I

TCP/IP

В этой части...

1

Фундамент Internet

Совершенно очевидно, что открывший эту книгу читатель принадлежит к одной из двух категорий людей — новичков или ветеранов изучения основ стратегии безопасной работы в сетях. Тема протокола IP (Internet Protocol — протокол Internet) чрезвычайно обширна и может отпугнуть начинающих, если не представить ее простым, понятным языком, постепенно объясняя неизвестные аббревиатуры, понятия и принципы работы. Поэтому основной целью первой главы является именно такое представление нового (для кого-то) материала. Для рассматриваемого здесь набора протоколов чаще используют аббревиатуру *TCP/IP* (Transmission Control Protocol/Internet Protocol — протокол управления передачей/протокол Internet). С помощью этого набора протоколов осуществляется взаимодействие между компьютерами по глобальной сети Internet. Существуют и другие протоколы для обмена данными по сети, например, протокол AppleTalk для компьютеров Apple. Как правило, такие протоколы применяются в корпоративных локальных сетях (intranet-сеть). Большинство соединений в Internet осуществляется с использованием стека протоколов TCP/IP, который признан стандартом для взаимодействия между компьютерами в глобальной сети.

У опытных специалистов, которые ежедневно работают с TCP/IP, может возникнуть желание пропустить эту главу, но мы все же рекомендуем хотя бы пролистать ее страницы. Тот, кому приходилось объяснять принципы работы IP (может быть, ваш непосредственный начальник), несомненно, оценит наш подход к изложению материала. Тот же, кто не уверен, в полноте своих знаний, наверняка, почерпнут много полезных сведений из этой вступительной главы.

Здесь, в одной главе, сконцентрирована общая информация о TCP/IP. Многие затронутые темы будут подробнее раскрыты в следующих главах, но сначала следует получить теоретическую базу для их уверенного восприятия. Ниже перечислены основные темы первой главы.

- **Базовая концепция TCP/IP**. Рассмотрены основы взаимодействия компьютеров по Internet с помощью стека протоколов TCP/IP.
- **Упаковка данных для их передачи по Internet**. В этом разделе изучаются методы инкапсуляции данных для их пересылки получателям, находящимся в различных сегментах сети.
- **Физические и логические адреса**. Освещена тема идентификации компьютера или хоста, подключенного к Internet.
- **Служба DNS (Domain Name System — система доменных имен)**. Основное внимание уделяется важности прямого и обратного преобразования символьных имен компьютеров и их IP-адресов.
- **Маршрутизация**. Изучаются методы выбора маршрута между двумя компьютерами при обмене данными по сети.

Модель TCP/IP

Существует множество мотивов, по которым пользователи очень часто получают информацию через Internet (взять хотя бы просмотр Web-страниц с удаленного Web-сервера). При этом кажется, что получение данных происходит практически мгновенно, но в процессе доставки информации незаметно для пользователя участвует множество устройств и выполняется масса отдельных задач.

Уровни

На рис. 1.1 изображена логическая схема взаимодействия двух удаленных компьютеров согласно модели TCP/IP. Итак, допустим, что нам нужно загрузить Web-страницу на компьютер, которому соответствует элемент блок-схемы, обозначенный как Web-броузер (см. рис. 1.1). Прежде чем отправить запрос на получение данных Web-страницы Web-серверу, на стороне отправителя этот запрос следует упаковать. Данные передаются на самый низкий уровень стека, проходя упаковку на каждом из промежуточных уровней. При этом на каждом уровне к сообщению добавляется определенная информация. Затем полученный пакет пересылается по Internet. На стороне компьютера-адресата сообщение распаковывается в обратном порядке, проходя через все уровни к самому верхнему. Каждый уровень использует предназначенную для него информацию. Оставшаяся часть сообщения передается на более высокий уровень вплоть до уровня приложений (элемент схемы, обозначенный как Web-сервер).

Кратко рассмотрим каждый из уровней модели TCP/IP.

Уровень приложений (application layer) является высшим уровнем модели TCP/IP. На этом уровне реализуется доступ приложений (в нашем примере Web-броузера и Web-сервера) к компьютерной сети.

Транспортный уровень (transport layer) расположен ниже уровня приложений. На этом уровне устанавливаются многие параметры взаимодействия двух компьютеров и обеспечивается надежная работа других, по своей сути ненадежных, уровней. Данный уровень служит посредником между уровнем приложений и нижними уровнями, ориентированными на передачу данных по сети.

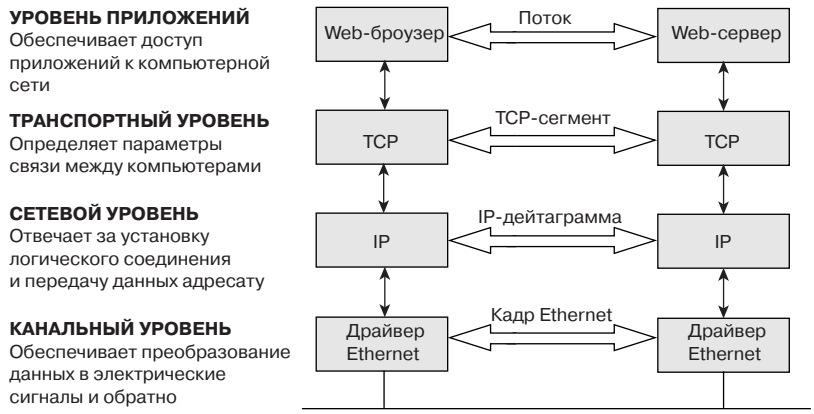


Рис. 1.1. Модель стека протоколов TCP/IP

Мы рассмотрим два протокола транспортного уровня: TCP, который гарантирует надежную доставку сообщений, и UDP (User Datagram Protocol — протокол доставки пользовательских дейтаграмм), который такой надежной доставки не гарантирует. В нашем примере требовалось использование TCP, так как потеря данных недопустима.

Сетевой уровень (network layer) отвечает за пересылку данных с одного компьютера на другой (в нашем случае запрос пересылается на Web-сервер) нередко только через один транзитный участок или переход (hop). *Переходом* мы будем называть участок сети между компьютером и маршрутизатором или между двумя маршрутизаторами на пути доставки пакета к адресату.

Канальный уровень (link layer) является самым низким в иерархии стека TCP/IP. На этом уровне обеспечивается взаимодействие с физической средой передачи данных. В нашем случае двоичные данные преобразовываются в электрические сигналы, так как физической средой передачи является Ethernet. Для получения и отправки данных используется определенный интерфейс.

Обмен данными

Еще раз обратимся к рис. 1.1. Теоретически процесс передачи данных описывается следующим образом. Запрос на получение Web-страницы проходит через уровни компьютера-отправителя (которые часто называют стеком TCP/IP) “сверху вниз”. Сообщение направляется компьютеру-адресату, где оно проходит обратное преобразование по стеку TCP/IP “снизу вверх”. На рис. 1.1 вертикальные стрелки между уровнями обозначают поток данных на локальном компьютере. Горизонтальные стрелки указывают на то, что каждый уровень передает определенную информацию (упаковывает сообщение) соответствующему уровню на удаленном компьютере. Несмотря на то что два компьютера не взаимодействуют непосредственно между собой, применение стека TCP/IP создает у пользователя такое впечатление.

Эта концепция крайне важна для правильного понимания материала этой главы и всей модели TCP/IP. Поэтому повторим основные моменты и закрепим терминологию. Термин *стек TCP/IP* используется для описания многоуровневой

модели обработки запросов и ответов. *Инкапсуляцией* называется упаковка сообщения одного протокола в сообщение другого и добавление к нему определенной информации (например, идентифицирующих заголовков), предназначенной для соответствующего уровня на удаленном компьютере. Каждый уровень на компьютере-отправителе добавляет к сообщению собственный заголовок, и на компьютере-получателе в первую очередь учитывается заголовок пакета для соответствующего уровня. Полученное сообщение проходит обратный процесс распаковки с удалением заголовков на каждом из уровней до тех пор, пока, наконец, не будет достигнут самый высокий уровень. При ответе на запрос процесс повторяется в обратном порядке, начиная с упаковки ответного сообщения на хосте Web-сервера и заканчивая его распаковкой и передачей уровню приложений, поддерживающему Web-броузер на компьютере-получателе.

Упаковка

Данные, обмен которыми осуществляется между двумя хостами, должны быть сохранены в каком-то формате, стандартном для каждого уровня стека TCP/IP. *Хост* (host) — это общий термин, которым можно назвать рабочую станцию, маршрутизатор, Web-сервер и т.д. Общей особенностью хостов является наличие соединения с сетью, по которой можно обмениваться данными. В общем случае все упакованные данные называются *пакетом* (package). Проблемы с терминологией возникают из-за того, что на каждом уровне стека TCP/IP при взаимодействии двух удаленных приложений под этим пакетом понимается различная информация (учитывая добавленные служебные заголовки). В этом разделе мы рассмотрим базовые понятия, касающиеся упаковки данных, а именно: бит, байт, пакет, инкапсуляция и интерпретация данных.

Биты, байты и пакеты

Наименьшей единицей информации принято считать *бит*. Значением бита может быть 0 или 1, поэтому бит часто называют двоичной цифрой (binary). Ясно, что в одном бите нельзя передать достаточный объем информации, поэтому их группируют по восемь. Восемь битов составляют один *байт*. Минимальный объем информации, пусть даже увеличенный в восемь раз, все равно остается недостаточно большим, но один байт способен хранить значение стандартного символа ASCII, например буквы или знака препинания, или целого числа до 255 (2^8–1).

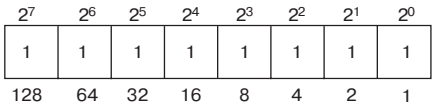


Рис. 1.2. Схема байта

На рис. 1.2 показана схема байта. Нас больше всего интересуют биты, для которых используется двоичный код — набор нулей и единиц. Каждый бит можно представить некоторой степенью основы двоичной системы счисления — числа 2. Значение байта составляет диапазон степеней числа 2 от 2^0 до 2^7. Это пояснить довольно просто: если значением всех битов одного байта является 0, то и значение байта равно 0, если же значением всех битов

одного байта является 1, то, сложив все значения степеней битов, начиная с наименьшего (2^0=1), получим 1+2+4+8+16+32+64+128=255 — максимальное значение одного байта. Проанализируем смысл этого значения позже, при обсуждении IP-адресов.

Только что мы выполнили преобразование двоичного значения в десятичное. Для преобразования байта данных из двоичного вида в десятичный достаточно представить его в виде степеней числа 2 и простым сложением полученных значений каждого бита получить искомое десятичное значение. Вот и весь секрет. Это не так сложно, как запуск ракеты в космос.

Для передачи по сети несколько байтов объединяются в один пакет. На рис. 1.3 показана истинная ситуация при передаче данных по сети — передача любого количества полезных данных обеспечивается за счет добавления определенного объема служебной информации. Требуются некоторые действия для упаковки данных перед отправкой по сети и их последующей распаковки на стороне адресата (и, конечно, для подтверждения достоверности сообщения). Для проверки целостности переданного пакета предназначено специальное поле CRC (Cyclic Redundancy Check — циклическая проверка четности с избыточностью), значение которого часто называют *контрольной суммой*.

Заголовок содержит информацию об адресате и отправителе, а также о типе передаваемой информации, что напоминает обычный почтовый конверт

Рис. 1.3. Пакет данных

Как и почтовый конверт, IP-пакет должен нести информацию об адресате и отправителе (см. рис. 1.3). В сетях, по крайней мере в сетях Ethernet, аналогом домашнего адреса можно считать MAC-адрес (Media Access Controller — контроллер доступа к среде) вашего сетевого адаптера. Этот аппаратный адрес присваивается производителем сетевого оборудования. MAC-адрес представляет собой 48-битовое число, т.е. может быть достаточно большим (2^{48}–1). О том, чем различаются IP- и MAC-адреса, рассказано в разделе "Адреса" этой главы.

Для создания *кадра* (frame) к нему должна быть добавлена информация заголовков каждого из уровней TCP/IP. В последнюю очередь к кадру добавляется информация физического уровня, и он передается в линию связи с помощью сетевого адаптера (NIC — network interface card). Заголовок кадра имеет размер 14 байт и содержит поля для хранения MAC-адресов отправителя и адресата, служебную информацию кадра (ее размер может изменяться) и 4-байтовую завершающую часть (окончание) для передачи кода CRC.

Инкапсуляция

Рассмотрим схему многоуровневой упаковки сообщения (рис. 1.4). Теоретически различные уровни стека протоколов одного компьютера "обращаются" к соответствующим уровням на другом компьютере. Уровни расположены один над другим, что позволяет говорить о "стеке протоколов TCP/IP". На каждом уровне пакет состоит из

собственного заголовка и самих данных, которые часто называют *полезной нагрузкой* (payload). Смысл всего процесса инкапсуляции заключается в передаче на другой компьютер какой-то информации, но на пути к адресату на каждом из уровней заголовки добавляются к уже существующей информации. Заголовки вышележащего уровня считаются данными для более низкого уровня.

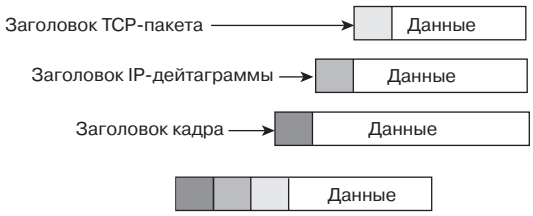


Рис. 1.4. Добавление заголовков

Предположим, что нам нужно послать сообщение или передать какую-либо информацию на удаленный компьютер. Сначала эта информация формируется в единое целое с помощью программы типа `telnet` или программы для работы с электронной почтой (более подробно эти программы рассмотрены ниже, в разделе "Протоколы стека TCP/IP"). TCP-пакет, который называют TCP-сегментом (TCP segment), состоит из TCP-заголовка и данных. UDP-пакет называют дейтаграммой, что часто проводит к путанице, так как дейтаграммой называется и пакет, формирующийся на сетевом уровне (протокол IP).

TCP-сегмент передается вниз по стеку протоколов на уровень протокола IP. На сетевом уровне в начало TCP-сегмента добавляется информация заголовка IP, и, таким образом, формируется IP-дейтаграмма. В действительности и заголовок, и данные TCP на уровне протокола IP рассматриваются как единый блок данных. IP-дейтаграмма передается на канальный уровень стека TCP/IP, где превращается в кадр — в начало IP-дейтаграммы добавляется заголовок кадра, содержащий служебную информацию для доставки этого кадра по физической среде передачи, например, по сети Ethernet.

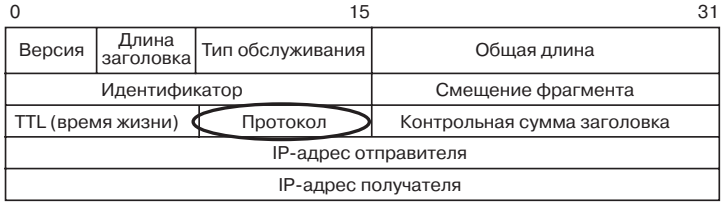
Когда пакет достигает компьютера-адресата, весь описанный процесс повторяется с точностью до наоборот — при прохождении сообщения снизу вверх по стеку TCP/IP на каждом из уровней удаляется соответствующий заголовок. С помощью заголовков осуществляется обмен информацией между аналогичными уровнями стека TCP/IP взаимодействующих компьютеров.

Интерпретация полученных данных

При прохождении сообщения по стеку протоколов его информация представляет собой набор нулей и единиц. Как же понять, что скрывается за этой последовательностью? Представим себе, например, заголовок IP-дейтаграммы. Как узнать, какой протокол был использован на более высоком уровне? Безусловно, эти сведения позволяет понять принципы работы протокола. В данном случае под термином *протокол* понимается набор согласованных правил или форматов об-

мена сообщениями. В каждом протоколе (например, IP, TCP, UDP или ICMP) используется собственный вариант схемы обмена данными и соответствующий формат этих данных.

Рассмотрим заголовок IP-дейтаграммы (рис. 1.5). Для каждого поля заголовка зарезервировано определенное количество битов. В поле "Протокол" сохраняется информация о протоколе более высокого уровня. Каждая строка IP-заголовка содержит 32 бит (с 0 по 31 включительно), что равняется четырем байтам. Счет с нуля несколько усложняет задачу определения места нужного байта или бита, но к этому придется привыкнуть. В первой строке отображены байты с 0 по 3, во второй — с 4 по 7, а в третьей — с 8 по 11. Обратите внимание на то, что выделенное поле "Протокол" находится в третьей строке. Длина предшествующего поля TTL составляет 1 байт. Этот байт является восьмым. Следовательно, поле "Протокол", длина которого тоже 1 байт, является 9-м байтом. Это значит, что для определения протокола, использованного на более высоком уровне, необходимо исследовать этот 9-й (по сути, 10-й, ведь счет начинается с 0) байт. То есть весь секрет в том, чтобы знать, в каком месте пакета определенного уровня следует искать необходимую информацию; расположение конкретного поля можно найти по известному смещению.



Заголовок IP-дейтаграммы без параметров общей длиной 20 байт

Рис. 1.5. Поля IP-дейтаграммы

Итак, мы определили место расположения поля "Протокол". Что же оно собой представляет и для каких целей служит? Значение этого поля указывает на то, какой протокол использовался для инкапсуляции данных на более высоком уровне. Предположим, что значение байта этого поля равно 17. На самом деле в поле хранится шестнадцатеричное значение, например 11, которое соответствует десятичному значению 17. Это означает, что на транспортном уровне был использован протокол UDP. Значение 6 свидетельствует, что встроенным пакетом является TCP-пакет, а значение 1 соответствует ICMP-пакету.

Шестнадцатеричная система счисления

Основой двоичной системы счисления является число 2, а двоичный код состоит из нулей и единиц. Именно двоичный код используется при работе всех компьютеров. Так зачем же нужно создавать дополнительные сложности и вводить новую систему счисления, основой которой является число 16? Проблема заключается в том, что при использовании двоичной системы счисления для записи большого числа приходится использовать значительное количество битов, и числа быстро становятся слишком громоздкими. Шестнадцатеричная система позволяет представить двоичные числа в более кратком виде. Один шестнадцатеричный символ позволяет заменить четыре двоичных разряда (2^4=16).

Рассмотрим, например, поле заголовка протокола IP размером в 8 бит. Его значение можно преобразовать в два шестнадцатеричных символа. Как уже указывалось, двоичное число 17 соответствует использованному протоколу UDP. Как же получить из десятичного 17 шестнадцатеричное 11?

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	1	0	0	0	1

Здесь показаны степени числа 2, составляющие десятичное число 17 (для получения этого числа нужно сложить $2^4=16$ и $2^0=1$). Эти биты разделены на два шестнадцатеричных числа, по 4 бит каждый. Крайние слева четыре бита (или шестнадцатеричный символ), которые называют еще битами старшего разряда (most significant), имеют значение 0001. Аналогично четыре крайних справа бита, называемых битами младшего разряда (least significant), тоже имеют значение 0001. Каждый шестнадцатеричный символ может принимать значения от 0 до 15. Каждый значащий разряд шестнадцатеричного числа в 16 раз больше предыдущего разряда. Поэтому при делении десятичного числа (17) на 16 число, получившееся в остатке (1), является младшей шестнадцатеричной цифрой. Продолжим деление, записывая остаток слева от первой цифры, и получим шестнадцатеричное число 11. Для того чтобы шестнадцатеричные значения можно было отличить от десятичных, к ним обычно добавляют приставку 0x, например 0x11.

Адреса

Скорее всего, вам знаком термин *IP-адрес*. Но что он действительно означает и для чего предназначен? Как именно один хост направляет информацию другому? Ответы на эти и другие вопросы можно найти в этом разделе.

Физические адреса

Можно искать физические MAC-адреса отправителя и получателя в заголовке IP-пакета до посинения, но так ничего и не обнаружить. MAC-адреса не имеют никакого отношения к протоколу IP, в котором используются логические адреса. В некоторых случаях MAC-адресов может вообще не существовать.

MAC-адреса используются для идентификации сетевого адаптера Ethernet в сети. Сетевой адаптер сам по себе никак не зависит от существования IP, заголовков IP-пакетов или логических IP-адресов. Итак, мы столкнулись с проблемой несоответствия двух понятий. Очевидно, что для реализации взаимодействия двух компьютеров должен существовать способ установки соответствия между логическими IP-адресами и физическими MAC-адресами.

Знаете ли вы IP-адрес вашего настольного компьютера? Если нет, то в этом нет ничего страшного или необычного. Все дело в том, что на сегодняшний день большинство наших компьютеров не имеют постоянного IP-адреса. Получить и зарезервировать определенный IP-адрес (или диапазон адресов) очень дорого. Большинству компьютеров IP-адреса выделяются на время одного сеанса работы в сети, для чего провайдер услуг Internet (ISP) использует специальное программное обеспечение, например протокол DHCP (Dynamic Host Configuration Protocol — протокол динамической конфигурации хостов).

Выделение IP-адреса

Протокол DHCP позволяет осуществлять динамическое назначение компьютерам IP-адресов и отказаться от трудоемкого процесса управления закрепленными за каждым хостом статическими IP-адресами. С помощью DHCP осуществляется централизованное и автоматическое назначение временных IP-адресов, что значительно повышает эффективность управления крупными сетями. Это очень удобно для администратора сети, но усложняет работу администратора системы безопасности (например, при поиске компьютеров с временными IP-адресами, которые являются источником потенциально опасных действий).

Каково же общее количество возможных IP-адресов? Их точное число — 2^{32} (так как адрес состоит из 32 бит), т.е. более 4 млрд. Однако не все IP-адреса доступны, существуют зарезервированные диапазоны адресов. Во время стремительного роста популярности Internet пришлось с грустью констатировать, что количество свободных IP-адресов быстро уменьшается. Встал вопрос: как же выйти из создавшегося положения?

Первый способ заключается в использовании DHCP отдельными сетевыми узлами, которые выделяют IP-адреса во временное пользование. Это позволяет снизить общее количество необходимых одновременно IP-адресов. Альтернативным способом является применение *зарезервированных адресов для частных сетей*. Организация IANA (Internet Assigned Numbers Authority — полномочный комитет по надзору за присвоением номеров Internet) зарезервировала некоторые диапазоны адресов для использования только во внутренних локальных сетях. Например, IP-адреса, начинающиеся с 192.168 и 172.16, могут использоваться только для обмена информацией между хостами в рамках локальной сети. Такой трафик не должен проходить через внешний шлюз конкретного узла. Этот метод позволяет сэкономить IP-адреса для отдельного узла Internet и использовать для его внутренних целей указанные адреса сетей класса B.

Продолжим. Некоторые из наших читателей знают IP-адрес своего компьютера. Уже хорошо. Но знаете ли вы на память свой MAC-адрес? Скорее всего, нет. Не так-то просто запомнить 48-битовый адрес, да это и не нужно.

Для преобразования физических MAC-адресов в логические IP-адреса предназначен протокол ARP (Address Resolution Protocol — протокол преобразования адресов). ARP как таковой не является протоколом Internet. Он служит для отправки кадра Ethernet (запроса) всем компьютерам данного сегмента сети. Такой запрос называют *широковещательным*. Широковещательное сообщение отправляется всем компьютерам сегмента или всей сети. Стоит особо подчеркнуть, что протокол ARP предназначен для опроса только хостов только локальной сети и не может применяться для обмена данными между хостами разных сетей.

Компьютер-отправитель посылает широковещательный ARP-запрос всем компьютерам локальной сети о наличии у них MAC-адреса, соответствующего определенному IP-адресу. Компьютер, которому предназначено сообщение, в ответ на запрос отправляет свой MAC-адрес. В результате такого обмена информацией и запрашивающий, и ответивший, а также все другие компьютеры, подключенные к сети, заносят новую запись в свои ARP-таблицы соответствий физических и логических адресов. Создание такой таблицы позволяет сократить количество новых ARP-запросов. В конечном итоге, все компьютеры сегмента локальной сети будут взаимодействовать с помощью MAC-, а не IP-адресов. При транзакции по стеку TCP/IP обмен информацией может осуществляться между соответствующими уровнями стека, но при действительной доставке данных используются именно MAC-адреса двух хостов.

Почему же MAC-адреса столь объемны? Ведь 48 бит дают огромное число возможных значений. Такой размер был выбран для получения абсолютно уникальных и "вечных" адресов. Эта фраза хороша только на слух, и уже сейчас планируется расширить длину MAC-адреса до 128 бит с целью удовлетворения существующих требований указания кода производителя в MAC-адресах сетевых адаптеров.

Логические адреса

IP-адрес состоит из 32 бит, уникально идентифицирующих хост. Это число записывается в виде четырех десятичных чисел, разделенных точками (например 192.168.5.5). Каждое из четырех чисел выбирается не случайно. Первая часть IP-адреса указывает на сеть, к которой подключен данный хост, а оставшиеся числа определяют место размещения хоста в этой сети. Различают несколько классов адресов. Класс определяет предельное число хостов, которые могут быть подключены к данной сети, или количество битов в уникальном IP-адресе хоста сети (табл. 1.1). Так, в адресах класса A предназначено 8 бит для сетевой части IP-адреса, а оставшиеся 24 бит — для определения конкретного хоста в этой сети. Таким образом, в сетях класса A могут работать более 16 млн. хостов (2^{24}–1). В качестве примера сети класса A может послужить сеть Массачусетского технологического института с диапазоном IP-адресов от 18.0.0.0 до 18.255.255.255.

Таблица 1.1. Классы IP-сетей

Класс	Сетевая часть адреса, бит	Узловая часть адреса, бит	Число хостов сети
A	8	24	более 16 млн.
B	16	16	более 65 тыс.
C	24	8	255

IP-адреса также классифицируют, начиная с класса A и заканчивая классом E. Классы A, B и C используются для хранения уникальных адресов. Адрес такого класса выделяется для одного конкретного компьютера. Адреса класса D являются широковещательными и предназначены для отправки пакетов указанной группе хостов. Адреса класса E зарезервированы для исследовательских целей. В табл. 1.2 перечислены классы IP-адресов и зарезервированные для них диапазоны.

Таблица. 1.2. Классы адресов

Класс	Начальный IP-адрес	Конечный IP-адрес
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

Использование CIDR

Вы могли уже встречать новый термин, касающийся назначения IP-адресов, — CIDR (Classless Inter-Domain Routing — бесклассовая междоменная маршрутизация). На протяжении длительного периода все IP-адреса должны были принадлежать к определенному классу, т.е. одна сеть могла объединять или чуть более 16 млн. хостов, или до 65534 хостов, или до 255 хостов. Однако чаще всего возникала потребность в сетях класса B, в которых работало от 255 до 65534 хостов. При этом многие IP-адреса таких сетей оставались свободными. С учетом того, что количество IP-адресов ограничено, необходимо было устранить проблему выделения лишних IP-адресов для сетей определенного класса.

CIDR представляет собой метод увеличения адресного пространства в Internet. С помощью CIDR устраняется 8-битовое ограничение на минимальную длину сетевой части адреса. В случае применения метода CIDR разделительная линия между сетевой и узловой частями

адреса может отделять любое число разрядов 32-битового адреса. При этом выделенный узлу диапазон сетевых адресов указывается уникальным образом. Например, запись всех адресов сети 192.168 по методу CIDR будет выглядеть следующим образом: 192.168/16. Первая часть этой записи является десятичной записью набора двоичных разрядов всей сети. После косой черты указывается число битов, представляющих сетевую часть IP-адреса. В данном случае CIDR-адрес аналогичен адресу сети класса B, но его легко изменить для обозначения менее крупной сети.

Маска подсети

Еще одним важным понятием является термин *маска подсети*. С помощью этой маски указывается, сколько битов в IP-адресе относится к сетевой части, а сколько — к хосту. Каждый бит сетевой части "маскируется" с помощью 1. Например, адрес сети класса A имеет 8-битовую сетевую часть и 24-битовую часть для представления хоста. Восемь последовательных единиц в двоичном коде соответствуют десятичному значению 255. Значит, маской подсети в данном случае будет 255.0.0.0. Соответственно, для сетей класса B маской подсети будет 255.255.0.0, а для сетей класса C — 255.255.255.0. Зачем же они нужны, если можно определить класс и число битов, зарезервированных для сети, проанализировав IP-адрес? Дело в том, что сетевые администраторы могут разбивать свои сети на несколько подсетей. Например, сеть класса C можно разбить на несколько отдельно адресуемых подсетей с помощью определенной маски подсети.

Порты

В соответствующих полях заголовков своих пакетов протоколы TCP и UDP сохраняют 16-разрядные номера портов. Это означает, что в этих полях может содержаться 65536 вариантов номера порта или службы: от 0 до 65535. Очень важно запомнить, что хотя каждой службе назначается стандартный номер порта, нет никакой гарантии, что так и будет в действительности. За службой `telnet`, например, почти повсеместно закреплен TCP-порт 23. Но не существует никаких препятствий для изменения этого номера на, скажем, 31337. А для хакера, взломавшего компьютер, лучшим способом скрыть свое присутствие будет выполнение своих действий через нестандартный порт. Если хакер запустит `telnet` на каком-нибудь порту с большим номером, то обнаружить несанкционированный доступ будет значительно сложнее. Любая служба может связаться с любым портом. Но если вы хотите обмениваться информацией с другими компьютерами, то все же лучше использовать стандартные порты. Для хостов под управлением UNIX для закрепления номеров TCP- и UDP-портов за стандартными, популярными службами используется информация файла `/etc/services`.

Рассмотрим пример типичной информации файла `/etc/services`. В данном фрагменте файла указаны названия служб и выделенные этим службам порты. Обратите внимание на то, что служба `domain` (это служба DNS) может работать и по протоколу TCP, и по UDP. Это кажется необычным, но вполне нормально. Большинство служб используют или тот, или другой протокол, но есть и исключения, как, например, служба DNS.

```
ftp........21/tcp
telnet.....23/tcp
smtp.......25/tcp
```

```
domain.....53/udp
domain.....53/tcp
```

Посмотрим, как служба указывается в пакете (рис. 1.6). В заголовке UDP-пакета содержится 16-битовое поле под названием “Порт получателя”. В этом поле определяется порт на удаленном компьютере, которому передается сообщение. В нашем примере значением этого поля является 53, т.е. дейтаграмма предназначена для службы DNS.

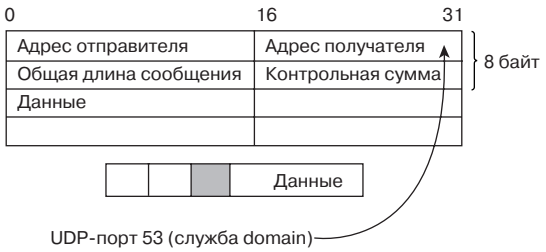


Рис. 1.6. Порт получателя в заголовке пакета UDP

Ранее особое значение придавалось портам с номерами до 1024, которые также называли *доверенными* (смешно звучит?) или *привилегированными*, так как эти порты были выделены для системных процессов. В этом был определенный смысл, пока сеть Internet была достаточно безопасной. На сегодняшний день все большее значение приобретают порты с номерами больше 1024, которые еще называют *временными портами* (ephemeral port) из-за того, что они могут использоваться любой службой по любой причине.

Протоколы стека TCP/IP

Еще раз вернемся к четырем базовым уровням модели TCP/IP (см. рис. 1.1). Взаимодействие со стеком TCP/IP осуществляется посредством какого-либо приложения. Для обмена файлами используется программа наподобие FTP, в качестве эмулятора терминала — `telnet`, а для пересылки текстовых сообщений (например, последнего анекдота сразу 50 друзьям) — приложение электронной почты. Такое приложение принимает сообщение пользователя и упаковывает его для передачи вниз по стеку TCP/IP.

Теперь рассмотрим методы передачи данных на транспортном уровне. Они реализуются на основе протоколов, ориентированных (TCP) или не ориентированных (UDP) на установление соединений. “Ориентированный на установление соединений” означает, что данный протокол выполняет максимальный объем действий для надежной доставки сообщений и полного согласования параметров соединения (handshake). Протокол, не ориентированный на установление соединений, отправляет сообщения “на удачу” и не гарантирует их доставку адресату. При этом надежность должна обеспечиваться на уровне приложений. В табл. 1.3 приведены некоторые свойства TCP и UDP.

UDP является одним из простейших протоколов, он только собирает пакеты и выдает их в сеть. На стороне получателя эти пакеты принимаются, разбираются

(последовательно удаляются заголовки сообщения при его передаче вверх по стеку), и пользователю выдается исходное сообщение. Конечно, на маршруте следования несколько пакетов могут оказаться утерянными, но часто это не вызывает серьезных последствий и даже допустимо. Например, при широковещательном распространении аудиоинформации потеря одного слова не приведет к утрате общего смысла текста. При передаче видеоданных на утерянный пакет укажет пустое место в окне изображения. В большинстве случаев такие потери не выходят за рамки допустимых. Передача данных с помощью UDP не означает, что они обязательно будут утеряны, просто сам протокол UDP не гарантирует их надежной доставки. Приложение может игнорировать утерянные пакеты, а может выдать запрос на их повторную передачу.

Таблица 1.3. Сравнение TCP и UDP

TCP	UDP
Надежный	Ненадежный
Ориентирован на установление соединений	Не ориентирован на установление соединений
Медленный	Быстрый

А что делать, если даже минимальная потеря данных недопустима? В этом случае нужно использовать протокол TCP, который гарантирует надежную доставку данных по назначению. Для проверки доставки пакетов в правильном порядке в TCP предусмотрено несколько механизмов. Одним из них является метод подтверждения получения.

Сообщение о подтверждении получения (ACK — acknowledgement) является важным элементом работы протокола TCP. За счет уведомления о получении каждого пакета и обеспечивается надежность TCP. Если пакет не был доставлен (то есть подтверждения не получено), то он отправляется повторно. Таким образом, TCP гарантирует доставку всех пакетов и поэтому считается надежным. Он работает чуть медленнее, чем UDP, но с помощью правильных настроек можно немного ускорить процесс обмена данными.

Последним из рассматриваемых протоколов будет ICMP (Internet Control Message Protocol — протокол управляющих сообщений Internet). Этот удобный простой набор приложений был первоначально разработан для поиска неисправностей в сетях и уведомления об ошибках. Самой известной программой из набора ICMP является программа проверки доступности адресата с помощью эхо-запроса (известная как `ping`). Благодаря ICMP также осуществляется управление потоком, изменение маршрута сообщений и сбор информации о сети (и это только некоторые из доступных функций). Более подробно о ICMP и его возможностях рассказано в главе 4, “Протокол ICMP”.

Служба DNS

Назвать что-то еще не означает понять, чем это является на самом деле, но часто правильное название — половина успеха. Я помню, когда впервые услышал о DNS (Domain Name System — система доменных имен). В то время основные поставщики программного обеспечения для работы с базами данных анонсировали

свои продукты для работы с распределенными базами данных, и вскоре я понял, что имею дело именно с таким продуктом. DNS — это распределенная база данных, так как единая таблица адресов не хранится на каком-то одном хосте; информация распределена по многим серверам.

Когда-то все IP-адреса и имена компьютеров хранились в таблицах, информация которых обновлялась по ночам. С увеличением глобальной сети такой подход стал непрактичным по целому ряду причин, среди которых можно выделить размер таблиц и проблему надежности работы при сбое только в одном сервере. Рассмотрим фрагмент файла `/etc/hosts` сервера UNIX.

```
127.0.0.1       loopback
127.20.1.41....relay relay.sans.org
172.20.31.19    goo goo.sans.org
```

На хостах UNIX и Windows 2000 по-прежнему используются небольшие локальные файлы `hosts`. В этих файлах хранится информация о IP-адресах и именах локальных хостов и хостов, вызов которых осуществляется чаще всего. Информация этих файлов была дополнена возможностями DNS. Конфигурация большинства серверов UNIX и Windows 2000 предусматривает поиск адреса интересующего хоста, начиная с файла `/etc/hosts`, и только если этот поиск будет неудачным, происходит обращение к службе DNS. Таким образом, бо́льшая часть обязанностей системных администраторов теперь перекладывается на администраторов DNS-серверов.

Перед изучением самой службы DNS следует разобраться в понятии доменов DNS. *Домен* представляет собой логический элемент базы данных DNS. Первоначально было только семь "общих" доменов первого уровня `.com`, `.org`, `.edu`, `.net`, и менее известные `.int`, `.gov` и `.mil`. Не так давно к ним были добавлены `.aero`, `.biz`, `.coop`, `.info`, `.museum`, `.name` и `.pro`. Существуют и двухбуквенные домены первого уровня, используемые для обозначения принадлежности определенному государству (например, `.us`, `.fr` и `.uk` для США, Франции и Великобритании соответственно). Эти домены состоят из более мелких доменов, к которым ежедневно обращаются сотни пользователей (например, `yahoo.com` и `sans.org`). И каждый из этих доменов является фрагментом единой системы DNS.

Теперь давайте разберемся, как в DNS осуществляется преобразование символьного имени хоста в IP-адрес и наоборот. Если не вдаваться в детали, то вообще для решения этой задачи предназначены две команды: `gethostbyaddr` и `gethostbyname`. Это преобразование необходимо, так как пользователи вызывают удаленный компьютер по его символьному имени, а компьютер может взаимодействовать с другим компьютером, только обладая его IP-адресом. Да и в самой IP-дейтаграмме нет поля для хранения имен хостов, а только для их IP-адресов.

Используя вызов `gethostbyaddr`, хост отправляет IP-адрес DNS-серверу. DNS-сервер должен найти соответствующее имя хоста в базе данных и вернуть его источнику запроса. Этот процесс только на первый взгляд кажется элементарным (более подробно он рассматривается в главе 6, "DNS"). Для обратного преобразования имени хоста в IP-адрес DNS-серверу отправляется запрос `gethostbyname`. Это весьма упрощенная схема работы службы DNS, приведенная здесь только для начального ознакомления с технологией.

Маршрутизация

Вы еще не забыли о том, что стек TCP/IP состоит из четырех уровней: приложений, транспортного, сетевого и канального?

Мы прервали наш рассказ на сетевом уровне. Сетевой уровень неразрывно связан с понятием маршрутизации, то есть метода доставки сообщений с одного хоста на другой независимо от физической схемы линий связи. Возможно, этот уровень стоило назвать "IP-уровень", так как именно здесь для маршрутизации используются IP-адреса. Важно помнить, что сам протокол IP не зависит от физической среды передачи данных.

Мы уже рассказали о методе доставки трафика хосту сети с тем же идентификатором сети и маской подсети, что и хост-отправитель. Для отправки широковещательного запроса всем компьютерам локальной сети используется протокол ARP. На этот запрос отвечает только один компьютер — тот, который обладает указанным IP-адресом. В ответе содержится MAC-адрес этого хоста. Как же тогда трафик доставляется в другие сети, если широковещательные ARP-запросы возможны только в локальных сетях? Для этой цели и существует маршрутизация.

На каждом хосте хранится таблица маршрутизации, в которой указан адрес маршрутизатора, используемого по умолчанию. Если интересующий нас хост находится за пределами локальной сети, трафик должен быть отправлен этому маршрутизатору по умолчанию. Маршрутизатор отвечает за передачу трафика на один переход ближе к адресу назначения. Конечной точкой этого перехода может быть как другой маршрутизатор, так и хост-получатель, если он работает в сети, непосредственно подключенной к интерфейсу маршрутизатора. Возникает вопрос: как маршрутизатор узнает, куда следует передать данные, и как он обновляет информацию о доступных маршрутах? Ведь это должен быть постоянный процесс, связанный с возможным отключением одних сетей (например, из-за неисправностей) и подключением других.

Для обновления таблиц маршрутизации, хранящихся на каждом маршрутизаторе, используются специальные протоколы динамической маршрутизации.

Протоколы маршрутизации можно разделить на две категории: протоколы типа IGP (Interior Gaterway Protocol — протокол внутреннего шлюза) и протоколы типа EGP (External Gaterway Protocol — протокол внешнего шлюза). Протокол IGP предназначен для маршрутизации трафика между компьютерами *автономной системы* (Autonomous System — AS). Автономной системой называют совокупность маршрутизаторов, объединенных по иерархическому принципу, и находящихся под единым управлением. Среди протоколов категории IGP широкое распространение получил протокол RIP (Routing Information Protocol — протокол маршрутной информации). Это простой протокол, который не требует значительных усилий при настройке работы, поддерживаемый практически всеми сетевыми устройствами. Протокол OSPF (Open Shortest Path First — открытый протокол маршрутизации по кратчайшему пути) — это еще один протокол категории IGP. Протоколы RIP и OSPF различаются методом получения обновлений о доступных маршрутах и критериями поиска наилучшего маршрута.

Протоколы категории EGP требуются для передачи пакетов между автономными системами. Эти протоколы позволяют связать отдельные AS в единую сеть, в которой все компьютеры могут беспрепятственно обмениваться информацией

друг с другом. Наиболее известным EGP-протоколом можно назвать протокол BGP (Border Gaterway Protocol — протокол граничного шлюза). На текущий момент протокол BGP обеспечивает обмен данными по основным магистралям Internet. BGP-серверы на магистральных линиях Internet должны поддерживать таблицы маршрутизации, в которых хранятся адреса всех внешних маршрутизаторов Internet. Это, действительно, сложная задача.

Резюме

В этой вводной главе был сконцентрирован материал по множеству самых разнообразных вопросов. Не вдаваясь в детали, мы постарались познакомить наших читателей с базовыми концепциями, необходимыми для понимания следующих глав этой книги.

Прежде всего, необходимо запомнить, что обмен информацией между двумя подключенными к сети компьютерами осуществляется посредством нескольких уровней обработки данных. На стороне отправителя сообщение передается вниз по стеку и на каждом уровне к нему добавляется определенный заголовок. На стороне получателя осуществляется обратный процесс — заголовки последовательно удаляются при обработке пакета на определенном уровне стека. Таким образом, осуществляется взаимодействие между соответствующими уровнями хоста-отправителя и хоста-получателя. Передающийся пакет на каждом уровне немного отличается, поэтому и имеет другое название.

На разных уровнях стека TCP/IP для достижения пакетом пункта назначения используются и IP-, и MAC-адреса. Номера портов позволяют обратиться к нужному приложению на удаленном хосте, например к `sendmail` или `telnet`. Протокол TCP ориентирован на установку соединений и гарантирует их надежную доставку, тогда как UDP не обеспечивает такой гарантии и считается ненадежным. Служба DNS позволяет выполнять преобразование имен хостов в их IP-адреса и наоборот. Передача дейтаграмм по указанному адресу осуществляется благодаря процессу маршрутизации. Различные аспекты работы стека протоколов TCP/IP будут подробнее рассмотрены в следующих разделах этой книги.