ВЫСШЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ

Р. Л. СМЕЛЯНСКИЙ

КОМПЬЮТЕРНЫЕ СЕТИ

Учебник
В двух томах

ТОМ 2

СЕТИ ЭВМ

Допущено
Учебно-методическим объединением
по классическому университетскому образованию
в качестве учебника для студентов высших учебных заведений,
обучающихся по направлениям 010400 «Прикладная математика и информатика»
и 010300 «Фундаментальная информатика и информационные технологии»

ACADEMIA

Москва
Издательский центр «Академия»
2011

УДК 004.7(075.8)
ББК 32.973.202я73
С501

Рецензенты:
зав. кафедрой вычислительной техники МЭИ (ТУ), д-р техн. наук, проф. *В. В. Топорков*; декан факультета вычислительной математики и кибернетики Казанского госуниверситета, зав. кафедрой системного анализа и информационных технологий, проф. *Р. Х. Латыпов*;
зав. кафедрой теоретической информатики Ярославского государственного университета им. П. Г. Демидова, д-р физ.-мат. наук, проф. *В. А. Соколов*

Смелянский Р. Л.
С501 Компьютерные сети : учебник для студ. высш. учеб. заведений : в 2 т. Т. 2. Сети ЭВМ / Р. Л. Смелянский. — М. : Издательский центр «Академия», 2011. — 240 с.
ISBN 978-5-7695-7153-4

Рассмотрены эталонная модель взаимодействия открытых систем и модель TCP/IP, протоколы IPv4 и IPv6, адресация в Интернете, протоколы ARP и RARP, алгоритмы маршрутизации, протоколы RIP, RIPv2, OSPF, BGP, маршрутизация в сетях MPLS, оптимизация и функционирование транспортного уровня, протоколы TCP и UDP, основные вопросы безопасности информации в сетях ЭВМ, а также такие приложения, как DNS, SNMP, электронная почта, протокол FTP, организация и основные протоколы World Wide Web, понятия web-сервиса и основные компоненты архитектуры SOA.

Для студентов учреждений высшего профессионального образования.

УДК 004.7(075.8)
ББК 32.973.202я73

Оригинал-макет данного издания является собственностью Издательского центра «Академия», и его воспроизведение любым способом без согласия правообладателя запрещается

ISBN 978-5-7695-7153-4 (Т. 2)
ISBN 978-5-7695-7152-7

© Смелянский Р. Л., 2011
© Образовательно-издательский центр «Академия», 2011
© Оформление. Издательский центр «Академия», 2011

ПРЕДИСЛОВИЕ

Данный учебник написан на основе материалов лекций по курсу «Компьютерные сети», который читается на факультете «Вычислительная математика и кибернетика» (ВМиК) Московского государственного университета им. М. В. Ломоносова.

Учебник, прежде всего, предназначен для подготовки специалистов и бакалавров по направлениям «Прикладная математика и информатика» и «Фундаментальная информатика и информационные технологии», а также он может быть использован для подготовки специалистов и бакалавров по направлению «Информатика и вычислительная техника».

Основной целью курса «Компьютерные сети» т. 2 «Сети ЭВМ» является приобретение студентами знаний и навыков в следующих областях:

- теоретические основы архитектурной и системотехнической организации вычислительных сетей, построение сетевых протоколов;
- основы Интернет-технологий;
- методы и средства обеспечения информационной безопасности компьютерных сетей;
- конфигурирование локальных сетей, реализация сетевых протоколов с помощью программных средств.

В учебнике не рассматриваются сетевые операционные системы, поскольку, во-первых, сегодня само понятие сетевой операционной системы выглядит странно, так как любая современная операционная система предполагает наличие функциональности для поддержки сетевого взаимодействия, во-вторых, на эту тему уже достаточно много издано, а в-третьих, на факультете ВМиК курсу «Компьютерные сети» предшествуют курсы «Операционные системы», «Базы данных» и »Программирование на языке Java».

На факультете ВМиК МГУ при изучении курса «Компьютерные сети» используется практикум, включающий в себя пять лабораторных заданий. Организация изучения курса на факультете ВМиК также предусматривает систему промежуточного контроля знаний студентов.

Для этой цели было разработано около 1 200 контрольных вопросов и задач. (Автор готов поделиться презентационными материалами, а также материалами для практических упражнений и промежуточного контроля с желающими, для чего достаточно обратиться по адресу smel@CS. MSU. SU.)

По окончании изучения второй части рассматриваемого курса студенты должны иметь представление о современных системах международной и отечественной стандартизации, стандартах построения компьютерных сетей, направлениях развития сетей должны знать:

• эталонную модель взаимодействия открытых систем, основы сетевых стеков OSI и TCP/IP, схему организации и основные принципы функционирования современных сетей, методы и алгоритмы работы сетевых протоколов для локальных, городских и региональных сетей;

• принципы организации и функционирования сетевого уровня, протоколы IPv4 и IPv6, адресацию в Интернете, алгоритмы ARP и RARP маршрутизации RIP, RIPv2, OSPF, иерархическую маршрутизацию и протокол BGP, маршрутизацию в сетях MPLS;

• методы оптимизации функционирования транспортного уровня и настройки протоколов TCP и UDP;

• основные вопросы безопасности информации в сетях ЭВМ и методы их решения (включая алгоритмы и методы шифрования данных в сетях), вопросы аутентификации в сетях (включая организацию электронно-цифровой подписи), системы обнаружения атак, технологию VPN;

• основы функционирования прикладных протоколов в Интернете (таких как DNS, SNMP), организацию и основные протоколы функционирования электронной почты, протокол FTP, организацию и основные протоколы функционирования World Wide Web, понятия web-сервиса и основные компоненты архитектуры SOA .

Изучение данного курса предполагает наличие определенного объема знаний, т.е. студент должен знать архитектуру современных вычислительных систем, организацию и функционирование операционной системы Unix и ее файловой системы, организацию и функционирование систем управления базами данных, элементы теории массового обслуживания (теории очередей), основы дискретной математики в части теории графов и сетей, а также основы теории автоматов.

Также студент должен иметь практический опыт программирования (желательно знать язык Java) и уметь строить математические модели с использованием указанных ранее разделов математики.

Несколько слов необходимо сказать об истории создания данного курса. В мае 1995 г. перед автором была поставлена задача создать за лето курс по компьютерным сетям. Ясно, что за такой срок написать курс невозможно, поэтому я посвятил эти два месяца изучению соответствующей литературы, которую в то время можно было найти на книжных полках. Надо сказать, что выбор был не так уж и велик. Воспользовавшись командировкой в США, я провел несколько дней в книжных магазинах Сан-Франциско, выбирая учебник, который можно было бы взять за основу курса. Просмотрев немало книг, я

остановил свой выбор на книге Тененбаума (Tanenbaum A. S. Computer Networks, 2-nd edition, Prentice Hall, 1988) по следующим соображениям:

- широта охвата материала;
- сбалансированность глубины и общности подачи материала;
- наличие ясной и доступной логики изложения;
- обилие ссылок на первоисточники;
- хороший язык изложения;
- возможность получения электронных версий рисунков, что было немаловажно при подготовке презентаций для лекций.

Однако темпы развития сетевых технологий и глубины их проникновения в общество столь велики, что материал курса постоянно приходилось дополнять, включать в него новые технологии, новые сетевые средства, протоколы. Каждый год курс дополнялся и изменялся. Изменялась при этом и организация изучаемого материала. Так, например, к концу 1990-х гг. стало ясно, что системы передачи данных представляют собой самостоятельную часть сетей, предназначенную для формирования канала для сетевого уровня. При этом в них и возникает задача маршрутизации, но уровень ее сложности существенно ниже, чем на сетевом уровне. Используемые в этих системах решения ближе к вопросам телефонии и коммутации, чем к сложным алгоритмам маршрутизации с многокритериальной оптимизацией, которые используются на сетевом уровне, и к сложным алгоритмам управления транспортными соединениями на транспортном уровне. Совершенствуя курс, автор также постарался учесть тенденцию разделения процессов построения и оптимизации маршрута и коммутации потоков данных на сетевом уровне, нашедшую отражение в концепции MPLS-сетей. Эта тенденция вызвана все возрастающими размерами компьютерной сети и ростом вычислительных затрат на процессы маршрутизации.

Наконец, осталось самое приятное — поблагодарить всех тех, кто помогал автору, критиковал материал, способствуя улучшению курса. Прежде всего, хотелось бы поблагодарить студентов третьего потока факультета ВМиК, которые слушали курс в 1995—2008 гг. и своими замечаниями, критикой способствовали его улучшению. Автор также очень признателен Д. Гамаюнову, помогавшему работать с рукописью и предоставившему часть материалов по вопросам сетевой безопасности, А. Петухову, предоставившему материалы по Web-технологиям, Д. Козлову и М. Забежайло — за обсуждение и конструктивные предложения отдельных частей курса, Н. Трусову и Е. Капинус — за работу по подготовке рукописи. Особенно хотел бы отметить доброжелательность Ю. В. Коровина, без поддержки которого эта книга вряд ли появилась бы.

Москва
август 2010 *Р. Л. Смелянский*

Глава 1

МОДЕЛЬ И ПРИМЕРЫ СЕТЕЙ ЭВМ

1.1. Модель сети ЭВМ

1.1.1. Общие сведения

Организация сети ЭВМ (см. т. 1 данного учебника) представляет собой совокупность системы передачи данных (СПД), транспортной среды и абонентских машин (рис. 1.1). Напомним, что *A*-машины — это абонентские машины, на которых работают сетевые приложения, а *R*-машины — это маршрутизаторы, отвечающие за доставку сообщений между *A*-машинами. Система передачи данных и *R*-машины образуют транспортную среду (ТС) сети ЭВМ. Шлюзы *G* — это устройства, обеспечивающие сопряжение двух разных транспортных сред, а *B* — устройство, обеспечивающее сопряжение разных СПД.

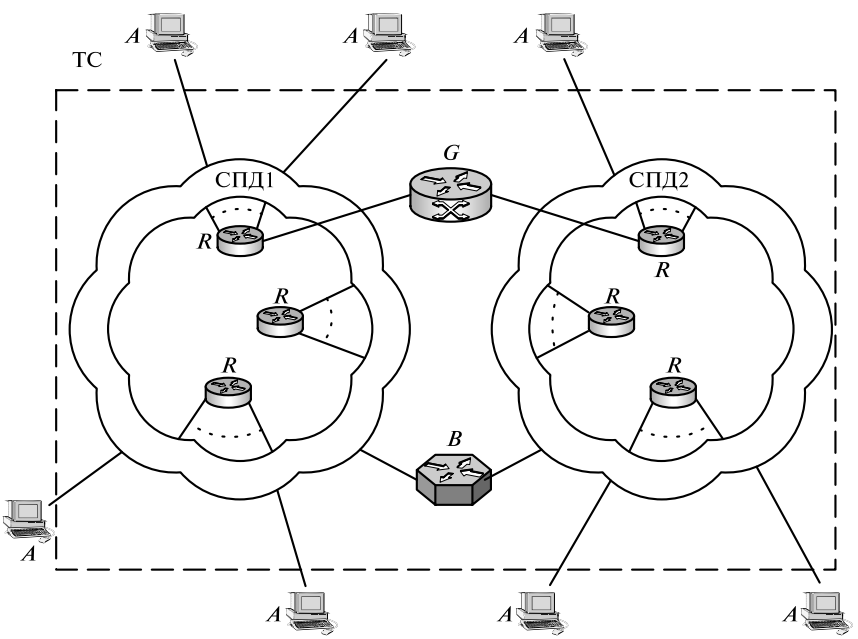

Рис. 1.1. Организация компьютерной сети

Система передачи данных — это совокупность каналов передачи данных и коммутирующих элементов.

В т. 1 данного учебника в качестве модели СПД были выбраны первые три уровня модели OSI с определенными оговоркам по функциям сетевого уровня, связанными с особенностями маршрутизации в СПД. В частности, обращалось внимание на то, что сетевой уровень в СПД сугубо внутренний. Сетевые уровни *A*-машин и *R*-машин с сетевым уровнем в СПД не взаимодействуют.

Теперь мы рассмотрим ряд существующих моделей и на их основе построим модель функционирования современной сети ЭВМ.

1.1.2. Эталонная модель OSI

Напомним, что модель взаимодействия открытых систем — OSI (Open Systems Interconnection), представленная на рис. 1.2, была разработана для определения международных стандартов компьютерных сетей. Эта модель описывает, как должна быть организована

Рис. 1.2. Модель взаимодействия открытых систем

система, открытая для взаимодействия с другими системами. Модель OSI имеет уровневую организацию, т.е. она включает в себя семь уровней: физический, канальный, сетевой, транспортный, сессии, представления и прикладной. Функции физического уровня и уровня канала данных подробно рассмотрены в т. 1 данного учебника, так же, как и функции сетевого уровня, применительно к системам СПД.

Ключевыми в этой модели являются понятия сервиса, интерфейса и протокола. Под сервисом понимают услуги, которые нижерасположенный уровень оказывает по запросам вышерасположенного. В этой модели нижерасположенный уровень свои услуги может предоставлять только вышерасположенному уровню. Интерфейс определяет формирование и передачу запроса на услугу.

Активные элементы уровня, т.е. элементы, которые могут сами совершать действия, в отличие от элементов, над которыми совершают действия, называются *активностями*. Активности могут быть программными и аппаратными. Активности одного и того же уровня на разных машинах называются *равнозначными*, или *одноименными*. Активности уровня $n + 1$ являются *пользователями сервиса*, создаваемого активностями уровня n, которые, в свою очередь, называются *поставщиками сервиса*. Сервис может быть разного качества. Например, с установлением соединения и без установления соединения, с подтверждением получения переданных данных и без подтверждения. Правила и соглашения по установлению соединения, его поддержанию и обмену данными по нему между активностями, расположенными на одинаковом уровне на разных машинах, называется *протоколом*.

Доступ к сервису в модели OSI осуществляется через так называемые *точки доступа к сервису* — SAP (Service Access Points), каждая из которых имеет уникальный адрес. Например, телефонная розетка на стене — это точка доступа к сервису автоматической телефонной станции (АТС). Каждой такой розетке сопоставлен определенный номер — номер телефона.

Взаимодействие между двумя соседними уровнями в этой модели можно описать следующим образом: активность на уровне $n + 1$ передает интерфейсную единицу данных — IDU (Interface Data Unit) активности на уровне n через SAP (рис. 1.3). IDU состоит из сервисной единицы данных — SDU (Service Data Unit) — и управляющей информации. SDU передается далее по сети равнозначной сущности, а затем — на уровень $n + 1$. Управляющая информация необходима нижерасположенному уровню, чтобы правильно передать SDU, но она не является частью передаваемых данных.

Чтобы передать SDU по сети нижерасположенному уровню, может потребоваться разбить ее на части (рис. 1.4). При этом каждая часть снабжается *заголовком* и *концевиком* и передается как самостоятельная единица данных протокола — PDU (Protocol Data Unit). Заголовок в PDU используется протоколом при передаче. В нем указывает-

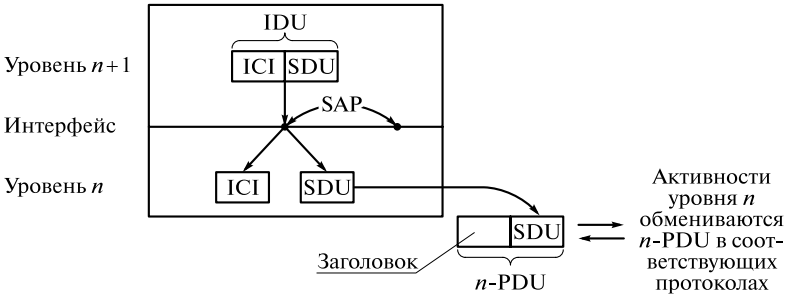

Рис. 1.3. Взаимосвязь уровней Сети через интерфейс:

SAP — точка доступа к сервису; IDU — интерфейсная единица данных; SDU — сервисная единица данных; PDU — единица данных протокола; ICI — контрольная информация интерфейса

ся, какой PDU содержит управляющую информацию, а какой — данные, порядковый номер PDU и т.д.

Формально сервис можно описать в терминах примитивных операций, или *примитивов*, с помощью которых пользователь или какая-либо активность получает доступ к сервису. С помощью этих примитивов активность вышерасположенного уровня сообщает активности нижерасположенного уровня, что необходимо сделать, чтобы вышерасположенная активность получила необходимую услугу (сервис). В свою очередь, нижерасположенная активность может использовать эти примитивы, чтобы сообщить вышерасположенной активности о выполненном действии.

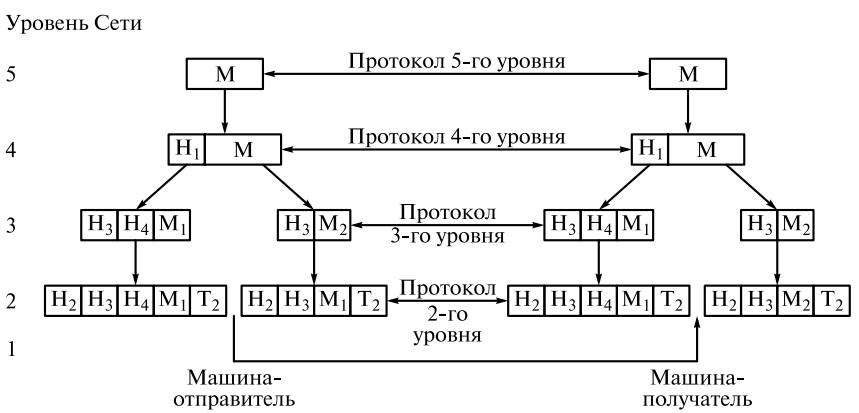

Рис. 1.4. Пример организации потока информации при виртуальном взаимодействии на 5-м уровне Сети:

M — сообщения; H_i — заголовок i-го уровня; T_i — концевик i-го уровня

Таблица 1.1

Классы примитивов сервиса

Класс примитивов	Значение
Request (Запрос)	Запрос определенных действий
Indication (Индикация)	Информирование о каком-либо событии
Response (Ответ)	Ответ на полученный ранее запрос
Confirm (Подтверждение)	Подтверждение ответа на сделанный ранее запрос

Примитивы можно подразделить на четыре класса, показанные в табл. 1.1. Эти классы подробно рассмотрены в подразд. 2.3.4. т. 1 данного учебника.

Сетевой уровень

Основная проблема, решаемая на сетевом уровне, — это выбор оптимального маршрута для пакета от отправителя к получателю. Для решения этой проблемы требуется решить две задачи: вычислить оптимальный маршрут и реализовать этот маршрут, т.е. обеспечить следование пакета именно по рассчитанному маршруту. При этом маршрут может быть определен заранее и прописан в таблице в каждом узле коммутации, которая не изменяется. Маршрут может быть определен в момент установления соединения, и тогда все пакеты, направляемые по данному соединению, будут следовать этим маршрутом. Наконец, маршрут может строиться динамически по ходу передачи в зависимости от загрузки сети и ее текущей конфигурации. Критерии для определения оптимальности маршрута могут быть разными в зависимости от конкретной ситуации.

По понятным причинам в сети ЭВМ не может быть единого центра расчета маршрута, поэтому может оказаться, что в транспортной среде циркулирует слишком много пакетов, использующих одни и те же маршруты либо их фрагменты, что будет приводить к перегрузкам каналов СПД, а следовательно, и к потере пакетов. Эта проблема также решается на сетевом уровне. Причем эту проблему, называемую перегрузкой в сети, не следует путать с проблемой управления потоком, заключающейся в том, чтобы отправитель не «заваливал» пакетами получателя. Различие и методы решения этих проблем подробно рассмотрим далее.

Как уже говорилось, некоторые виды сетевого сервиса предполагают, что каждый пакет движется по своему отдельному маршруту. Это означает, что достигать сетевого уровня получателя они будут в разное время. При этом никто не гарантирует, что ранее отправленный пакет придет раньше отправленного позднее. Следовательно, на

сетевом уровне необходимо уметь восстанавливать исходную последовательность пакетов.

Пакеты при передаче могут теряться и дублироваться (причины этого рассмотрим далее). Сетевой уровень должен уметь бороться с подобными явлениями и исправлять возникающие нарушения.

Как видно из рис. 1.2, сетевой уровень является границей транспортной среды. Поскольку за использование транспортной среды, как правило, предполагается оплата, то на этом уровне присутствуют функции учета числа байтов (или символов), которые послал и получил абонент сети. Причем, если абоненты расположены в разных странах, где действуют различные тарифы, необходимо должным образом скорректировать цену услуги.

Если пакет адресован в другую сеть, т.е. в другую транспортную среду, то при его передаче требуется предпринять надлежащие меры для учета различия форматов пакетов в разных транспортных средах, различия способов адресации, допустимых размеров пакетов и т.д. Все эти проблемы решаются шлюзом (см. устройство *G* на рис. 1.1) на сетевом уровне.

Транспортный уровень

Основная функция транспортного уровня — принять данные с вышерасположенного уровня сессии, разделить их, если требуется, на более мелкие единицы — сегменты, передать на сетевой уровень и позаботиться, чтобы все они дошли в целости до адресата. Все это необходимо сделать эффективно и так, чтобы вышерасположенный уровень не зависел от того, как именно это было сделано. В нормальных условиях транспортный уровень должен создавать специальное сетевое соединение для каждого транспортного соединения по запросу уровня сессии. Если для транспортного соединения требуется высокая скорость передачи, то транспортный уровень может потребовать у сетевого уровня создания нескольких сетевых соединений, между которыми транспортный уровень будет распределять передаваемые данные. И наоборот, если требуется обеспечить недорогое транспортное соединение, то транспортный уровень может использовать одно и то же соединение на сетевом уровне для нескольких транспортных соединений. В любом случае такое мультиплексирование должно быть незаметным на уровне сессии.

Сетевой уровень определяет, какой тип сервиса предоставить вышерасположенным уровням и пользователям сети. Наиболее часто используемым сервисом является канал типа «точка—точка», без ошибок обеспечивающий доставку сообщений или байтов в той последовательности, в какой они были отправлены. Другим видом сервиса является доставка отдельных сообщений без гарантии сохранения их последовательности или, например, рассылка одного

сообщения многим в режиме вещания. В каждом конкретном случае сервис определяют при установке транспортного соединения.

Транспортным называется уровень, обеспечивающий соединение типа точка — точка. Активности транспортного уровня на машине отправителя общаются с равнозначными активностями транспортного уровня на машине получателя. При этом машина-отправитель и машина-получатель не обязаны быть соседними (см. рис. 1.2). Этого нельзя сказать про активности на нижерасположенных уровнях, которые общаются с равнозначными активностями на соседних машинах. В этом заключается одно из основных отличий уровней 1 ... 3 от уровней 4 ... 7. Последние обеспечивают соединение типа точка — точка, что хорошо видно из рис. 1.2.

Внимательно сравнив рис. 1.2 с рис. 1.2 из т. 1 данного учебника, увидим, что если *A*-машины с *R*-машинами взаимодействуют на физическом, канальном, сетевом и других уровнях, то с СПД они взаимодействуют только на физическом и канальном уровнях. Сетевой уровень СПД сугубо внутренний.

Многие *A*-машины мультипрограммные, поэтому транспортный уровень для одной такой машины должен поддерживать несколько транспортных соединений. Чтобы определить, к какому соединению относится тот или иной пакет, в его заголовке (см. H_4 на рис. 1.4) помещается необходимая информация.

Транспортный уровень также отвечает за установление и разрыв транспортного соединения в сети. Это предполагает наличие механизма именования, т. е. процесс на одной машине должен уметь указать, с кем в сети ему следует обменяться информацией. Транспортный уровень также должен предотвращать «захлебывание» получателя при обращении с «очень быстро говорящим» отправителем. Механизм такого предотвращения, называемый *управлением потоком*, имеется и на других уровнях. Однако, как мы увидим далее, управление потоком между *A*-машинами в сети отличается от управления потоком между *R*-машинами (маршрутизаторами).

Уровень сессии

В модели ISO уровень сессии позволяет пользователям *A*-машин (напомним, что пользователем может быть программа) устанавливать между собой *сессии*. Сессия позволяет передавать данные, как это может делать транспортный уровень, а кроме того, этот уровень имеет более сложный сервис, полезный в некоторых приложениях. Например, на уровне сессии можно осуществлять вход в удаленную систему, передавать файл между двумя приложениями и т. д.

Одним из видов услуг на этом уровне является *управление диалогом*. Потоки данных здесь могут быть разрешены одновременно в обоих направлениях либо поочередно в одном направлении. Сервис на уровне сессии будет управлять направлением передачи.

Другой вид сервиса на уровне сессии — *управление маркером*. Для некоторых протоколов недопустимо выполнение одной и той же операции на обоих концах соединения одновременно. В этом случае уровень сессии выделяет активной стороне маркер, и операцию может выполнять только тот, кто владеет маркером.

Также примером сервиса на уровне сессии является *синхронизация*. Пусть требуется передать файл, время пересылки которого составит два часа, между машинами, время наработки на отказ у которых — один час. Ясно, что «в лоб» передачу такого файла средствами транспортного уровня не решить. Уровень сессии позволяет расставлять контрольные точки. В случае отказа одной из машин передача возобновится с последней контрольной точки.

Уровень представления

В модели ISO уровень представления обеспечивает решение часто возникающих проблем, связанных с представлением данных при передаче. В основном это проблемы семантики и синтаксиса передаваемой информации. Данный уровень имеет дело с информацией, а не с потоком битов.

Типичным примером услуги на этом уровне является унифицированная кодировка данных. Дело в том, что на разных машинах используются разные способы кодировки символов, например ASCII, Unicode и другие, а также разные способы представления целых: в прямом, обратном или дополнительном коде. Нумерация битов в байте на разных машинах одной и той же сети тоже может быть разная, т.е. слева направо или справа налево, и т.д.

Пользователи, как правило, используют структуры данных, а не случайный набор байтов. Чтобы машины с разными кодировкой и представлением данных могли взаимодействовать, передаваемые структуры данных определяются специальным абстрактным способом, не зависящим от кодировки, используемой при передаче. Уровень представления работает со структурами данных в абстрактной форме, преобразует это представление во внутреннее представление для конкретной машины и из внутреннего машинного представления в стандартное представление для передачи по сети.

Уровень приложений

Уровень приложений обеспечивает работу часто используемых приложений, например передачу файлов.

Разные операционные системы используют разные механизмы именования, представления текстовых строк и т.д. Для передачи файлов между разными системами необходимо преодолевать подобные различия. Чтобы пользователям не приходилось каждый раз заново бороться с этими трудностями на уровне приложений имеется приложение FTP.

На уровне приложений находятся также такие часто используемые приложения, как электронная почта, удаленная загрузка программ, удаленный просмотр информации и т. д.

1.1.3. Модель TCP/IP

Рассмотрим другую эталонную модель, прототипом для которой послужил прародитель Интернета — сеть ARPA. Далее будет приведена история создания этой сети, а сейчас лишь отметим, что она была создана в результате, научно-исследовательских работ, проведенных по инициативе Министерства обороны США. Позднее к этому проекту подключились сотни университетов и государственных учреждений США.

С самого начала эта сеть задумывалась как объединение нескольких разных сетей.

Одной из основных целей этого проекта была разработка унифицированных способов соединения сетей для создания систем передачи данных, обладающих высокой живучестью. Под живучестью понимается способность системы сохранять в приемлемых пределах качество и объемы предоставляемого сервиса при выходе из строя ее компонентов. Так появилась модель TCP/IP, получившая название по именам двух основных протоколов: протокола управления передачей — TCP (Transmission Control Protocol) и межсетевого протокола — IP (Internet Protocol).

Другой целью проекта ARPA было создание протоколов, не зависящих от характеристик конкретных *A*-машин, маршрутизаторов, шлюзов и т. п. Кроме того, связь должна была поддерживаться, даже если отдельные компоненты сети стали выходить из строя во время соединения. Другими словами, связь в этой сети должна поддерживаться до тех пор, пока источник информации и получатель информации работоспособны. Архитектура сети ARPA не должна была ограничивать приложения, начиная от простой передачи файлов до передачи речи и изображения в реальном времени.

Межсетевой уровень

В силу приведенных требований выбор организации транспортной среды был очевиден: сеть с коммутацией пакетов с межсетевым уровнем без соединений. Такой уровень, называемый *межсетевым*, является основой всей архитектуры сети. Его назначение — обеспечить доставку пакетов, движущихся в сети независимо друг от друга, даже если получатель принадлежит другой сети. Причем пакеты могут поступать к получателю не в том порядке, в котором они были посланы. Упорядочить их в надлежащем порядке — задача вышерасположенного уровня.

Межсетевой уровень определяет межсетевой протокол IP и формат пакета. Обратим внимание, что ни протокол, ни формат пакета не являются официальными международными стандартами в отличие от протоколов эталонной модели OSI, в которой большинство протоколов имеют статус международных стандартов.

Итак, назначением межсетевого уровня в модели TCP/IP является доставка IP-пакета по назначению и по оптимальному маршруту. Критерии оптимизации маршрута могут быть самые разные: например, самый короткий маршрут или самый быстрый, или самый дешевый, или такой, который не проходит через определенные сети, расположенные на недружественных территориях, и т.д. Это как раз то, за что отвечает сетевой уровень в OSI-модели.

Под межсетевым уровнем в TCP/IP-модели «великая пустота». Эта модель никак не регламентирует организацию и функционирование СПД, равно как и связь с ней. Она также ничего не говорит о том, что в СПД происходит, кроме того, что хост-машина* (в данном случае *A*-машина или *R*-машина) должна быть связана с сетью через некоторый протокол. Никаких ограничений на этот протокол, а также рекомендаций в этой модели нет.

Транспортный уровень

Над межсетевым уровнем в модели TCP/IP расположен *транспортный уровень*. Как и в OSI-модели, задача этого уровня — обеспечить связь типа точка—точка между двумя равнозначными активностями. В рамках модели TCP/IP было разработано два транспортных протокола. Первый протокол — TCP (Transmission Control Protocol), т.е. надежный протокол с соединением, который получает поток байтов, фрагментирует его на отдельные сообщения и передает их на межсетевой уровень. На машине-получателе равнозначная активность TCP-протокола собирает эти сообщения в поток байтов. TCP-протокол также обеспечивает управление потоком.

Второй протокол — UDP (User Datagram Protocol), т.е. ненадежный протокол без соединения для тех приложений, которые используют свои механизмы фрагментации и управления потоком. Этот протокол часто используется для передачи коротких сообщений в клиент-серверных приложениях, а также там, где скорость передачи важнее ее аккуратности. Напомним, что ненадежным называется протокол без уведомления о получении кадра, фрейма, пакета, сообщения и т.п.

Уровень приложений

В модели TCP/IP нет уровней сессии и представления, поскольку необходимость в них была неочевидна для ее создателей. В настоящее

* Хостом обычно называют машину в сети, у которой есть IP-адрес

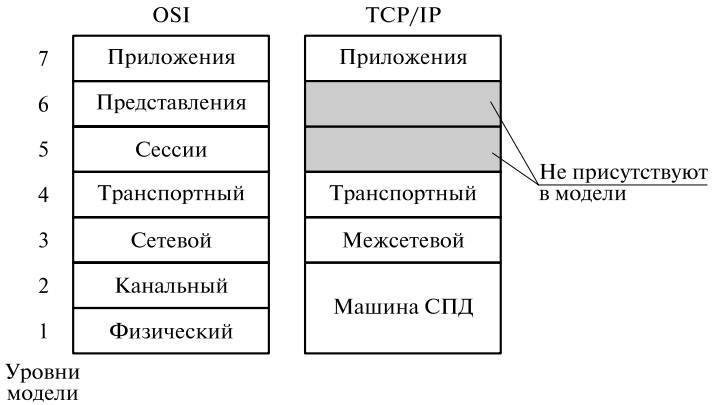

Рис. 1.5. Соответствие эталонных моделей OSI и TCP/IP

время разработчик сложного приложения берет реализацию функций этих уровней на себя.

Уровень приложений располагается сразу над транспортным. Этот уровень сначала включал в себя следующие приложения: передача файлов — FTP, электронная почта — SMTP. Позднее к ним добавились служба имен домена — DNS (Domain Name Service), отображающая логические имена *A*-машин на их сетевые адреса, протокол для работы с гипертекстовыми документами во Всемирной паутине — HTTP и некоторые другие.

На рис. 1.5 показано соответствие уровней двух рассмотренных эталонных моделей.

1.1.4. Выбор модели

Теперь необходимо выбрать модель, которую мы будем использовать для дальнейшего изучения сетей ЭВМ. Сразу оговоримся, что разнообразие моделей сетей не исчерпывается уже рассмотренными. Например, имеется модель SNA (System Network Architecture) от компании IBM и модель DNA (Digital Network Architecture) от компании DEC. Все указанные модели имеют много общего. Например, модель SNA была положена в основу модели OSI. Однако наиболее популярны модели OSI и TCP/IP.

Модели TCP/IP и OSI имеют много общего. Обе эти модели имеют уровневую организацию и поддерживают понятие стека протоколов. Назначение их уровней примерно одинаковое. Все уровни этих моделей от транспортного и ниже используют протоколы для поддержки взаимодействия типа точка—точка, не зависящего от организации СПД, а все уровни выше транспортного ориентированы на приложения.

Наибольшее значение модели OSI методологическое: в ней явно определены и четко выделены понятия сервиса, интерфейса, протокола, уровня. Это разделение строго проведено сверху донизу. Сервис определяет, что делает уровень, но ничего не говорит о том, как он это делает. Интерфейс уровня определяет для вышерасположенного уровня доступ к сервису. Протокол определяет реализацию сервиса.

Здесь можно провести аналогию с объектно-ориентированным программированием. У каждого объекта имеется набор методов — сервис, который определяет те операции, которые этот объект может выполнять. Иными словами, сервис — это семантика методов. Каждый метод имеет интерфейс: набор параметров, имя и т. п. Реализация методов скрыта в объекте (протоколе) и невидима пользователю.

В модели TCP/IP нет столь же четкого выделения этих понятий. В ней понятие протокола оторвано от остальных частей, и в ней нет единой, хорошо продуманной, концепции построения. Этот факт является следствием того, как создавались эти модели.

Модель TCP/IP создавалась post factum, т. е. после реализации основных протоколов, а модель OSI — до начала их реализации, поэтому понятие протокола в ней абсолютно не зависит от остальных частей модели. Например, изначально протоколы канального уровня в модели OSI создавались для соединений типа точка—точка. Позднее, когда появились каналы с множественным доступом, на этот уровень были добавлены соответствующие протоколы. Никаких других изменений не последовало.

Модель TCP/IP была создана, когда стек TCP/IP уже существовал, поэтому:

- в этой модели нет четкого разграничения понятий сервиса, интерфейса и протокола;
- эта модель годится только для описания стека TCP/IP;
- уровень хост—сеть в ней по существу уровнем не является, это больше интерфейс;
- в этой модели отсутствуют уровни, соответствующие СПД.

Модели OSI и TCP/IP имеют разное число уровней. Обе они имеют уровень приложений, транспортный уровень и сетевой уровень. Все остальные уровни в них разные. Модель OSI поддерживает на сетевом уровне и сервис с соединением, и сервис без соединения. На транспортном уровне этой модели поддерживается сервис только с соединением. В модели TCP/IP наоборот: сетевой уровень обеспечивает сервис без соединения, а транспортный — и с соединением, и без соединения.

По существу модель OSI доказала свою эффективность как методологический инструмент и стала популярной, чего нельзя сказать о ее протоколах, а с TCP/IP все наоборот: модели по существу нет, а протоколы получили широкое распространение.

В данном курсе мы будем использовать понятия и организацию модели OSI, а изучать протоколы сетевого, транспортного и при-

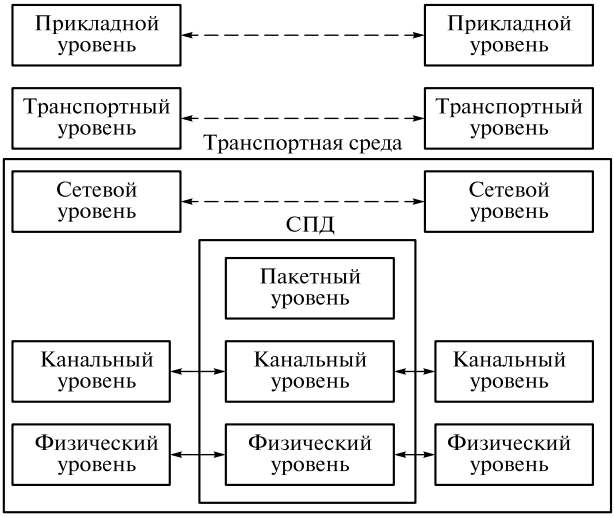

Рис. 1.6. Структура и организация гибридной модели, выбранной для дальнейшего изучения сетей ЭВМ

кладного уровней, следуя модели TCP/IP. Протоколы физического и канального уровней уже рассматривались в т. 1 данного учебника.

Гибридная модель, выбранная для дальнейшего изучения сети ЭВМ, представлена на рис. 1.6. Эта модель состоит из следующих пяти уровней: физического, канала данных, сетевого, транспортного и прикладного. При изучении будем использовать понятия уровня, сервиса, интерфейса, протокола, следуя модели OSI, откуда мы возьмем распределение сервисов и функций между уровнями, а набор протоколов и типовых приложений возьмем из модели TCP/IP.

1.2. Стандартизация в сети Интернет

В т. 1 данного учебника была подробно рассмотрена организация международной стандартизации. Поскольку одним из основных компонентов выбранной нами модели сети является TCP/IP, на которой основывается Интернет, то необходимо рассмотреть, кто есть кто в мире стандартизации Интернета.

Процесс стандартизации для сети Интернет с самого ее зарождения был хорошо организован. Этот процесс начался в 1969 г. с момента начала стандартизации протоколов сети ARPANET и всегда имел значительную финансовую поддержку как по линии государственного бюджета США, так и благодаря спонсированию этой деятельности промышленностью и бизнесом. В частности, такие компании как, например, APNIC, ARIN, Cisco Systems, IBM, Microsoft,

Ripe NCC ежегодно выделяют по 100 тыс. долл. для поддержки развития системы стандартов сети Интернет.

За процесс стандартизации информационных технологий в Интернете отвечают организации ISOC, IAB, IETF, IRTF, IESG [18], которые структурно взаимосвязаны следующим образом: на верхнем уровне иерархии в рассматриваемой организационной структуре располагается ISOC, а ниже расположены IAB, IETF, IRTF и IESG, отвечающие за отдельные направления работ.

ISOC (Internet Society — интернет-сообщество, www.isoc.org) — ассоциация экспертов, отвечающая за разработку стандартов технологий сети Интернет. В рассматриваемой организационной структуре ISOC располагается на верхнем уровне иерархии. Это некоммерческая неправительственная международная профессиональная организация, членами которой являются 175 организаций и около 9000 физических лиц из более чем 170 стран мира.

Работа ISOC сосредоточена на решении следующих основных задач:

- организация процесса стандартизации технологий сети Интернет;
- формирование внешней политики интернет-сообщества;
- поддержка инфраструктуры (организационно-административное управление деятельностью, управление финансами, защита прав интеллектуальной собственности и др.);
- образование и обучение, в том числе организация ежегодных семинаров по обучению интернет-технологиям (Network Training Workshops — NTW), организация системы учебных центров (Sustainable Internet Training Centers — SITCs) и др.;
- поддержка членства в ISOC как для организаций, так и для персональных членов.

IAB (Internet Architecture Board — Совет по архитектуре сети Интернет) — группа технических советников в составе ISOC, отвечающая за развитие архитектуры сети Интернет, управление разработкой и сопровождение стандартов протоколов и сервисов в Интернете и, прежде всего, спецификаций стека протоколов TCP/IP. Данный совет несет ответственность за управление редактированием и публикацией спецификаций RFC (Request for Comments), осуществляемое издательским органом RFC Editor (http://www.rfc-editor.org), а также за управление присваиванием номеров спецификациям RFC.

Результаты стандартизации в сети Интернет публикуются в виде RFC-документов. Эти документы являются доступными на файловых серверах Интернета для всех специалистов, что обеспечивает открытость процессу стандартизации. Каждой вновь разработанной спецификации или редакции уже существующего RFC-документа присваивается очередной свободный номер RFC-документа. Предыдущие версии пересмотренного документа остаются в каталогах системы стандартов с прежними номерами, но помечаются как изменявшиеся.

IAB выполняет представительские функции ISOC при взаимодействии с другими организациями. Деятельность IAB поддерживается

напрямую и косвенно как правительством США, так и промышленностью.

IETF (Internet Engineering Task Force — рабочая группа по проектированию Интернет-технологий, www.ietf.org) по существу является большим международным открытым сообществом разработчиков, операторов, изготовителей и исследователей в области сетевых технологий, занимающихся вопросами развития архитектуры сети Интернет и способов ее использования. IETF открыта для всех, кто интересуется интернет-технологиями. Основная сфера деятельности IETF состоит собственно в разработке стандартов сети Интернет, а также в их эффективной реализации и тестировании.

IRTF (Internet Research Task Force — исследовательская группа интернет-технологий, www.irtf.org) — подразделение IAB, которое выполняет долгосрочные исследовательские программы, связанные с вопросами развития архитектуры, базовых протоколов и сетевых приложений сети Интернет. Руководящие органы IRTF назначаются IAB.

IESG (Internet Engineering Steering Group — группа технического управления сети Интернет, www.ietf.org) отвечает за техническое управление процессом стандартизации интернет-технологий, осуществляет экспертизу проектов спецификаций, разрабатываемых IETF, несет ответственность за принятие интернет-стандартов и их дальнейшее продвижение.

Более подробно вопросы стандартизации в сети Интернет изложены в [17].

1.3. Примеры сетей ЭВМ

1.3.1. ARPANET

В начале 1960-х гг. Министерство обороны США поставило задачу создания командных пунктов и пунктов управления связью, которые были бы способны сохранить работоспособность при ядерной войне. Обычные телефонные линии были ненадежны: поражение АТС района означало потерю связи со всеми абонентами этого района. Для решения этой задачи Министерство обороны США обратилось к своему агентству перспективных разработок — ARPA (Advanced Research Project Agency, иногда DARPA), не имеющему лабораторий, научных сотрудников и т. п. Это бюрократическая организация с самостоятельным бюджетом, из которого она выделяет гранты университетам и компаниям, если их идеи кажутся им интересными.

В 1962 г. исследования ARPA по вопросам военного применения компьютерных технологий возглавил доктор Ликлайдер, который предложил для этих целей использовать взаимодействие имеющихся государственных компьютеров, а также способствовал привлечению

к этим работам частного сектора и университетских ученых. В этом же году появился отчет, выполненный П. Бараном в корпорации RAND по заказу военно-воздушных сил [70], в котором исследовались различные модели коммуникационных систем и оценивалась их живучесть. В отчете предлагалась децентрализованная система управления и связи, которая могла бы функционировать при выходе из строя какой-то ее части. Одна из рекомендаций автора касалась построения системы передачи цифровых данных между большим числом пользователей.

Вскоре основным направлением проводимых агентством исследований стали компьютерные сети. Главная идея состояла в построении сети из равноправных узлов, каждый из которых должен был иметь собственные блоки приема, обработки и формирования сообщений, обеспечивающие высокую живучесть сети даже при выходе из строя множества узлов. Первые эксперименты по объединению удаленных узлов были проведены уже в 1965 г., когда были соединены компьютеры TX-2 Массачусетского технологического института (MIT Lincoln Lab) и Q-32 корпорации SDC (System Development Corporation) в Санта-Монике. Правда, обмена пакетами между ними в это время еще не проводилось, обмен осуществлялся посимвольно.

В 1967 г. на симпозиуме ACM (Association for Computer Machinery) был представлен план создания национальной сети с коммутацией пакетов. Вскоре после симпозиума Робертс опубликовал план построения такой сети — ARPANET (Advanced Research Projects Agency NETwork), и уже в 1969 г. Министерство обороны США утвердило этот проект. Первым узлом новой сети стал университет штата Калифорния в Лос Анжелесе (UCLA) — Центр испытаний сети, а вскоре к нему присоединились Стэнфордский исследовательский институт (SRI), университет штата Калифорнии в Санта-Барбара (UCSB) и университет штата Юта. На узлах этой сети использовались IMP (Interface Message Processor) — машины, разработанные корпорацией Bolt Baranec & Newman, Inc (BBN). Так были осуществлены первые передачи знаков из одних машин в другие.

Согласно плану Робертса каждая подсеть должна была содержать как минимум один IMP-компьютер, соединенный линиями связи с двумя другими такими компьютерами в других подсетях. В подсети должен был использоваться дейтаграммный способ передачи данных. Программное обеспечение состояло из следующих протоколов: IMP — хост, IMP — IMP, IMP-отправитель — IMP-получатель (рис. 1.7).

Позднее для подключения терминалов к сети был создан вариант IMP, названный TIP (Terminal Interface Processor). Затем к одному IMP стали подключать несколько хост-машин, и одна хост-машина получила возможность соединяться с несколькими IMP. После этого начались эксперименты со спутниковой связью и радиосвязью. Например, был поставлен следующий эксперимент: грузовик с обо-

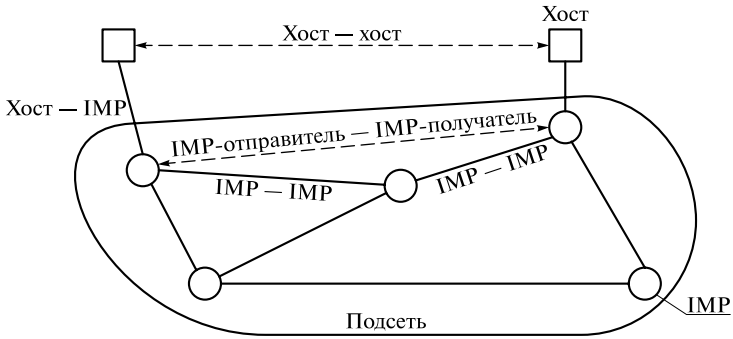

Рис. 1.7. Организация сети APRANET

рудованием двигался по дорогам южной Калифорнии, пытаясь связаться с машинами, находящимися в исследовательском институте Стэнфорда (SRI), которые через спутниковый канал передавали сообщения в Лондон.

В результате проведенных экспериментов стало ясно, что имеющиеся протоколы плохо пригодны для межсетевой связи, поэтому после выполнения дополнительных работ в 1974 г. появился стек протоколов TCP/IP. Контракты на реализацию этих протоколов ARPA передало в университет Беркли, где шли работы над операционной системой UNIX. Версия UNIX BSD 4.3 с реализацией TCP/IP, сокетов и другого сетевого программного обеспечения стала быстро распространяться.

К 1982 г. ARPANET уже насчитывала более 200 IMP-машин.

1.3.2. Хронология развития сети Интернет

1982 г. — рождение современного Интернета. TCP/IP стал официальным протоколом в ARPANET.

1984 г. — число хостов в ARPANET превысило тысячу.

1986 г. — Национальный фонд науки США (The National Science Foumdation) создал сеть NSFNET, открывшую доступ через Интернет к суперкомпьютерным центрам. После этого число сетей, подключенных к ARPANET, стало расти очень быстро. Во второй половине 1980-х гг. этот конгломерат сетей стали рассматривать как сеть сетей, а позднее как Интернет. В значительной степени рост сети Интернет происходил за счет подключения таких сетей, как SPAN — сеть космической физики NASA, HEP — сеть физики высоких энергий, BITNET — сеть машин класса mainframe фирмы IBM, EARN — европейская сеть научно-исследовательских организаций.

1989 г. — рождение идеи Всемирной паутины (World Wide Web — WWW), инициатор — британский ученый Тим Бернерс-Ли.