Для системных администраторов UNIX

3-е издание

TCP/IP

Сетевое администрирование

Символ®

O'REILLY®

Крэйг Хант

TCP/IP

Network Administration

Third Edition

Craig Hunt

O'REILLY®

TCP/IP

Сетевое администрирование

Третье издание

Крэйг Хант

Санкт-Петербург — Москва
2004

Крэйг Хант

TCP/IP. Сетевое администрирование, 3-е издание

Перевод М. Зислиса

Главный редактор	*А. Галунов*
Зав. редакцией	*Н. Макарова*
Научный редактор	*С. Маккавеев*
Редакторы	*А. Лосев, А. Петухов*
Корректор	*С. Беляева*
Верстка	*Н. Гриценко*

Крэйг Хант
TCP/IP. Сетевое администрирование, 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 816 с., ил.
ISBN 5-93286-056-1

Третье издание книги «TCP/IP. Сетевое администрирование» – это полноценное руководство по настройке и сопровождению сети TCP/IP, которое предназначается как системным администраторам, так и пользователям домашних компьютеров с доступом к сети Интернет. Повествование начинается с основ: зачем нужны протоколы, как они работают, как адресация и маршрутизация позволяют передавать данные по сети и как настроить сетевое соединение.

Помимо базовой настройки книга рассказывает о современных протоколах маршрутизации (RIPv2, OSPF и BGP) и пакете gated, который реализует работу с ними. Кроме того, книга является руководством по настройке многих важных сетевых служб, в том числе DNS, Apache, sendmail, Samba, PPP и DHCP. Две главы посвящены безопасности и разрешению проблем. Третье издание включает новую главу, посвященную настройке сервера Apache и раздел, в котором обсуждается настройка Samba с целью организации совместного доступа к файлам и принтерам в гетерогенной сети Unix/Windows. Справочные приложения подробно описывают синтаксис таких программ, как gated, pppd, named, dhcpd и sendmail. Книга охватывает реализации TCP/IP для систем Linux, Solaris, BSD и System V.

ISBN 5-93286-056-1
ISBN 0-596-00297-1 (англ)

Издательство «Символ-Плюс». 199034, Санкт-Петербург, 16 линия, 7,
тел. (812) 324-5353, edit@symbol.ru. Лицензия ЛП N 000054 от 25.12.98.
Налоговая льгота – общероссийский классификатор продукции
ОК 005-93, том 2; 953000 – книги и брошюры.
Подписано в печать 08.01.2004. Формат 70x100 1/16 . Печать офсетная.
Объем 51 печ. л. Тираж 2000 экз. Заказ N
Отпечатано с диапозитивов в Академической типографии «Наука» РАН
199034, Санкт-Петербург, 9 линия, 12.

Посвящается Алане. Ты начало новой жизни.

Оглавление

Предисловие

Первое издание книги «TCP/IP. Сетевое администрирование» было написано в 1992 году. За истекшие десять лет многое изменилось, но некоторые вещи остались все теми же. TCP/IP по-прежнему сохраняет свое лидерство среди протоколов связи, объединяющих разнотипные компьютерные системы. Он остается фундаментом для взаимодействия и обмена данными, для глобальных компьютерных сетей. Примечательно, что протоколы IP (Internet Protocol, протокол Интернета, или межсетевой протокол), TCP (Transmission Control Protocol, протокол управления передачей) и UDP (User Datagram Protocol, протокол пользовательских дейтаграмм), составляющие базу TCP/IP, не изменились. Изменились способы применения TCP/IP и управления этими протоколами.

Символичен для этих перемен тот факт, что дома у моей тещи есть подключение к сети TCP/IP, которое позволяет ей обмениваться электронной почтой, изображениями и гипертекстовыми документами с другими людьми своего поколения. Для нее это просто «выход в Интернет», но правда такова, что в ее домашней машине реализован полноценный стек протоколов TCP/IP, работает динамическое получение IP-адреса, а кроме того, используются типы данных, которые десять лет назад попросту не существовали.

В 1991 году протоколы TCP/IP были инструментом для опытных пользователей. Сетевые администраторы заведовали ограниченным числом систем и могли рассчитывать, что пользователи этих систем обладают определенным уровнем специальных знаний. Но это в прошлом. В 2002 году потребность в профессиональных сетевых администраторах выше, чем когда-либо ранее, поскольку контингент пользователей становится все более разношерстным и не столь подготовленным к самостоятельному решению технических проблем. В этой книге содержится информация для тех, кто хочет эффективно решать задачи сетевого администрирования TCP/IP.

«TCP/IP. Сетевое администрирование» стала первым сборником полезной информации для профессиональных сетевых администраторов TCP/IP и по сей день остается лучшей из подобных книг. За первым изданием последовал целый поток книг о TCP/IP и Интернете. Однако очень немногие из них сосредоточены на том, что действительно необходимо знать системному администратору об администрировании TCP/IP. Большинство книг – либо академические тексты, написанные с точки зрения архитектора протокола,

либо инструкции по использованию приложений TCP/IP. В них отсутствует практическая информация о сетях, которая необходима системным администраторам Unix. В настоящей книге сделан упор на TCP/IP и Unix, а также на поиск правильного соотношения между теорией и практикой.

Я горжусь предшествующими изданиями этой книги. Что же касается настоящего издания, я постарался сделать все возможное, чтобы не только сохранить настрой книги, но и улучшить ее. Рассмотрено динамическое назначение адресов при помощи протокола DHCP (Dynamic Host Configuration Protocol, протокол динамической настройки узлов). Материал, посвященный системе доменных имен (DNS), теперь охватывает BIND версии 8 и, в меньшей степени, BIND 9. Настройка электронной почты рассмотрена на примере текущей версии sendmail (8), а примеры, связанные с операционной системой, базируются на текущих версиях Solaris и Linux. Из протоколов маршрутизации описаны RIPv2 (Routing Information Protocol version 2, протокол маршрутной информации версии 2), OSPF (Open Shortest Path First, протокол предпочтения кратчайшего пути) и BGP (Border Gateway Protocol, протокол граничных шлюзов). Кроме того, добавлена глава, посвященная настройке веб-сервера Apache, новый материал по `xinetd`, а также информация о создании брандмауэров на базе `iptables`. Отмечу, что эти дополнительные темы не очень сильно увеличили объем книги.

TCP/IP – это набор протоколов связи, определяющих правила общения различных видов компьютеров между собой. «TCP/IP. Сетевое администрирование» – книга о том, как создать собственную сеть на базе TCP/IP. Эта книга является одновременно руководством, отвечающим на вопросы «как» и «почему» из области сетей TCP/IP, и справочником по отдельным сетевым приложениям.

Для кого эта книга

Эта книга предназначена всем владельцам Unix-машин, подключенных к сети TCP/IP.[1] Очевидно, в эту категорию попадают администраторы сетей и систем, отвечающие за настройку и сопровождение машин сети, но также и пользователи, которые желают узнать, каким образом их компьютеры общаются с другими системами. Провести границу между «системным администратором» и «конечным пользователем» довольно сложно. Человек может считать себя пользователем, но, работая на Unix-машине, ему, скорее всего, приходится заниматься и задачами системного администрирования.

В последние несколько лет, словно грибы после дождя, появляются книги для «чайников» и «идиотов». Эта книга не для людей, которые считают себя «идиотами» в отношении Unix. Кроме того, эта книга едва ли пригодится

[1] Большая часть текста применима не только к Unix-системам. Многие форматы файлов и команды, а также все описания протоколов справедливы для операционных систем Windows 98/NT/2000 и других. Администраторам NT-систем можно порекомендовать книгу «Windows NT TCP/IP Network Administration», O'Reilly.

«гениям» от сетевого администрирования. Однако читатели, не относящиеся к этим крайностям, найдут в книге немало полезной информации.

Предполагается, что читатели хорошо разбираются в работе компьютеров и знакомы с основами администрирования Unix-систем. Если это не так, изучить основы поможет книга Элин Фриш (Jeen Frisch) «Essential System Administration» (Основы системного администрирования), O'Reilly, серия Nutshell Handbook).

Структура книги

Книга состоит из трех логических частей: основные понятия, руководство, справочник. Три первых главы в общих чертах рассказывают о протоколах и службах TCP/IP. Они содержат основные понятия, необходимые для понимания последующих глав. Последующие главы содержат практические инструкции по различным темам. Главы с 4 по 7 посвящены планированию сети и настройке основных программных пакетов, необходимых для ее работы. Главы с 8 по 11 рассказывают о настройке основных сетевых служб. Главы 12 и 13 – о двух насущных вопросах, связанных с обеспечением надежной работы сети: безопасности и разрешении проблем. Завершает книгу ряд приложений-справочников, посвященных важным командам и программам.

Книга состоит из следующих глав:

Глава 1 «Обзор TCP/IP» содержит историю TCP/IP, описание архитектуры протокола, а также объясняет принципы его функционирования.

Глава 2 «Доставка данных» описывает адресацию и передачу данных по сети адресатам.

Глава 3 «Сетевые службы» посвящена отношениям между системами клиент–сервер, а также различным службам, жизненно важным для существования современной сети Интернет.

Глава 4 «Начинаем работу» открывает обсуждение установки и настройки сети, рассказывая о предварительном планировании, которое является необходимым шагом при создании сетей.

Глава 5 «Базовая настройка» посвящена вопросам настройки TCP/IP на уровне ядра Unix и настройки системы для запуска сетевых служб.

Глава 6 «Настройка интерфейса» расскажет о том, как связать сетевой интерфейс и сетевое программное обеспечение. Глава содержит примеры настройки интерфейсов Ethernet и PPP.

Глава 7 «Настройка маршрутизации» описывает, как организовать маршрутизацию, позволяющую машинам сети корректно взаимодействовать с другими сетями. В частности, освещены статические таблицы маршрутизации, распространенные протоколы маршрутизации, а также `gated` – пакет, реализующий последние версии некоторых из протоколов маршрутизации.

Глава 8 «Настройка DNS» посвящена администрированию программы сервера имен, который преобразует имена машин сети в адреса Интернета.

Глава 9 «Локальные службы сети» описывает настройку многих из распространенных сетевых серверов, в частности сервера настройки DHCP, сервера печати LPD, почтовых серверов POP и IMAP, сетевой файловой системы NFS (Network File System), серверов файлов и печати Samba, а также сетевой информационной службы NIS (Network Information System).

Глава 10 «sendmail» рассказывает о настройке sendmail – демона, отвечающего за доставку сообщений электронной почты.

Глава 11 «Настройка Apache» описывает настройку веб-сервера Apache.

В главе 12 «Сетевая безопасность» речь идет о том, как использовать современный Интернет, не подвергаясь излишнему риску. Глава посвящена угрозам безопасности, связанным с работой в сети, и возможным способам защиты от них.

Глава 13 «Разрешение проблем TCP/IP» рассказывает, какие действия можно предпринять, если что-то идет не так. Описаны методы и инструменты диагностирования проблем TCP/IP, приводятся примеры реальных проблем и их решений.

Приложение A «Инструментарий PPP» – это справочное руководство по различным программам, применяемым в настройке последовательных портов для работы по TCP/IP. Описаны программы `dip`, `pppd` и `chat`.

Приложение B «gated, справочник» – это справочное руководство по языку настройки пакета маршрутизации `gated`.

Приложение C «named, справочник» – это справочное руководство по серверу имен BIND (Berkeley Internet Name Domain).

Приложение D «dhcpd, справочник» – это справочное руководство по демону `dhcpd` (Dynamic Host Configuration Protocol Daemon).

Приложение E «sendmail, справочник» является справочным руководством по синтаксису, параметрам и ключам настройки sendmail.

В приложении F «Файл httpd.conf в Solaris» приводится содержимое файла настройки Apache, о котором идет речь в главе 11.

Приложение G «Выдержки из RFC» содержит информативные справочные фрагменты по протоколам из документов RFC, которые дополняют примеры диагностирования и разрешения проблем главы 13. Кроме того, в приложении содержатся сведения о том, где взять полноценные документы RFC.

Версии Unix

Большинство примеров книги относятся к Red Hat Linux, наиболее популярному в настоящее время дистрибутиву Linux, а также к Solaris 8, операционной системе от Sun, основанной на Unix System V. По счастью, программные средства TCP/IP достаточно стандартны в разных системах, что делает приво-

димые примеры универсальными – они должны работать в любых системах Linux, System V и BSD. Незначительные различия в выходных данных команд и параметрах командной строки не должны представлять затруднений.

Некоторые дополнительные сетевые приложения имеют собственные номера версий, не зависящие от версий операционных систем. Номера версий дополнительных пакетов упоминаются, когда это уместно. Наиболее важные из подобных пакетов:

BIND

Приводимое описание пакета BIND относится к версиям ветви 8, работающим в ОС Solaris 8. BIND 8 – это версия пакета BIND, поставляемая в составе Solaris и поддерживающая все стандартные типы записей ресурсов. В отношении базовых настроек версии BIND 8 и более новая BIND 9 не сильно различаются.

sendmail

Информация о пакете sendmail соответствует версии 8.11.3 и должна быть применима для всех других версий sendmail ветви 8.

Типографские соглашения

В книге использованы следующие типографские соглашения:

Курсив

Применяется для отображения имен файлов, каталогов, узлов, доменов, а также для выделения новых терминов.

`Моноширинный шрифт`

Применяется для отображения содержимого файлов и вывода команд, а также для выделения команд, параметров и ключевых слов в тексте.

`Моноширинный полужирный шрифт`

Применяется в примерах для выделения команд, набираемых пользователем.

`Моноширинный курсив`

Используется в примерах и в тексте для выделения переменных, значения которых должны быть подставлены в зависимости от обстоятельств. (Например, переменную *`filename`* необходимо заменять конкретным именем файла.)

`%, #`

Команды, вводимые в диалоговом режиме, отмечены приглашением стандартного интерпретатора команд C shell (`%`). Если команда должна выполняться с полномочиями администратора, она отмечается стандартным приглашением суперпользователя (`#`). В примерах, где участвует сразу несколько машин сети, приглашению может предшествовать имя той машины, на которой выполняется команда.

`[ ключ ]`

В описании синтаксиса команд необязательные аргументы заключаются в квадратные скобки. Так, запись `ls [ -l ]` означает, что ключ `-l` является необязательным.

Нам важно знать ваше мнение

Мы тщательно, насколько представляется возможным, проверили всю информацию в настоящей книге, но вы можете обнаружить, что возможности программ изменились (или наши ошибки!). Пожалуйста, сообщайте обо всех найденных ошибках, а также присылайте предложения, связанные с последующими изданиями книги по адресу:

O'Reilly & Associates, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (в США или Канаде)
(707) 829-0515 (международный/местный)
(707) 829-0104 (факс)

Издательством O'Reilly создана веб-страница, посвященная этой книге, на которой доступна информация о найденных ошибках и будут появляться разнообразные дополнительные сведения. Страница доступна по адресу:

http://www.oreilly.com/catalog/tcp3

Комментарии и технические вопросы, связанные с книгой, присылайте по адресу электронной почты:

bookquestions@oreilly.com

На веб-сайте издательства O'Reilly представлена дополнительная информация о книгах, конференциях, программном обеспечении, источниках информации и Сети O'Reilly (O'Reilly Network):

http://www.oreilly.com

Чтобы узнать, чем еще занимается Крэйг Хант, посетите его веб-сайт, *http://www.wrotethebook.com*.

Благодарности

Я хотел бы поблагодарить многих людей, которые помогли подготовить эту книгу. Те, кто участвовал в подготовке двух первых изданий, в первую очередь заслуживают благодарности; их вклад живет и в этом издании. Первое издание: Джон Уок (John Wack), Мэтт Бишоп (Matt Bishop), Вьетс Венема (Wietse Venema), Эрик Оллман (Eric Allman), Джефф Хониг (Jeff Honig),

Скотт Брим (Scott Brim) и Джон Дорган (John Dorgan). Второе издание: снова Эрик Оллман, Брайан Косталес (Bryan Costales), Крикет Ли (Cricket Liu), Пол Альбитц (Paul Albitz), Тед Лемон (Ted Lemon), Элизабет Цвики (Elizabeth Zwicky), Брент Чепмен (Brent Chapman), Симсон Гарфинкель (Simson Garfinkel), Джефф Седайо (Jeff Sedayao), а также Элин Фриш (Жleen Frisch).

В третьем издании книга стала лучше благодаря участию людей, многие из которых сами являются авторами. Они не только помогли мне с техническими вопросами – благодаря им я стал лучше писать. Три автора заслуживают отдельной благодарности. Многочисленные комментарии Крикета Ли, одного из авторов лучшей в мире книги о DNS, позволили улучшить раздел, посвященный системе доменных имен. Дэвид Колье-Браун (David Collier-Brown), один из авторов книги «Using Samba», написал исчерпывающую техническую рецензию на материал по Samba. Чарльз Олдс (Charles Aulds), автор бестселлера об администрировании сервера Apache, оказал помощь при написании главы о настройке Apache. Все эти люди помогли мне улучшить книгу в третьем издании. Спасибо!

Сотрудники издательства O'Reilly & Associates постоянно помогали мне. Деб Кэмерон (Deb Cameron), мой редактор, заслуживает отдельной благодарности. Ее усилий хватало и на развитие книги, и на общение со своей новорожденной красавицей, Вифанией Розой. Эмили Квилл (Emily Quill) играла роль выпускающего редактора и куратора проекта. Джефф Холкомб (Jeff Holcomb) и Джейн Эллин (Jane Ellin) выполняли проверку качества. За техническую помощь спасибо Леанне Сойлемез (Leanne Soylemez). Том Динз (Tom Dinse) создал указатель. Автором обложки является Эди Фридман (Edie Freedman), а стилевое оформление текста делала Мелани Вонг (Melanie Wang). Нил Уоллз (Neil Walls) занимался преобразованием текста из формата Microsoft Word в формат редактора Framemaker. Иллюстрации предшествующих изданий, созданные Крисом Райли (Chris Reilley) и Робертом Романо (Robert Romano), были обновлены стараниями Роберта Романо и Джессамина Рида (Jessamyn Read).

Наконец, я хочу поблагодарить мою семью – Кэти, Сару, Дэвида и Ребекку. Только их стараниями давление сроков сдачи материала до сих пор не свело меня с ума. Ребята, вы лучше всех.

1

Обзор TCP/IP

- *TCP/IP и Интернет*
- *Модель обмена данными*
- *Архитектура протоколов TCP/IP*
- *Уровень доступа к сети*
- *Уровень Internet*
- *Транспортный уровень*
- *Прикладной уровень*

Любой, кто пользуется настольной системой Unix – будь то инженер, преподаватель, ученый или деловой человек, – избрал в качестве второго занятия системное администрирование. Работа с сетями на подобных компьютерах ставит перед нами еще и задачи, связанные с сетевым администрированием.

Сетевое администрирование и системное администрирование не тождественны. Задачи системного администрирования – создание новых пользователей или резервных копий – ограничены одной независимой компьютерной системой. Дела обстоят совсем иначе в сетевом администрировании. Как только компьютер подключается к сети, он начинает взаимодействовать со многими другими системами. Качество выполнения задач сетевого администрирования оказывает определенное влияние не только на локальную систему, но и на многие системы сети, а следовательно, твердое понимание основ сетевого администрирования приносит пользу всем.

Объединение компьютеров в сети невероятно повышает их способность к общению, – а большинство компьютеров используются в основном для обмена данными, а не для вычислений. Вычислительными задачами для науки и бизнеса занимаются многочисленные суперкомпьютеры, но их число бледнеет в сравнении с миллионами систем, занятых под задачи вроде отправки почтовых сообщений или извлечения данных из удаленного хранилища. Более того, если оценить число настольных систем, используемых преимущественно для подготовки документов, позволяющих передавать мысли и идеи другим людям (речь идет о сотнях миллионов машин), станет понятно, почему компьютеры можно рассматривать как устройства для обмена информацией.

Положительное влияние компьютерных коммуникаций приводит к росту числа и видов компьютеров, участвующих в работе сетей. Одним из главных преимуществ TCP/IP является возможность прозрачного сообщения всех видов аппаратного обеспечения и операционных систем.

Название «TCP/IP» связано с целым семейством протоколов передачи данных. Оно происходит от названий двух протоколов этого семейства: протокола управления передачей (TCP, Transmission Control Protocol) и протокола Internet (IP, Internet Protocol). TCP/IP – это традиционное имя семейства протоколов, которое и используется в книге. Другое название – семейство протоколов Internet (IPS, Internet Protocol Suite) – также является приемлемым.

Настоящая книга содержит практические пошаговые инструкции по настройке и сопровождению сетевых приложений TCP/IP для компьютерных Unix-систем. TCP/IP является ведущей технологией создания локальных и корпоративных сетей, а также фундаментом всемирной сети Интернет. TCP/IP – самый важный программный комплекс для сетевого администратора Unix.

Первая часть книги рассказывает об основах TCP/IP и принципах передачи данных по сети. Вторая часть посвящена настройке и применению TCP/IP в системе Unix. Начнем с краткого исторического экскурса.

TCP/IP и Интернет

В 1969 году управление передовых исследований (Advanced Research Projects Agency, ARPA) финансировало исследования и разработку в рамках проекта по созданию экспериментальной сети на базе коммутации пакетов. Эта сеть, получившая имя *ARPAnet*, создавалась с целью изучения методов обеспечения устойчивой, надежной, не зависящей от оборудования передачи данных. Многие из применяемых сегодня методов передачи данных родились в недрах ARPAnet.

Экспериментальная сеть имела такой успех, что многие из организаций-участниц проекта начали использовать ее возможности на постоянной основе. В 1975 году ARPAnet из экспериментальной сети превратилась в рабочую, а административные полномочия были переданы управлению оборонных коммуникаций (Defense Communications Agency, DCA).[1] Однако развитие ARPAnet не прекратилось после смены статуса: базовые протоколы TCP/IP были разработаны несколько позже.

Протоколы TCP/IP были приняты в качестве военных стандартов (Military Standards, MIL STD) в 1983 году, и в этот момент всем подключенным к сети узлам предписывалось перейти на новые протоколы. Для того чтобы облегчить переход, управление DARPA[2] финансировало Болта, Беранека и Ньюмана (Bolt, Beranek, Newman; BBN), которые реализовали TCP/IP для системы Berkeley (BSD) Unix. Так начался союз систем Unix и TCP/IP.

---

[1] Управление DCA в настоящее время носит название управления оборонных информационных систем (Defense Information Systems Agency, DISA).

[2] В 80-е годы управление ARPA, входящее в состав Министерства обороны США, получило новое название: Defense Advanced Research Projects Agency (DARPA). Вне зависимости от названия, перед управлением во все времена стояла одна задача – финансирование передовых исследований.

Термин *Internet* вошел в употребление примерно в то же время, когда были приняты стандарты по TCP/IP. В 1983 году сеть ARPAnet была поделена на MILNET, рассекреченную часть оборонной информационной сети (Defense Data Network, DDN), и новую сеть ARPAnet, размерами поменьше предыдущей. Термин «Internet» использовался для обозначения сети в целом: MILNET плюс ARPAnet.

В 1985 году национальным научным фондом (National Science Foundation, NSF) была создана сеть NSFNet, которая подключилась к существовавшей в то время сети Internet. Изначально сеть NSFNet объединяла пять суперкомпьютерных центров NSF. Она уступала ARPAnet размерами, равно как и скоростью (56 Кбит/с). И все же создание NSFNet стало значительным событием в истории сети Internet, поскольку фонд NSF создал не только сеть, но и новое видение того, как можно использовать Internet. Идея NSF заключалась в том, чтобы каждый ученый и каждый инженер в США получил доступ к сети. С этой целью в 1987 году фонд NSF создал новую, трехзвенную топологию сети, которая объединяла магистральные, региональные и локальные сети. В 1990 году сеть ARPAnet перестала существовать формально, в 1995 году сеть NSFNet утратила свою роль первичной магистральной сети Internet.

Сегодня сеть Интернет имеет невообразимые размеры и состоит из сотен тысяч сетей, разбросанных по всему миру. Глобальная сеть больше не зависит от базовой (или магистральной) сети или от работы правительственных структур. Современный Интернет – это творение поставщиков услуг, работающих на коммерческой основе. Национальные поставщики сетевых услуг, то есть поставщики первого звена, а также региональные поставщики услуг создают собственно инфраструктуру. Поставщики услуг Интернета (Internet Service Providers, ISPs) обеспечивают локальный доступ и обслуживание пользователей. Сеть сетей сходится воедино на территории США, в точках доступа к сети (Network Access Points, NAPs).

Сеть Интернет вышла далеко за изначальные рамки. Сети и управления, благодаря которым создавался Интернет, утратили свою решающую роль в жизни этой сети. Интернет эволюционировал из простой магистральной сети, через трехзвенную иерархическую структуру, в гигантскую сеть взаимосвязанных, распределенных сетевых концентраторов. Начиная с 1983 года, сеть растет экспоненциально, удваиваясь в размерах ежегодно. И даже эти невероятные изменения никак не повлияли на один простой факт: основой сети Интернет является семейство протоколов TCP/IP.

Одним из признаков успеха сети является путаница, связанная с термином *internet*. Исходное его значение – сеть, построенная на протоколе IP. Сегодня *internet* – это общий термин, обозначающий целый класс сетей. Сеть internet (со строчной буквы «i») – это произвольный набор физически обособленных сетей, связанных общим протоколом с целью формирования объединяющей логической сети. Internet (с прописной буквы «I») – глобальный набор взаимосвязанных сетей, который берет начало в сети ARPAnet и объединяет физические сети в одну логическую при помощи протокола IP.

В этой книге оба термина («internet» и «Internet») относятся к сетям, построенным на основе TCP/IP.

Поскольку наличие TCP/IP является обязательным условием подключения к Интернету, рост этой сети приводит к росту интереса к TCP/IP. Распространение TCP/IP в организациях привело к осознанию того факта, что потенциал протоколов может быть использован в любых сетевых приложениях. Протоколы Интернета часто используются для построения локальных сетей, пусть даже не подключенных к Интернету. Кроме того, TCP/IP широко применяется в построении корпоративных сетей. Корпоративные сети на основе TCP/IP, в которых для распространения корпоративной информации применяются методологии Интернета и веб-инструменты, носят название *интрасетей* (*intranets*). TCP/IP является основанием для всех систем подобного рода.

Особенности TCP/IP

Популярность протоколов TCP/IP росла быстро не просто потому, что протоколы существовали, а подключение к сети Интернет требовало их применения. В нужный момент они позволили удовлетворить важную потребность (глобальную передачу данных), а кроме того, обладали рядом особенностей, необходимых для решения задачи:

- Свободно распространяемые открытые стандарты протоколов, не зависящих от конкретного аппаратного обеспечения или операционной системы. Широкая поддержка делает TCP/IP идеальным выбором для объединения разнообразных программных и аппаратных составляющих даже в случаях, когда нет необходимости работать с сетью Интернет.
- Независимость от конкретных физических сетевых устройств, позволяющая интегрировать самые разные типы сетей посредством TCP/IP. TCP/IP может работать через Ethernet, соединение DSL, коммутируемое соединение, оптоволоконный канал, то есть в качестве физического транспорта может использоваться почти любая система.
- Универсальная схема адресации, позволяющая произвольному устройству TCP/IP обращаться к любому другому устройству сети по уникальному адресу, даже во всемирной сети Интернет.
- Стандартизированные высокоуровневые протоколы, согласующие работу распространенных пользовательских служб.

Стандарты протоколов

Протокол – это свод официальных правил поведения. В международных отношениях протоколы призваны воспрепятствовать возникновению проблем, вызванных культурными различиями стран. Оговаривая ряд общих правил, которые широко известны и не привязаны к обычаям конкретной нации, дипломатические протоколы минимизируют возможные недоразумения: всем известно, как следует себя вести и как интерпретировать пове-

дение других сторон. Точно так же, чтобы компьютеры смогли общаться, необходимо определить набор правил, которым подчиняется это общение.

В обмене данными подобные наборы правил также называют *протоколами*. В гомогенных сетях правила передачи данных определяются единственным поставщиком компьютеров и направлены на эффективное использование операционной системы и аппаратной архитектуры. Гомогенная сеть в нашей аналогии – это культура отдельной страны, которую могут назвать домом лишь ее граждане. TCP/IP создает гетерогенную сеть с открытыми протоколами, не зависящими от деталей архитектуры компьютера и реализации операционной системы. Протоколы TCP/IP доступны всем, они развиваются и изменяются по единодушному решению, а не по распоряжению конкретного производителя. Кто угодно может создавать продукцию, соответствующую спецификациям этих открытых протоколов.

Открытая природа протоколов TCP/IP требует открытости процесса разработки стандарта и свободного доступа к соответствующим документам. Разработка стандартов Интернета происходит на открытых сессиях комитета по технологической поддержке сети Интернет (Internet Engineering Task Force, IETF). Разработанные таким образом протоколы публикуются в виде документов RFC (*Request for Comments*, запрос комментариев).[1] Как и следует из названия, стилистика и содержание этих документов гораздо менее строги, чем в большинстве стандартов. Документы RFC содержат широкий спектр интересных и полезных сведений и не ограничиваются формальными спецификациями протоколов обмена данными. Существует три основных типа документов RFC: стандарты (standards, STD), современные практики (best current practices, BCP), а также уведомляющие (for your information, FYI).

Документы RFC, определяющие официальные стандарты протоколов, обозначаются аббревиатурой STD и получают STD-номера в дополнение к RFC-номерам. Создание официального стандарта Интернета – строго последовательный процесс. Стандарты RFC становятся таковыми, лишь пройдя через *три уровня зрелости*:

Заявка стандарта (Proposed Standard)

Спецификация протокола, который является достаточно важным и уже получил достаточно широкую поддержку интернет-сообщества, чтобы предлагаться в качестве стандарта. Такая спецификация прозрачна и закончена, но не является стандартом, и, более того, может никогда не достигнуть статуса стандарта.

Проект стандарта (Draft Standard)

Спецификация протокола, для которой существует по меньшей мере две независимых, взаимозаменяемых реализации. Проект стандарта – это окончательная спецификация, используемая в широком тестировании.

[1] Желаете узнать, как создаются стандарты Интернета? Прочтите документ RFC 2026, *The Internet Standards Process* (Процесс разработки стандартов Интернета).

Проект изменяется лишь в том случае, когда этого требуют результаты тестирования.

Стандарт Интернета (Internet Standard)

Спецификация получает статус стандарта лишь после всеобъемлющего тестирования и лишь в том случае, когда применение определенного этой спецификацией протокола может принести значительную выгоду интернет-сообществу.

Стандарты делятся на две категории. *Техническая спецификация* (*Technical Specification*, TS) дает определение протокола. *Формулировка применимости* (*Applicability Statement*, AS) описывает случаи, когда протокол следует применять. Применимость стандарта имеет три возможных уровня:

Обязательный (Required)

Стандартный протокол, является обязательной частью любой реализации TCP/IP. Стек протоколов должен включать этот протокол, чтобы соответствовать стандарту.

Рекомендованный (Recommended)

Стандартный протокол, рекомендованный к включению во все реализации TCP/IP. Его присутствие не является обязательным условием.

Факультативный (Elective)

Факультативный стандарт. Решение по реализации принимает разработчик конкретного пакета приложений.

Два других уровня применимости (*ограниченного применения* и *не рекомендован*) связаны с документами RFC, существующими обособленно от процесса стандартизации. Протокол «ограниченного применения» используется только в особых случаях, скажем, в ходе экспериментов. Протоколы, «не рекомендованные» к применению, являются устаревшими либо имеют ограниченную функциональность. Документы RFC, *не принадлежащие процессу стандартизации*, бывают трех типов:

Экспериментальные (Experimental)

Применение экспериментальных документов RFC ограничено исследованиями и разработкой.

Исторические (Historic)

Исторические RFC являются устаревшими, их применение не рекомендуется.

Уведомляющие (Informational)

Уведомляющие документы RFC содержат информацию, представляющую интерес для широких слоев сообщества сети Интернет, но не содержат определений протоколов.

Подмножество уведомляющих RFC носит имя уведомлений FYI (For Your Information, примите к сведению). Документу FYI, помимо номера RFC, присваивается еще и номер FYI. Эти документы содержат вводный и подготови-

тельный материал по сети Интернет и сетям TCP/IP в целом. Документы FYI не упомянуты в RFC 2026 и не являются составляющей процесса стандартизации. Тем не менее некоторые документы FYI представляют интерес.[1]

Вторая группа документов RFC, лежащих за пределами задачи документирования протоколов, носит имя BCP (Best Current Practices, лучшие современные практики). Задачей документов BCP является формальное документирование методов и процедур. Некоторые из них описывают принципы, которым подчиняются действия самой организации IETF; примером такого BCP-документа может послужить RFC 2026. Прочие содержат рекомендации по функционированию сетей или служб (для примера возьмем RFC 1918, *Address Allocation for Private Internets,* выделение адресов в закрытых интернет-сетях). Документы BCP последнего типа зачастую представляют особый интерес для сетевых администраторов.

В настоящее время существует более трех тысяч документов RFC. Так или иначе, администратору систем, объединенных в сеть, придется прочесть некоторые из них. Важно не только уметь понять эти документы в процессе чтения, но и знать, какие именно следует читать. Чтобы определить, как RFC соответствуют случаю, воспользуйтесь категориями и уровнями применимости. (Для начала неплохо сосредоточить внимание на тех документах RFC, которым присвоен номер STD.) Для понимания прочитанного необходимо понимание языка, на котором происходит обмен данными. Документы RFC определяют спецификации для реализаций протоколов в терминах языка, который является уникальным для обмена данными.

Модель обмена данными

В разговоре о компьютерных сетях приходится пользоваться терминами, имеющими особые значения. Даже компьютерные специалисты из других областей знакомы далеко не с каждым термином сетевой алфавитной каши. Привычное замечание – английский язык и компьютерный жаргон не являются эквивалентными (и даже совместимыми) языками. И хотя описания и примеры должны прояснить значение терминов сетевого жаргона, некоторые из слов время от времени оказываются двусмысленными. Таким образом, для понимания терминологии обмена данными необходима общая точка отсчета.

Для описания структуры и функциональности протоколов обмена данными часто используется архитектурная модель, разработанная Международной организацией по стандартизации (International Standards Organization, ISO). Данная модель как раз является общей точкой отсчета для разговора о коммуникациях и называется *опорной моделью взаимодействия открытых систем* (*Open Systems Interconnect* (OSI) *Reference Model*). Термины, определенные в рамках модели, нашли глубокое понимание и имеют широкое хож-

[1] Более подробно о документах FYI можно узнать из RFC 1150, *FYI on FYI: An Introduction to the FYI Notes* (Введение в уведомления FYI).

дение в сетевом сообществе – настолько широкое, что обмен данными сложно обсуждать, не применяя терминологию OSI.

Опорная модель OSI состоит из семи уровней, определяющих функциональность протоколов обмена данными. Каждый из уровней модели OSI представляет функцию, выполняемую при передаче данных между приложениями, взаимодействующими по сети. Названия и краткие функциональные описания уровней приведены на рис. 1.1, где протоколы похожи на столбик из кирпичей. По этой причине структура, о которой идет речь, часто именуется *стеком*, или *стеком протоколов*.

***Рис. 1.1.** Опорная модель OSI*

Уровень не содержит определения отдельного протокола; определяемая уровнем функциональность может быть реализована любым числом протоколов. Таким образом, каждый уровень способен вмещать произвольное число протоколов, каждый из которых реализует службу, соответствующую функциональности этого уровня. Так, протоколы передачи файлов и электронной почты реализуют пользовательские службы, и оба принадлежат к прикладному уровню.

Каждый протокол реализует взаимодействие с протоколами равного положения. *Протокол равного положения* – это реализация того же протокола на эквивалентном уровне удаленной системы; так, локальный протокол пе-

редачи файлов является протоколом равного положения с удаленным протоколом передачи файлов. Успешное взаимодействие протоколов одного уровня должно следовать установленным стандартам. Теоретически работа каждого протокола направлена лишь на взаимодействие с протоколами равного уровня, и ему нет дела до соседних уровней.

Но есть и другой момент, нуждающийся в согласовании: передача данных между уровнями в пределах отдельного компьютера. Дело в том, что в передаче данных от локального приложения эквивалентному удаленному приложению участвуют все уровни. Вышележащие уровни в вопросах передачи данных по сети полагаются на нижележащие. Данные передаются вниз по стеку, с одного уровня на другой, и, в конце концов, передаются по сети протоколами физического уровня. На стороне «собеседника» данные передаются вверх по стеку приложению-адресату. Отдельный уровень может не знать, каким образом работают соседи; необходимым знанием является лишь способ передачи и получения данных. Изоляция функций сетевого взаимодействия на различных уровнях сводит к минимуму воздействие технологических изменений на все семейство протоколов. Создание новых приложений не требует внесения изменений в физическую базу сети, а установка нового сетевого оборудования не требует модификации пользовательских программ.

Несмотря на эффективность модели OSI протоколы TCP/IP не до конца следуют установленной структуре модели. Поэтому в разговоре о TCP/IP мы будем интерпретировать уровни модели OSI следующим образом:

Прикладной уровень (Application Layer)

Прикладной уровень иерархии протоколов содержит сетевые процессы, с которыми работают пользователи. В тексте книги приложение TCP/IP – это любой сетевой процесс, протекающий выше транспортного уровня. Под это определение подпадают все процессы, с которыми напрямую взаимодействуют пользователи, а также прочие процессы данного уровня, о которых пользователи могут и не знать.

Уровень представления (Presentation Layer)

Чтобы взаимодействующие приложения смогли обмениваться данными, необходимо соглашение о представлении данных. В OSI стандартная функциональность представления данных обозначена уровнем представления. Функция представления часто реализуется в приложениях TCP/IP, а также такими протоколами TCP/IP, как XDR и MIME.

Сеансовый уровень (Session Layer)

Аналогично уровню представления, сеансовый уровень не является самостоятельным в иерархии протоколов TCP/IP. Сеансовый уровень OSI отвечает за управление сеансами (соединениями) взаимодействия приложений. В TCP/IP эта функциональность заложена преимущественно на транспортном уровне, а термин «сеанс» не имеет хождения; для описания вектора взаимодействия приложений применяются термины «сокет» (socket, гнездо) и «порт».

Транспортный уровень (Transport Layer)

Большая часть информации в рассказе о TCP/IP связана с протоколами транспортного уровня. В опорной модели OSI транспортный уровень гарантирует получение адресатом данных в неизмененном виде. В TCP/IP такая функциональность возложена на *протокол управления передачей* (*Transmission Control Protocol*, TCP). При этом в TCP/IP существует вторая служба транспортного уровня – *протокол пользовательских дейтаграмм* (*User Datagram Protocol*, UDP), который не гарантирует надежной доставки данных.

Сетевой уровень (Network Layer)

Сетевой уровень управляет сетевыми соединениями и проводит границу между протоколами более высокого уровня и подробностями реализации собственно сети. Сетевой уровень TCP/IP обычно подразумевает протокол Internet (IP), который разграничивает вышележащие уровни и сеть, а также отвечает за адресацию и доставку данных.

Канальный уровень (Data Link Layer)

Надежная доставка данных по физической сети находится в ведении канального уровня. Этот уровень TCP/IP, как правило, не содержит протоколов. В большинстве документов RFC, упоминающих канальный уровень, рассматриваются вопросы интеграции IP и существующих канальных протоколов.

Физический уровень (Physical Layer)

Физический уровень определяет характеристики аппаратного обеспечения, необходимого для осуществления передачи данных, в частности такие свойства, как уровни напряжения, количество и расположение контактов интерфейсов. В качестве примеров стандартов физического уровня можно упомянуть стандарты на интерфейсные разъемы RS232C V.35, а также монтажные стандарты для локальных сетей, такие как IEEE 802.3. TCP/IP не определяет физические стандарты, но пользуется существующими.

Терминология опорной модели OSI способствует более прозрачному описанию TCP/IP, но для полного понимания системы следует воспользоваться архитектурной моделью, точнее отражающей структуру TCP/IP. Модель, которую мы используем для описания TCP/IP, представлена в следующем разделе.

Архитектура протоколов TCP/IP

Несмотря на отсутствие универсальных правил описания TCP/IP посредством многоуровневой модели, модели TCP/IP обычно содержат менее семи уровней. Большинство описаний TCP/IP определяют от трех до пяти функциональных уровней архитектуры протокола. Четырехуровневая модель, приведенная на рис. 1.2, состоит из трех уровней (прикладной, узел-узел, доступ к сети) модели DOD Protocol Model из первого тома руководства по протоколам DDN и дополнительного уровня Internet. Такая модель обеспечивает приемлемую иллюстрацию уровней иерархии протоколов TCP/IP.

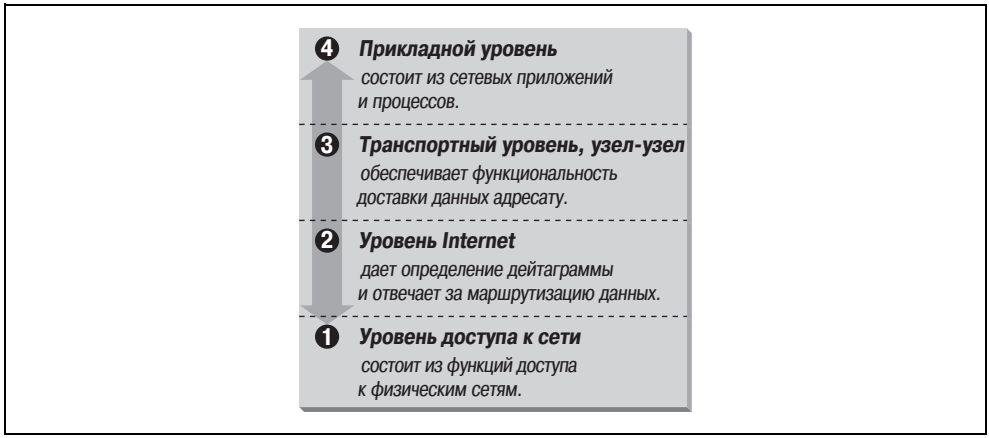

Рис. 1.2. *Архитектура TCP/IP*

Как и в модели OSI, данные передаются вниз по стеку при отправке в сеть и вверх по стеку – при получении из сети. Четырехуровневая структура TCP/IP проявляется в способе обработки данных при их прохождении вниз по стеку, от прикладного уровня непосредственно к физической сети. Каждый уровень стека добавляет управляющую информацию, гарантируя корректную доставку. Блок управляющей информации называется *заголовком* (*header*), поскольку предшествует передаваемым данным. Каждый уровень интерпретирует всю информацию, полученную от вышележащего уровня, в качестве данных и добавляет к этим данным собственный заголовок. Дополнение информации по доставке на каждом уровне носит название *инкапсуляции* (рис. 1.3).

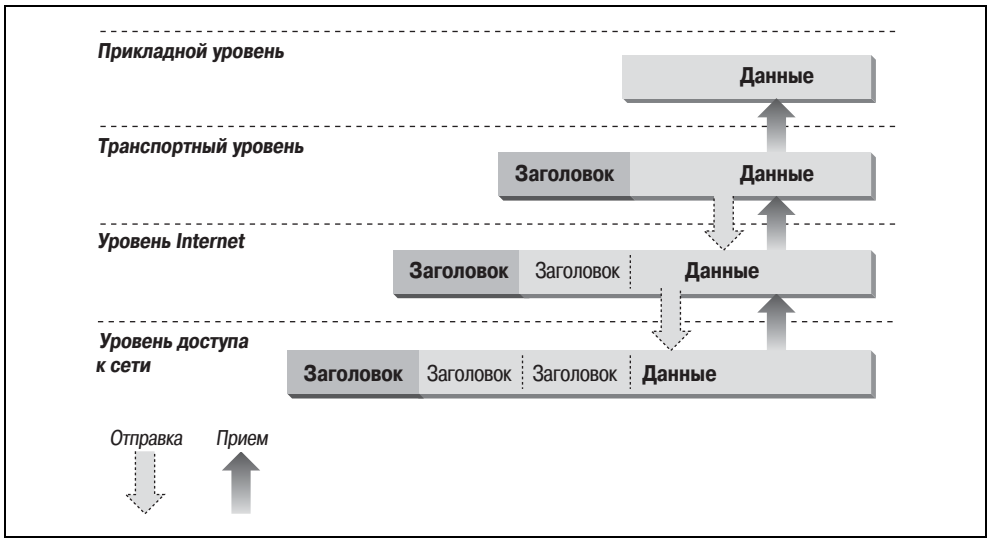

Рис. 1.3. *Инкапсуляция данных*

При получении данных происходит обратный процесс. Каждый уровень удаляет соответствующий заголовок и передает данные вышележащему уровню. При передаче вверх по стеку информация, получаемая от нижележащих уровней, интерпретируется в качестве заголовка и сопутствующих данных.

Каждому уровню соответствуют определенные структуры данных. Теоретически уровень не обязан знать о структурах данных, применяемых на соседних уровнях, однако на практике структуры данных уровня проектируются таким образом, чтобы хорошо сочетаться со структурами «соседей» в целях повышения эффективности передачи данных. Тем не менее каждому уровню соответствует собственная структура данных и специальная терминология ее описания.

На рис. 1.4 отражены термины, применяемые на различных уровнях TCP/IP в отношении передаваемых данных. Приложения TCP считают данные *потоком* (*stream*), а приложения UDP – *сообщением* (*message*). На транспортном уровне TCP данные хранятся в *сегментах* (*segment*), на транспортном уровне UDP – в *пакетах* (*packet*). Уровень Internet рассматривает данные в качестве блоков, называемых *дейтаграммами* (*datagrams*). Многочисленные типы сетей, поверх которых работает TCP/IP, также используют разнообразную терминологию в области передаваемых данных. В большинстве сетей приняты термины *пакет* (*packet*) или *фрейм* (*frame*, блок данных). На рис. 1.4 показана сеть, передающая фрагменты данных, называемые *фреймами*.

Рассмотрим более подробно функциональность каждого уровня, поднимаясь от уровня доступа к сети в направлении прикладного уровня.

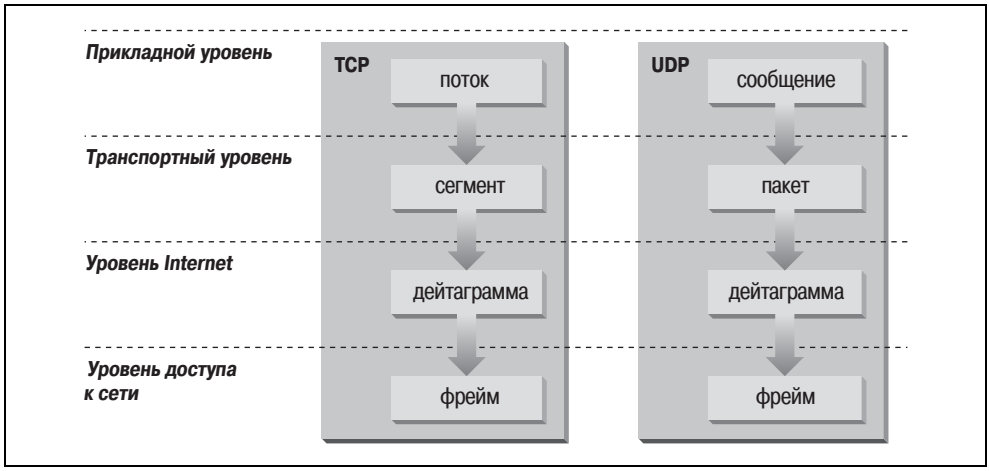

Рис. 1.4. *Структуры данных*

Уровень доступа к сети

Уровень доступа к сети является низшим уровнем иерархии протоколов TCP/IP. Протоколы этого уровня предоставляют системе средства доставки данных другим устройствам сети с прямым подключением. Этот уровень определяет способы использования сети для передачи IP-дейтаграмм. В отличие от протоколов более высоких уровней, протоколы доступа к сети должны обладать подробными сведениями о работе сети (структуре пакетов, системе адресации и т. д.), чтобы корректно форматировать передаваемые данные, следуя ограничениям, налагаемым сетью. Уровень доступа к сети TCP/IP может включать всю функциональность трех нижних уровней опорной модели OSI (сетевого, канального и физического).

Уровень доступа к сети часто остается незамеченным пользователями. Архитектура TCP/IP скрывает функциональность нижележащих уровней, а все более широко известные протоколы (IP, TCP, UDP и прочие) являются высокоуровневыми. Появление новых аппаратных возможностей сетевых устройств диктует необходимость разработки новых протоколов уровня доступа к сети. Такая разработка позволяет существующим TCP/IP-сетям использовать новое оборудование. Следствием такого положения является существование многочисленных протоколов доступа – по одному на каждый стандарт физической сети.

Функциональность данного уровня включает инкапсуляцию IP-дейтаграмм в передаваемые по сети фреймы, а также отображение IP-адресов в физические адреса, применяемые в сети. Одним из достоинств TCP/IP является универсальная схема адресации. Адрес IP должен подвергнуться преобразованию в адрес, уместный в физической сети, служащей для передачи дейтаграммы.

Протоколы уровня доступа к сети определяются следующими документами RFC:

- RFC 826, *Address Resolution Protocol* (ARP, протокол разрешения адресов); протокол выполняет отображение IP-адресов в адреса Ethernet.
- RFC 894, *A Standard for the Transmission of IP Datagrams over Ethernet Networks* (стандарт передачи IP-дейтаграмм в сетях Ethernet); документ определяет, каким образом производится инкапсуляция IP-дейтаграмм с целью передачи в сетях Ethernet.

Реализация протоколов этого уровня в системах Unix обычно представлена сочетанием драйверов устройств и сопутствующих программ. Модули, обозначаемые именами сетевых устройств, отвечают за инкапсуляцию и доставку данных по сети, а сопутствующие функции, вроде отображения адресов, возлагаются на самостоятельные программы.

Уровень Internet

Следующим в иерархии является *уровень Internet*. Наиболее важный протокол этого уровня – протокол Internet (IP). В существующей сети Интернет

применяется протокол IP версии 4 (IPv4), определенный в RFC 791. Существуют и более современные версии IP. IP версии 5 – это экспериментальный протокол потокового транспорта (Stream Transport, ST), применяемый для доставки данных в системах реального времени. IPv5 не получил рабочего распространения. IPv6 – стандарт IP, предоставляющий значительным образом расширенные диапазоны адресации. В IPv6 применяется совершенно иная структура адреса, так что протоколы IPv6 и IPv4 не способны к взаимодействию. Несмотря на существование стандарта IPv6 эта версия IP пока не получила широкого распространения в действующих коммерческих сетях. Нашей целью является изучение работы действующих сетей, поэтому мы не станем вдаваться в подробности IPv6. В тексте главы и большей части книги аббревиатура «IP» относится к IPv4. Именно протокол IPv4 является предметом настройки при необходимости организовать обмен данными между системами, и именно этому протоколу будет уделено основное внимание.

Протокол Internet – это сердце TCP/IP. Он обеспечивает работу базовой службы доставки пакетов, на которой построены сети TCP/IP. Все протоколы этого и соседствующих уровней используют протокол Internet для доставки данных. Все входящие и исходящие потоки данных TCP/IP проходят через IP независимо от пункта назначения.

Протокол Internet

Протокол Internet (IP) – это строительный элемент сетей Интернет. Он имеет следующую функциональность:

- Определяет дейтаграмму, базовую единицу передачи в сетях Интернет
- Определяет схему интернет-адресации
- Осуществляет обмен данными между уровнем доступа к сети и транспортным уровнем
- Выполняет маршрутизацию дейтаграмм, адресованных удаленным узлам
- Отвечает за разбиение и сборку дейтаграмм

Прежде чем перейти к более подробному рассмотрению этих функций, взглянем на некоторые свойства IP. Во-первых, протокол IP работает *без создания логических соединений*. Это означает, что передача данных не требует обмена управляющей информацией (подтверждения связи, известного в качестве «рукопожатия») с целью установки сквозного соединения. Напротив, протоколы, работающие *на основе соединений*, обмениваются управляющей информацией с удаленными системами, проверяя их готовность принимать данные. Успешное подтверждение связи является признаком того, что соединение установлено. Протокол Internet делегирует установку соединений протоколам других уровней для случаев, когда требуется наличие соединений.

Кроме того, в вопросах обнаружения ошибок и восстановления после ошибок протокол IP полагается на протоколы других уровней. Протокол Internet иногда называют *ненадежным протоколом*, поскольку он не реализует обнаружение и восстановление после ошибок. Однако это не означает, что на

протокол IP нельзя положиться – как раз наоборот. Можно положиться на IP в плане точной доставки данных в доступную сеть, но невозможно с его помощью проверить, что данные были корректно получены. Подобные проверки при необходимости реализуются посредством протоколов других уровней архитектуры TCP/IP.

Дейтаграмма

Протоколы TCP/IP создавались для организации передачи данных в ARPAnet, сети на базе *коммутации пакетов*. *Пакет* – это блок данных, содержащий информацию, необходимую для доставки. В этом отношении пакет похож на письмо, адрес получателя отражен на конверте. Сеть с пакетной коммутацией использует адресную информацию пакетов для коммутации пакетов из одной физической сети в другую, перемещая их ближе к пункту назначения. Пакеты путешествуют по сети независимо друг от друга.

Формат пакета, определяемый протоколом Internet, называется *дейтаграммой*. Содержимое IP-дейтаграммы наглядно показано на рис. 1.5. Первые пять или шесть 32-битных слов дейтаграммы содержат управляющую информацию, составляющую *заголовок* (*header*). По умолчанию заголовок имеет длину в пять слов, шестое является необязательным. Для указания переменной длины заголовка (в словах) используется поле *Internet Header Length* (IHL, *длина заголовка Internet*). Заголовок содержит всю необходимую для доставки пакета информацию.

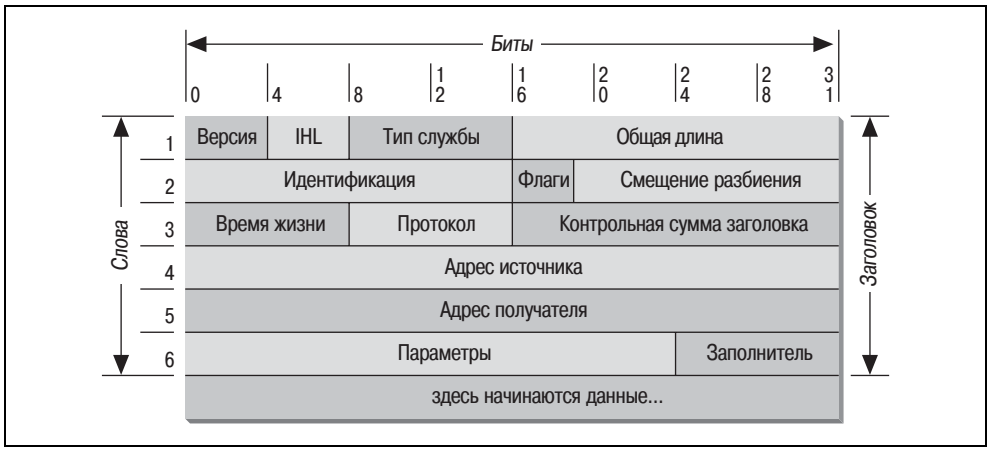

Рис. 1.5. *Формат дейтаграмм IP*

Протокол Internet выполняет доставку дейтаграммы на основе *адреса получателя* из пятого слова заголовка. Адрес получателя – это стандартный 32-битный адрес IP, соответствующий определенной сети и конкретному узлу этой сети. (Формат IP-адресов описан в главе 2.) Если адресом получателя является адрес узла локальной сети, пакет доставляется напрямую в пункт назначения. В противном случае пакет передается на шлюз (gateway) для до-

ставки. *Шлюзы* занимаются коммутацией пакетов между физически обособленными сетями. Принятие решения о том, какой именно шлюз использовать, называется *маршрутизацией*. Протокол IP принимает такое решение для каждого пакета.

Маршрутизация дейтаграмм

Шлюзы Internet применяют протокол Internet для маршрутизации пакетов по сетям, а потому их можно с большой степенью точности называть *IP-маршрутизаторами*. Традиционный жаргон TCP/IP включает лишь два типа сетевых устройств – *шлюзы* и *узлы*. Шлюзы, в отличие от узлов, передают пакеты между сетями. Однако если узел входит в несколько сетей (*многосетевой* узел), он получает возможность передавать пакеты из одной сети в другую. При пересылке пакетов многосетевой узел действует точно так же, как любой шлюз, и, вообще говоря, получает статус шлюза. Существующая сегодня терминология передачи данных различает шлюзы и маршрутизаторы[1], но мы будем считать термины *шлюз* и *IP-маршрутизатор* взаимозаменяемыми.

На рис. 1.6 отражена пересылка пакетов посредством шлюзов. Узлы (*оконечные системы*) производят обработку пакетов на всех четырех уровнях протоколов, в то время как шлюзы (*промежуточные системы*) обрабатывают пакеты лишь до уровня Internet, на котором происходит принятие решений по маршрутизации.

Система способна доставить пакет только на другое устройство, принадлежащее той же физической сети. Пакеты, адресованные узлом *A1* узлу *C1*, пересылаются через шлюзы *G1* и *G2*. Узел *A1* доставляет пакет шлюзу *G1*, с кото-

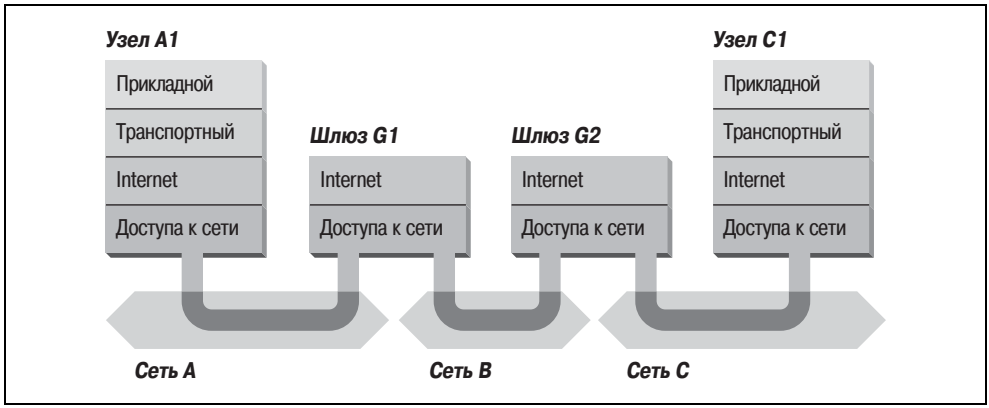

Рис. 1.6. *Маршрутизация посредством шлюзов*

[1] Согласно существующей терминологии, шлюз переносит данные, связывая различные протоколы, тогда как маршрутизатор переносит данные между различными сетями. Таким образом, система, передающая почту между сетями TCP/IP и X.400, является шлюзом, но традиционный IP-шлюз является маршрутизатором.

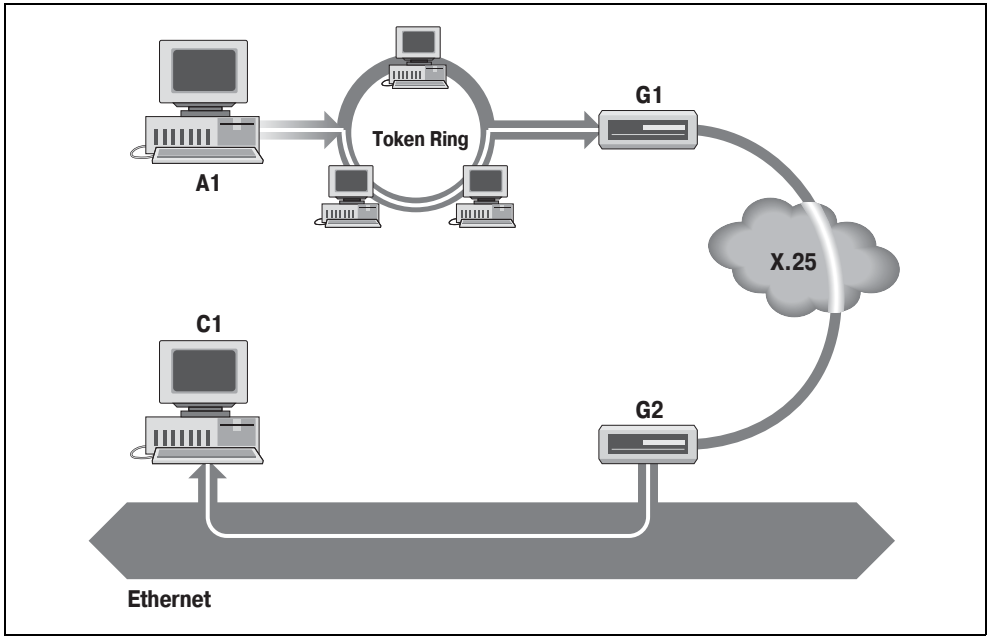

Рис. 1.7. *Сети, шлюзы и узлы*

рым вместе входит в сеть *A*. Шлюз *G1* доставляет пакет шлюзу *G2* по сети *B*. Шлюз *G2*, в свою очередь, доставляет пакет напрямую узлу *C1*, с которым вместе входит в сеть *C*. Узел *A1* понятия не имеет о существовании иных шлюзов, кроме *G1*. Он посылает пакеты, адресованные в сети *C* и *B*, этому локальному шлюзу и делегирует ему задачу корректной пересылки пакетов по нужным маршрутам. Точно так же узел *C1* отправляет свои пакеты шлюзу *G2*, чтобы обратиться к узлу сети *A* либо к узлу сети *B*.

Другой взгляд на маршрутизацию приведен на рис. 1.7. Данная иллюстрация акцентируется на том факте, что физические сети, через которые путешествует дейтаграмма, могут различаться и даже быть несовместимыми. Узел *A1* кольцевой сети передает дейтаграмму через шлюз *G1*, направляя ее узлу *C1* Ethernet-сети. Шлюз *G1* пересылает данные по сети X.25 шлюзу *G2*, который и производит доставку на *C1*. Дейтаграмма проходит через три физически разделенных сети, но рано или поздно в целости и сохранности достигает узла *C1*.

Разбиение дейтаграмм

В процессе маршрутизации дейтаграммы через различные сети на одном из шлюзов может возникнуть необходимость в разбиении дейтаграммы на более мелкие фрагменты. Дейтаграмма, полученная из одной сети, может оказаться слишком крупной, чтобы ее вместил один пакет второй сети. Такая ситуация может возникать лишь в случаях, когда шлюз связывает физически различные сети.

Для каждой сети определяется значение *MTU* (maximum transmission unit, максимальная единица передачи), которое обозначает *максимально допустимый размер пакета* в этой сети. Если полученная из первой сети дейтаграмма длиннее значения MTU второй сети, она разбивается на ряд фрагментов с целью передачи. Данный процесс называется *разбиением дейтаграммы*. Представьте поезд, везущий стальные болванки. Каждый из вагонов поезда вмещает больше стали, чем грузовики, которые повезут груз дальше, по шоссе, поэтому каждый вагон поезда разгружается на множество грузовых машин. Сеть Ethernet точно так же физически отличается от сети X.25, как железная дорога от шоссе; протокол IP должен разделить относительно крупные пакеты Ethernet на более мелкие, прежде чем их можно будет передать по сети X.25.

Формат каждого фрагмента – такой же, как для обычной дейтаграммы. Второе слово заголовка обозначает фрагмент дейтаграммы и содержит информацию о том, как производить сборку фрагментов в целях восстановления исходной дейтаграммы. Поле *Идентификация* содержит информацию о том, к какой дейтаграмме принадлежит фрагмент, а поле *Смещение разбиения* – о том, каким по счету элементом является фрагмент дейтаграммы. В поле *Флаги* присутствует бит «Другие фрагменты» (More Fragments), позволяющий протоколу IP определить, что собраны все фрагменты исходной дейтаграммы.

Передача дейтаграмм в транспортный уровень

При получении дейтаграммы, адресованной локальному узлу, протокол IP обязан передать информативную часть дейтаграммы подходящему протоколу транспортного уровня. Задача решается при помощи *номера протокола*, указанного в третьем слове заголовка дейтаграммы. Каждому протоколу транспортного уровня присвоен уникальный номер протокола. С этими номерами и работает протокол IP. Речь о номерах протоколов пойдет в главе 2.

Как можно видеть из этого краткого обзора, на протокол IP возложены многие важные функции. Разумеется, приведенного описания недостаточно для полного понимания дейтаграмм, шлюзов, маршрутизации, IP-адресации и всех прочих элементов протокола IP: в каждой из последующих глав будет уделено больше внимания этим темам. А сейчас мы переходим к другим протоколам и уровню TCP/IP Internet.

ICMP, протокол управляющих сообщений Internet

Неотъемлемой частью IP является *протокол управляющих сообщений* (*Internet Control Message Protocol,* ICMP), определенный документом RFC 792. Данный протокол принадлежит уровню Internet и использует функциональность доставки дейтаграмм для отправки собственных сообщений. Сообщения ICMP выполняют следующие информативные, управляющие и связанные с ошибками функции TCP/IP:

Управление потоками данных (Flow control)

Если скорость поступления дейтаграмм слишком высока для обработки, узел-адресат или промежуточный шлюз отправляет ICMP-сообщение по-

давления источника (Source Quench Message) передающему узлу. Сообщение предписывает источнику временно прервать посылку дейтаграмм.

Обнаружение недостижимых адресатов

В случае когда сообщение не может быть доставлено адресату, система, обнаружившая эту проблему, отправляет сообщение Destination Unreachable источнику дейтаграммы. Если недостижимый адресат является сетью или узлом, сообщение исходит от промежуточного шлюза. Если пунктом назначения является порт, сообщение исходит от конкретного узла. (О портах мы поговорим в главе 2.)

Перенаправление маршрутов

ICMP-сообщение перенаправления (Redirect Message), исходящее от шлюза, предписывает узлу воспользоваться другим шлюзом, предположительно, по той причине, что это будет более эффективный выбор. Это сообщение может применяться лишь в случаях, когда узел-источник находится в одной сети с обоими шлюзами. Взгляните на рис. 1.7. Если узел сети X.25 направляет дейтаграмму шлюзу *G1*, этот шлюз может перенаправить узел к шлюзу *G2*, поскольку узел-источник и шлюзы *G1* и *G2* входят в одну сеть. С другой стороны, если источником дейтаграммы является узел сети Token Ring, *G1* уже не сможет перенаправить его к шлюзу *G2*. Как раз по той причине, что *G2* не входит в кольцевую сеть Token Ring.

Проверка состояния удаленных узлов

Узел может отправить ICMP-сообщение эхо (Echo Message), чтобы проверить работоспособность протокола Internet удаленной системы. Получив сообщение эхо, система возвращает данные из полученного пакета узлу-источнику. На основе сообщений Echo Message построена работа команды `ping`.

Транспортный уровень

Непосредственно над уровнем Internet расположен *транспортный уровень узел-узел*, или просто *транспортный уровень*. Наиболее важными протоколами транспортного уровня являются *протокол управления передачей* TCP (*Transmission Control Protocol*) и *протокол пользовательских дейтаграмм* UDP (*User Datagram Protocol*). TCP обеспечивает надежную доставку данных со сквозным обнаружением и устранением ошибок. UDP – это нетребовательная к ресурсам служба доставки дейтаграмм, работающая без образования логических соединений. Оба протокола выполняют доставку данных между прикладным уровнем и уровнем Internet. Разработчики приложений вольны выбирать, какая из этих служб точнее отвечает задачам приложения в каждом конкретном случае.

UDP, протокол пользовательских дейтаграмм

Протокол пользовательских дейтаграмм (User Datagram Protocol, UDP) дает прикладным программам прямой доступ к службе доставки дейтаграмм,

работающей подобно службе доставки IP. Приложения получают возможность обмениваться протоколами по сети с минимальными «накладными расходами».

UDP – это ненадежный протокол доставки дейтаграмм без организации логических соединений. Повторимся, характеристика «ненадежный» говорит лишь о том, что средствами протокола невозможно убедиться, что данные корректно получены адресатом. В пределах одного компьютера UDP выполняет доставку данных безукоризненно. Для доставки данных соответствующим процессам приложений в UDP применяются 16-битные номера для *исходного* и *целевого* портов в первом слове заголовка сообщения. Формат сообщения UDP приведен на рис. 1.8.

***Рис. 1.8.** Формат сообщения UDP*

Зачем разработчикам приложений использовать UDP в качестве службы доставки данных? Есть ряд веских причин. Если объем передаваемых данных невелик, издержки на создание соединений и отслеживание надежности доставки могут оказаться больше, чем затраты на повторную передачу всех данных. В таком случае UDP становится оптимальным протоколом транспортного уровня. Другими явными кандидатами на использование UDP являются приложения, работающие по принципу *запрос-ответ*. Ответ можно считать подтверждением приема запроса. Если ответ не поступил в течение определенного промежутка времени, приложение просто повторяет запрос. Кроме того, существуют приложения, в которых реализуются собственные методы обеспечения надежной доставки данных, так что они не ищут подобной функциональности в протоколах транспортного уровня. Для каждого из упомянутых типов приложений дополнительный уровень подтверждения приема станет причиной снижения производительности.

TCP, протокол управления передачей

Для обеспечения надежной доставки данных на уровне транспортного протокола в приложениях используется протокол TCP, проверяющий факт доставки данных по сети в нужном порядке. TCP – *надежный*, *потоковый протокол*, *требующий создания логических соединений*. Рассмотрим более подробно каждую из этих характеристик.

Надежность в TCP обеспечивает механизм *подтверждения приема с повторной передачей* (*Positive Acknowledgment with Retransmission*, PAR). Система,

в которой применяется PAR, повторяет отправку данных до тех пор, пока не получит от системы-адресата подтверждение, что данные успешно получены. Единицей обмена данными для взаимодействующих модулей TCP является *сегмент* (рис. 1.9). Каждый сегмент содержит контрольную сумму, посредством которой получатель определяет целостность данных. Если сегмент данных получен в целости и сохранности, получатель отправляет источнику *подтверждение*. Поврежденные сегменты данных просто игнорируются получателем. По истечении установленного интервала ожидания модуль-источник TCP повторно выполняет передачу всех сегментов, для которых не были получены подтверждения.

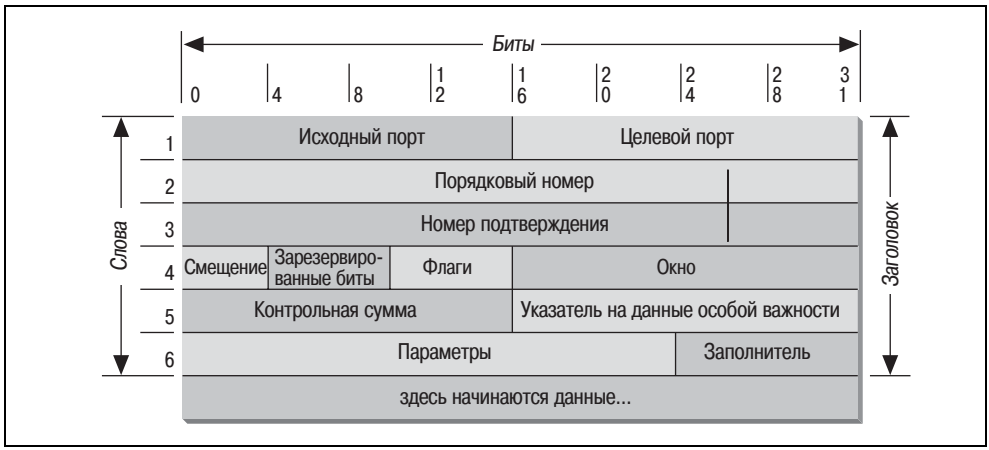

***Рис. 1.9.** Формат сегмента TCP*

TCP ориентирован на работу с соединениями. В целях обмена данными между двумя узлами образуется сквозное логическое соединение. Перед началом передачи данных (беседы) узлы обмениваются управляющей информацией (*рукопожатием*). Управляющий статус сегмента TCP отражается посредством соответствующего флага поля *Флаги* в четвертом слове заголовка сегмента.

В TCP применяется установление соединения с помощью *тройного рукопожатия*; производится обмен тремя сегментами. Простейший вариант тройного рукопожатия показан на рис. 1.10. Узел *A* открывает соединение, посылая узлу *B* сегмент с установленным битом «синхронизации порядковых номеров» (Synchronize sequence numbers, SYN). Сегмент сообщает узлу *B*, что *A* желает создать соединение и уведомляет о том, какой порядковый номер будет использоваться в качестве начального в сегментах *A*. (Порядковые номера применяются для сохранения порядка следования данных.) Узел *B* отвечает узлу *A* сегментом с установленными битами «подтверждения» (Acknowledgment, ACK) и синхронизации (SYN). Сегмент *B* подтверждает получение сегмента от *A*, а также уведомляет, какой порядковый номер станет начальным для сегментов *A*. Наконец, узел *A* передает сегмент, подтверждающий получение сегмента от *B*, а также первый блок непосредственно данных.

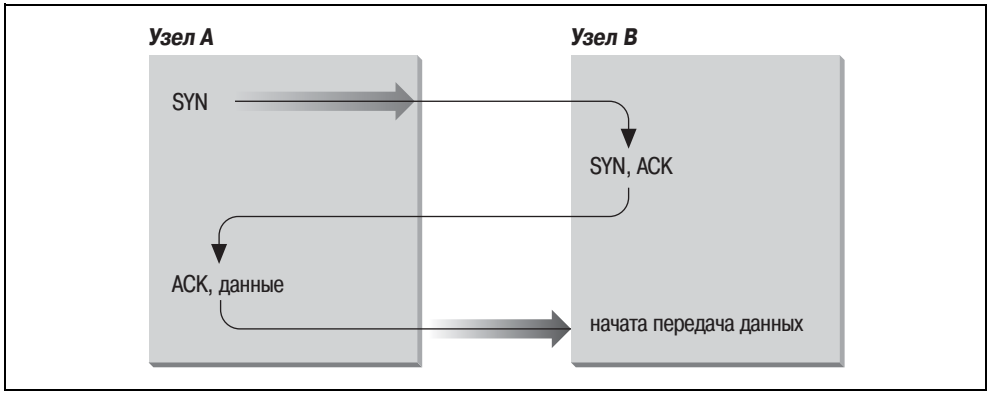

***Рис. 1.10**. Тройное рукопожатие*

После такого обмена TCP-модуль узла *A* обладает всеми свидетельствами того, что удаленный TCP-модуль функционирует и готов принимать данные. Как только соединение установлено, передача данных получает зеленый свет. После завершения передачи данных взаимодействующие модули обмениваются тройным рукопожатием, содержащим сегменты с так называемым битом FIN (No more data from sender, у источника больше нет данных), в целях закрытия соединения. Именно сквозной обмен данными становится логическим соединением между двумя системами.

В TCP данные считаются непрерывным потоком байтов, а не набором независимых пакетов. Следовательно, TCP предпринимает меры для сохранения последовательности отправки и получения байтов. Этой цели служат поля заголовка сегмента TCP – *Порядковый номер* и *Номер подтверждения*.

Стандарт TCP не определяет конкретных чисел, с которых должна начинаться нумерация; каждая система самостоятельно выбирает точку начала отсчета. Чтобы корректно отслеживать порядок в потоке данных, каждая из взаимодействующих сторон должна знать исходный номер второй стороны. Две стороны соединения синхронизируют системы нумерации байтов, обмениваясь SYN-сегментами в ходе рукопожатия. Поле *Порядковый номер* SYN-сегмента содержит *исходный порядковый номер* (*Initial Sequence Number*, ISN), который является точкой отсчета для *системы нумерации байтов*. Из соображений безопасности ISN следует быть случайным числом.

Каждый байт данных нумеруется последовательно, начиная с номера ISN, так что первый байт непосредственно данных имеет порядковый номер ISN+1. Порядковый номер в заголовке сегмента с данными указывает на порядковое положение в потоке данных первого байта данных сегмента. Например, если первый байт потока данных имел порядковый номер 1 (при ISN = 0), а 4000 байт данных уже получены адресатом, первый байт данных текущего сегмента является байтом 4001, и будет иметь порядковый номер 4001.

Подтверждающий сегмент (Acknowledgment Segment, ACK) выполняет две функции: *подтверждения приема* и *управления потоком*. Подтверждение

сообщает источнику, какой объем данных получен и сколько еще данных адресат способен принять. Номер подтверждения – это порядковый номер следующего байта, ожидаемый адресатом. Стандарт не требует создания подтверждения для каждого пакета. Номер подтверждения является подтверждением получения всех байтов, вплоть до этого номера. Например, если первый отправленный байт имел номер 1 и 2000 байт данных уже получены адресатом, номер подтверждения будет иметь значение 2001.

Поле *Окно* содержит значение *окна*, то есть количество байт, которое способен принять адресат. Если адресат способен принять еще 6000 байт, окно имеет значение 6000. Окно является указанием источнику, что можно продолжать передачу сегментов, если общий объем передаваемых байт меньше байтового окна адресата. Адресат управляет потоком байтов источника, изменяя размер окна. Нулевое окно предписывает отправителю прекратить передачу, пока не будет получено ненулевое значение окна.

На рис. 1.11 приведен поток данных TCP с нулевым значением исходного порядкового номера. Адресат получил и подтвердил получение 2000 байт, поэтому текущий номер подтверждения – 2001. Кроме того, адресат обладает возможностью принять еще 6000 байт, а потому предъявляет окно со значением 6000. Источник отправляет сегмент размером в 1000 байт с порядковым номером 4001. Для байтов 2001 и последующих еще не были получены подтверждения, однако источник продолжает передачу данных, пока не исчерпаны ресурсы окна. Если на момент заполнения окна источником для уже отправленных данных не получены подтверждения, по истечении определенного интервала ожидания источник повторно передает данные, начиная с первого неподтвержденного байта.

В отсутствие последующих подтверждений повторная передача начнется с байта 2001. Данный метод гарантирует надежность доставки данных адресату.

Кроме того, TCP отвечает за доставку полученных от IP данных соответствующему приложению. Приложение, которому предназначаются данные, обозначается 16-битным числом, *номером порта*. Значения *Исходный порт*

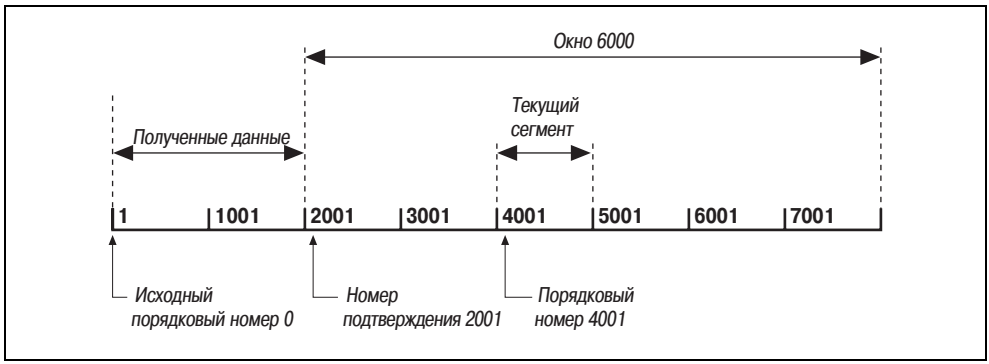

Рис. 1.11. *Поток данных TCP*

и *Целевой порт* содержатся в первом слове заголовка сегмента. Корректный обмен данными с прикладным уровнем – это важная составляющая функциональности служб транспортного уровня.

Прикладной уровень

Вершиной архитектуры протоколов TCP/IP является *прикладной уровень*, в который входят все процессы, использующие протоколы транспортного уровня в целях доставки данных. Существует большое число прикладных протоколов. Большинство из них связаны с пользовательскими службами, а число служб этого уровня постоянно растет.

Перечислим наиболее широко известные и распространенные прикладные протоколы:

Telnet

Протокол сетевых терминалов (Network Terminal Protocol) обеспечивает диалоговую работу с удаленными системами по сети.

FTP

Протокол передачи файлов (File Transfer Protocol) используется для передачи файлов в диалоговом режиме.

SMTP

Простой протокол передачи почты (Simple Mail Transfer Protocol) обеспечивает доставку сообщений электронной почты.

HTTP

Протокол передачи гипертекста (Hypertext Transfer Protocol) выполняет доставку веб-страниц по сети.

HTTP, FTP, SMTP и Telnet – наиболее широко распространенные приложения TCP/IP. При этом как пользователи, так и системные администраторы в своей работе встречаются и со многими другими. Вот некоторые из не менее широко применяемых приложений TCP/IP:

Domain Name System (DNS, Система доменных имен)

Известное также в качестве *службы имен*, это приложение ассоциирует IP-адреса с именами, назначаемыми сетевым устройствам. Система DNS подробно рассматривается в этой книге.

Open Shortest Path First (OSPF, Протокол предпочтения кратчайшего пути)

Маршрутизация является одним из несущих элементов конструкции TCP/IP. Протокол OSPF используется сетевыми устройствами для обмена информацией по маршрутизации. Маршрутизация также является одной из основных тем этой книги.

Network File System (NFS, Сетевая файловая система)

Данный протокол позволяет организовывать совместный доступ к файлам многих узлов сети.