К этому нужно еще добавить и последствия большого количества политических компромиссов,



неизбежных при принятии международных стандартов по такому злободневному вопросу, как построение открытых вычислительных сетей.

Из-за своей сложности протоколы OSI требуют больших затрат вычислительной мощности центрального процессора, что делает их наиболее подходящими для мощных машин, а не для сетей персональных компьютеров.

Стек OSI — международный, независимый от производителей стандарт. Его поддерживает правительство США в своей программе GOSIP, в соответствии с которой все компьютерные сети устанавливаемые в правительственных учреждениях США после 1990 года, должны или непосредственно поддерживать стек OSI, или обеспечивать средства для перехода на этот стек в будущем. Тем не менее стек OSI более популярен в Европе, чем в США, так как в Европе осталось меньше старых сетей, работающих по своим собственным протоколам. Большинство организаций пока только планируют переход к стеку OSI, и очень немногие приступили к созданию пилотных проектов. Из тех, кто работает в этом направлении, можно назвать Военно-морское ведомство США и сеть NFSNET. Одним из крупнейших производителей, поддерживающих OSI, является компания AT&T, ее сеть Stargroup полностью базируется на этом стеке.

## СТЕК TCP/IP

Стек TCP/IP был разработан по инициативе Министерства обороны США более 20 лет назад 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 популярным протоколам IP и TCP, внес университет Беркли, реализовав протоколы стека в своей версии ОС UNIX. Популярность этой операционной системы привела к широкому распространению протоколов TCP, IP и других протоколов стека. Сегодня этот стек используется для связи компьютеров всемирной информационной сети Internet, а также в огромном числе корпоративных сетей.

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

Основными протоколами стека, давшими ему название, являются протоколы IP и TCP. Эти протоколы в терминологии модели OSI относятся к сетевому и транспортному уровням соответственно. IP обеспечивает продвижение пакета по составной сети, а TCP гарантирует надежность его доставки.

За долгие годы использования в сетях различных стран и организаций стек TCP/IP вобрал в себя большое количество протоколов прикладного уровня. К ним относятся такие популярные протоколы, как протокол пересылки файлов FTP, протокол эмуляции терминала telnet, почтовый протокол SMTP,

используемый в электронной почте сети Internet, гипертекстовые сервисы службы WWW и многие другие.

Сегодня стек TCP/IP представляет собой один из самых распространенных стеков транспортных протоколов вычислительных сетей. Действительно, только в сети Internet объединено около 10 миллионов компьютеров по всему миру, которые взаимодействуют друг с другом с помощью стека протоколов TCP/IP.

Стремительный рост популярности Internet привел и к изменениям в расстановке сил в мире коммуникационных протоколов — протоколы TCP/IP, на которых построен Internet, стали быстро теснить бесспорного лидера прошлых лет — стек IPX/SPX компании Novell. Сегодня в мире общее количество компьютеров, на которых установлен стек TCP/IP, сравнялось с общим количеством компьютеров, на которых работает стек IPX/SPX, и это говорит о резком переломе в отношении администраторов локальных сетей к протоколам, используемым на настольных компьютерах, так как именно они составляют подавляющее число мирового компьютерного парка и именно на них раньше почти везде работали протоколы компании Novell, необходимые для доступа к файловым серверам NetWare. Процесс становления стека TCP/IP в качестве стека номер один в любых типах сетей продолжается, и сейчас любая промышленная операционная система обязательно включает программную реализацию этого стека в своем комплекте поставки.

Хотя протоколы TCP/IP неразрывно связаны с Internet и каждый из многомиллионной армады компьютеров Internet работает на основе этого стека, существует большое количество локальных, корпоративных и территориальных сетей, непосредственно не являющихся частями Internet, в которых также используют протоколы TCP/IP. Чтобы отличать их от Internet, эти сети называют сетями TCP/IP или просто IP-сетями.

Поскольку стек TCP/IP изначально создавался для глобальной сети Internet, он имеет много особенностей, дающих ему преимущество перед другими протоколами, когда речь заходит о построении сетей, включающих глобальные связи. В частности, очень полезным свойством, делающим возможным применение этого протокола в больших сетях, является его способность фрагментировать пакеты. Действительно, большая составная сеть часто состоит из сетей, построенных на совершенно разных принципах. В каждой из этих сетей может быть установлена собственная величина максимальной длины единицы передаваемых данных (кадра). В таком случае при переходе из одной сети, имеющей большую максимальную длину, в сеть с меньшей максимальной длиной может возникнуть необходимость деления передаваемого кадра на несколько частей. Протокол IP стека TCP/IP эффективно решает эту задачу.

Другой особенностью технологии TCP/IP является гибкая система адресации, позволяющая более просто по сравнению с другими протоколами аналогичного назначения включать в интерсеть сети других технологий. Это

свойство также способствует применению стека TCP/IP для построения больших гетерогенных сетей.

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, как и всегда, за получаемые преимущества надо платить, и платой здесь оказываются высокие требования к ресурсам и сложность администрирования IP-сетей. Мощные функциональные возможности протоколов стека TCP/IP требуют для своей реализации высоких вычислительных затрат. Гибкая система адресации и отказ от широковещательных рассылок приводят к наличию в IP-сети различных централизованных служб типа DNS, DHCP и т. п. Каждая из этих служб направлена на облегчение 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 протоколов Internet, однако факт остается фактом — сегодня это самый популярный стек протоколов, широко используемый как в глобальных, так и локальных сетях.

## СТЕК IPX/SPX

Этот стек является оригинальным стеком протоколов фирмы Novell, разработанным для сетевой операционной системы NetWare еще в начале 80-х годов. Протоколы сетевого и сеансового уровней Internetwork Packet Exchange (IPX) и Sequenced Packet Exchange (SPX), которые дали название стеку, являются прямой адаптацией протоколов XNS фирмы Xerox, распространенных в гораздо меньшей степени, чем стек IPX/SPX. Популярность стека IPX/SPX непосредственно связана с операционной системой Novell NetWare, которая еще сохраняет мировое лидерство по числу установленных систем, хотя в последнее время ее популярность несколько снизилась и по темпам роста она отстает от Microsoft Windows NT.

Многие особенности стека IPX/SPX обусловлены ориентацией ранних версий ОС NetWare (до версии 4.0) на работу в локальных сетях небольших размеров, состоящих из персональных компьютеров со скромными ресурсами. Понятно, что для таких компьютеров компании Novell нужны были протоколы, на реализацию которых требовалось бы минимальное количество оперативной памяти (ограниченной в IBM-совместимых компьютерах под управлением MS-DOS объемом 640 Кбайт) и которые бы быстро работали на процессорах небольшой вычислительной мощности. В результате протоколы стека IPX/SPX до недавнего времени хорошо работали в локальных сетях и не очень — в больших корпоративных сетях, так как они слишком перегружали медленные глобальные связи широковещательными пакетами, которые интенсивно используются несколькими протоколами этого стека (например, для установления связи между клиентами и серверами). Это обстоятельство, а также тот факт, что стек IPX/SPX является собственностью фирмы Novell и на

его реализацию нужно получать лицензию (то есть открытые спецификации не поддерживались), долгое время ограничивали распространенность его только сетями NetWare. Однако с момента выпуска версии NetWare 4.0 Novell внесла и продолжает вносить в свои протоколы серьезные изменения, направленные на их адаптацию для работы в корпоративных сетях. Сейчас стек IPX/SPX реализован не только в NetWare, но и в нескольких других популярных сетевых ОС, например SCO UNIX, Sun Solaris, Microsoft Windows NT.

## СТЕК NETBIOS/SMB

Этот стек широко используется в продуктах компаний IBM и Microsoft. На физическом и канальном уровнях этого стека используются все наиболее распространенные протоколы Ethernet, Token Ring, FDDI и другие. На верхних уровнях работают протоколы NetBEUI и SMB.

Протокол NetBIOS (Network Basic Input/Output System) появился в 1984 году как сетевое расширение стандартных функций базовой системы ввода/вывода (BIOS) IBM PC для сетевой программы PC Network фирмы IBM. В дальнейшем этот протокол был заменен так называемым протоколом расширенного пользовательского интерфейса NetBEUI — NetBIOS Extended User Interface. Для обеспечения совместимости приложений в качестве интерфейса к протоколу NetBEUI был сохранен интерфейс NetBIOS. Протокол NetBEUI разрабатывался как эффективный протокол, потребляющий немного ресурсов и предназначенный для сетей, насчитывающих не более 200 рабочих станций. Этот протокол содержит много полезных сетевых функций, которые можно отнести к сетевому, транспортному и сеансовому уровням модели OSI, однако с его помощью невозможна маршрутизация пакетов. Это ограничивает применение протокола NetBEUI локальными сетями, не разделенными на подсети, и делает невозможным его использование в составных сетях. Некоторые ограничения NetBEUI снимаются реализацией этого протокола NBF (NetBEUI Frame), которая включена в операционную систему Microsoft Windows NT.

Протокол SMB (Server Message Block) выполняет функции сеансового, представительного и прикладного уровней. На основе SMB реализуется файловая служба, а также службы печати и передачи сообщений между приложениями.

Стеки протоколов SNA фирмы IBM, DECnet корпорации Digital Equipment и AppleTalk/AFP фирмы Apple применяются в основном в операционных системах и сетевом оборудовании этих фирм.

Модель OSI	IBM/Microsoft	TCP/IP	Novell	Стек OSI
Прикладной	SMB	Telnet, FTP, SNMP,	NCP, SAP	X.400 X.500 FTAM

Представительный		SMTP, WWW		Представительный протокол OSI
Сеансовый	NetBIOS	TCP		Сеансовый протокол OSI
Транспортный			SPX	Транспортный протокол OSI
Сетевой		IP, RIP, OSPF	IPX, RIP, NLSP	ES-ES IS-IS
Канальный	802.3 (Ethernet), 802.5 (Token Ring), FDDI, Fast Ethernet, SLIP, 100VG-AnyLAN, X.25, ATM, LAP-B, LAP-D, PPP			
Физический	Коаксиал, экранированная и неэкранированная витая пара, оптоволокно, радиоволны			

*Рис. 1.30. Соответствие популярных стеков протоколов модели OSI*

На рис. 1.30 показано соответствие некоторых, наиболее популярных протоколов уровням модели OSI. Часто это соответствие весьма условно, так как модель OSI — это только руководство к действию, причем достаточно общее, а конкретные протоколы разрабатывались для решения специфических задач, причем многие из них появились до разработки модели OSI. В большинстве случаев разработчики стеков отдавали предпочтение скорости работы сети в ущерб модульности — ни один стек, кроме стека OSI, не разбит на семь уровней. Чаще всего в стеке явно выделяются 3-4 уровня: уровень сетевых адаптеров, в котором реализуются протоколы физического и канального уровней, сетевой уровень, транспортный уровень и уровень служб, вбирающий в себя функции сеансового, представительного и прикладного уровней.

### 3. ЗАДАНИЕ НА САМОСТОЯТЕЛЬНУЮ РАБОТУ

1. Перечислите функции физического уровня модели OSI
2. Перечислите функции канального уровня модели OSI
3. Перечислите функции сетевого уровня модели OSI
4. Перечислите функции транспортного уровня модели OSI
5. Перечислите функции сеансового уровня модели OSI
6. Перечислите функции представительного уровня модели OSI
7. Перечислите функции прикладного уровня модели OSI
8. Перечислите стандартные стеки коммуникационных протоколов
9. Перечислите функции прикладного уровня модели TCP/IP
10. Перечислите функции межхостового уровня модели TCP/IP
11. Перечислите функции межсетевого уровня модели TCP/IP
12. Перечислите функции уровня сетевого доступа модели TCP/IP

13. Перечислите известные вам стеки коммуникационных протоколов

## **5. КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Что такое протокол и чем он отличается от интерфейса?
2. Что такое интерфейс и чем он отличается от протокола?
3. Что такое стек коммуникационных протоколов?
4. В чем заключается разница между эталонной моделью TCP/IP и стеком TCP/IP?
5. Где используется стек NetBIOS/ SMB?
6. Где используется стек IPX/SPX?
7. Где используется стек TCP/IP?
8. Где используется стек OSI?
9. Что такое протокол NetBEUI и для чего он используется?
10. Что такое протокол SMB и для чего он используется?