СОДЕРЖАНИЕ

1

АНАЛИЗ ПАКЕТОВ И ОСНОВЫ ОРГАНИЗАЦИИ СЕТЕЙ

Каждый день в вычислительной сети может произойти миллион самых разных событий: от простого заражения шпионской программой до сложной ошибки конфигурирования маршрутизатора. И самое лучшее, на что можно надеяться, — быть готовым во всеоружии знаний и инструментальных средств отреагировать на осложнения подобного рода.

Чтобы действительно разобраться в затруднениях, возникающих в сети, необходимо перейти на уровень пакетов. Все затруднения в сети начинаются именно на этом уровне, где могут обнаружиться скверные реализации даже самых опрятно выглядящих приложений, а заслуживающие, на первый взгляд, полного доверия сетевые протоколы оказаться зловредными. Здесь от нас ничего не скроется, поскольку на этом уровне нет ни вводящей в заблуждение структуры меню, ни привлекательной графики, ни неблагонадежных работников и вообще никакой секретной информации, кроме зашифрованной. И чем больше нам удается сделать на уровне пакетов, тем лучше мы можем контролировать свою сеть и разрешать возникающие в ней затруднения. Это и есть область действия анализа пакетов.

Рассмотрению именно этой области и посвящена данная книга. Из реальных сценариев вам предстоит узнать, как бороться с медленной передачей данных по сети, выявлять узкие места в приложениях и даже отслеживать

хакеров. Прочитав эту книгу, вы сможете реализовать методики анализа пакетов, которые помогут вам разрешать даже самые сложные затруднения, возникающие в вашей сети.

В этой главе представлены самые основы с акцентом на передачу данных по сети. Материал этой главы поможет вам получить в свое распоряжение инструментальные средства, которые потребуются для дальнейшего изучения различных сценариев из реальной эксплуатации сетей.

Анализ пакетов и их анализаторы

Анализ пакетов, иногда еще называемый *анализом протоколов*, описывает процесс перехвата и интерпретации действующих данных по мере их продвижения по сети, чтобы лучше понять, что же в ней происходит. Как правило, анализ пакетов проводится *анализатором пакетов* — инструментальным средством, применяемым для перехвата первичных данных, передаваемых по проводам сети.

Анализ пакетов может оказать помощь в следующем.

- Уяснить характеристики сети.
- Выяснить, кто находится в сети.
- Определить, кто или что "съедает" доступную пропускную способность сети.
- Выявить моменты, когда использование сети достигает своего пика.
- Выявить зловредную деятельность в сети.
- Обнаружить небезопасные и громоздкие приложения.

Для анализа пакетов имеются различные программы: как бесплатные, так и коммерческие. Каждая такая программа предназначена для определенных целей. К числу самых распространенных программ анализа пакетов относятся `tcpdump`, OmniPeek и Wireshark (именно последней и уделяется основное внимание в этой книге). Программы OmniPeek и Wireshark снабжены графическим пользовательским интерфейсом, тогда как `tcpdump` является утилитой командной строки.

Оценка анализатора пакетов

Выбирая анализатор пакетов, необходимо принять во внимание целый ряд факторов, включая следующие.

- **Поддержка сетевых протоколов.** Все анализаторы пакетов способны интерпретировать различные протоколы, а большинство из них — наиболее распространенные сетевые протоколы (например, IPv4 и ICMP),

транспортные протоколы (например, TCP и UDP) и даже протоколы уровня приложений (например, DNS и HTTP). Хотя они могут и не поддерживать нетрадиционные и более сложные протоколы (например, IPv6, SMBv2 и SIP). Поэтому, выбирая анализатор пакетов, убедитесь, что в нем поддерживаются применяемые вами протоколы.

- **Удобство использования.** Обращайте особое внимание на интерфейс анализатора пакетов, простоту его установки и общую последовательность операций. Выбранная вами программа должна соответствовать уровню вашей квалификации. Так, если у вас имеется весьма скромный опыт анализа пакетов, вам вряд ли стоит выбирать такие сложные анализаторы пакетов, действующие в режиме командной строки, как утилита `tcpdump`. А если вы имеете немалый опыт анализа пакетов, то вам подойдет и более развитая, хотя и сложная программа. По мере приобретения необходимого опыта вы можете даже найти полезным сочетать в отдельных случаях несколько программ анализа пакетов.
- **Стоимость.** Самое замечательное, что многие анализаторы пакетов бесплатны и практически ничем не уступают их коммерческим аналогам. А самое главное отличие бесплатных анализаторов пакетов от коммерческих заключается в их механизмах отчетности. В состав коммерческих продуктов, как правило, входит специальный модуль формирования отчетов, тогда как в бесплатных приложениях такие средства отсутствуют, и поэтому они обеспечивают лишь ограниченную отчетность.
- **Поддержка программ.** Даже если вы овладели основами работы с программой анализа пакетов, вам иногда потребуется дополнительная поддержка для решения новых задач по мере их появления. Оценивая имеющуюся поддержку программ, обращайте внимание на документацию для разработчиков, публичные форумы и списки рассылки пользователей. И несмотря на возможную нехватку формализованной коммерческой поддержки бесплатных программ анализа пакетов вроде Wireshark, сообщество пользователей и участников их разработки нередко ведет активные дискуссионные клубы, вики-страницы и блоги, чтобы помочь извлечь наибольшую пользу из выбранного вами анализатора пакетов.
- **Доступ к исходному коду.** Некоторые анализаторы пакетов относятся к программному обеспечению с открытым исходным кодом. Это дает возможность просматривать исходный код программы, а иногда и вносить в него необходимые коррективы. Если у вас особый или сложный случай для применения анализатора пакетов, то такая возможность окажется для вас весьма привлекательной. Исходный код большинства коммерческих анализаторов пакетов недоступен.

- **Поддержка операционной системы.** К сожалению, не во всех анализаторах пакетов поддерживается каждая операционная система. Если вы работаете консультантом, у вас может возникнуть потребность перехватывать и анализировать пакеты в самых разных операционных системах. Следовательно, вам потребуется инструментальное средство, работающее в большинстве операционных систем. Следует также иметь в виду, что пакеты иногда придется перехватывать на одной машине, а просматривать на другой. Отличия в операционных системах могут вынудить вас пользоваться разными приложениями на отдельных машинах.

Принцип действия анализаторов пакетов

В процессе анализа пакетов задействованы как программные, так и аппаратные средства. Этот процесс делится на следующие стадии.

1. **Сбор данных.** Прежде всего анализатор пакетов собирает первичные двоичные данные из сети. Как правило, это делается переключением избранного сетевого интерфейса в *смешанный режим (promiscuous mode).* В этом режиме сетевая плата может принимать весь трафик в сегменте сети, а не только адресуемый ей трафик.
2. **Преобразование.** Далее перехваченные двоичные данные преобразуются в удобочитаемую форму. На это способно большинство развитых анализаторов пакетов, работающих в режиме командной строки. На этой стадии сетевые данные могут автоматически интерпретироваться только на самом элементарном уровне, оставляя бо́льшую часть анализа конечному пользователю вручную.
3. **Анализ.** Наконец, анализатор пакетов проводит анализ перехваченных и преобразованных данных. В частности, он проверяет протокол перехваченных в сети данных, исходя из извлекаемой информации, и далее начинает анализ характерных особенностей этого протокола.

Установление связи между компьютерами

Чтобы полностью уяснить анализ пакетов, необходимо точно знать, каким образом устанавливается связь между компьютерами. В этом разделе мы рассмотрим основные положения о сетевых протоколах, модель OSI (Open Systems Interconnections — взаимодействие открытых систем), сетевые фреймы данных и аппаратную поддержку всего этого.

Сетевые протоколы

Современные сети состоят из разных систем, работающих на различных платформах. Для сообщения между этими системами в качестве общепонятного языка служит ряд сетевых *протоколов*. К числу самых распространенных относятся TCP (Transmission Control Protocol — протокол управления передачей), IP (Internet Protocol — межсетевой протокол), ARP (Address Resolution Protocol — протокол преобразования адресов), а также DHCP (Dynamic Host Configuration Protocol — протокол динамического конфигурирования хоста). Все эти протоколы логически сгруппированы для совместной работы в так называемый *стек протоколов*.

Проколы удобно сравнить с правилами, регулирующими употребление естественного языка. В каждом языке имеются определенные правила, например, порядок спряжения глаголов, приветствия людей и даже надлежащей благодарности кого-нибудь. Протоколы действуют сходным образом, давая возможность определить порядок маршрутизации пакетов, установления сетевого соединения и подтверждения приема данных.

Протокол может быть как очень простым, так и крайне сложным в зависимости от его функций. И хотя различные протоколы могут существенно отличаться, многие сетевые протоколы призваны решать следующие вопросы.

- **Установление соединения.** Кто инициирует установление соединения: клиент или сервер? Какой информацией следует обменяться, прежде чем устанавливать соединение?
- **Согласование характеристик соединения.** Шифруется ли передача данных по сетевому протоколу? Каким образом происходит обмен ключами шифрования между связывающимися хостами (т.е. сетевыми узлами)?
- **Форматирование данных.** Каким образом организованы данные, содержащиеся в пакете? В каком порядке данные обрабатываются принимающим их устройством?
- **Обнаружение и исправление ошибок.** Что происходит в том случае, если пакет слишком долго достигает места своего назначения? Каким образом клиент возобновляет свою работу, если он не в состоянии установить соединение с сервером в кратчайший период времени?
- **Разрыв соединения.** Каким образом один хост дает знать другому, что связь окончена? Какую завершающую информацию следует передать, чтобы корректно разорвать соединение?

Семиуровневая модель OSI

Сетевые протоколы подразделяются по своим функциям на основании принятой в данной отрасли стандартной модели OSI. Эта иерархическая модель состоит из семи уровней и очень удобна для понимания особенностей связи и передачи данных по сети. На рис. 1.1, *справа*, показаны уровни модели OSI, а соответствующая терминология для обозначения данных на каждом из этих уровней — *слева*. На самом верху данной модели находится уровень приложений, представляющий программы, предназначенные для доступа к сетевым ресурсам, а в самом низу — физический уровень, где, по существу, данные перемещаются по сети. Протоколы на каждом уровне действуют совместно, обеспечивая надлежащую обработку данных на уровнях, расположенных непосредственно выше и ниже.

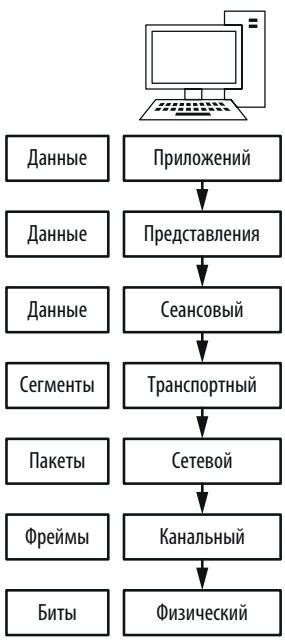

Рис. 1.1. Иерархическое представление семи уровней модели OSI

Каждый уровень модели OSI выполняет конкретную функцию, как поясняется ниже.

- **Уровень приложений (седьмой).** На этом самом верхнем уровне модели OSI предоставляются средства для доступа пользователей к сетевым ресурсам. Как правило, это единственный уровень, доступный конечным пользователям, поскольку на нем предоставляется интерфейс, на основании которого они осуществляют всю свою деятельность в сети.

ПРИМЕЧАНИЕ *Первоначально модель OSI была обнародована в 1983 году Международной организацией по стандартизации (ISO) в виде документа под названием ISO 7498. Модель OSI является всего лишь рекомендуемым в данной отрасли стандартом. Это означает, разработчики сетевых протоколов не должны строго придерживаться данной модели. На самом деле модель OSI не является единственной для организации сетей. Например, некоторые разработчики отдают предпочтение модели DoD, т.е. модели Министерства обороны США, иначе называемой моделью TCP/IP.*

- **Уровень представления данных (шестой).** На этом уровне получаемые данные преобразуются в формат, удобный для их чтения на уровне приложений. Порядок кодирования и декодирования данных на этом уровне зависит от протокола, применяемого на уровне приложений для передачи и приема данных. На уровне приложений может также использоваться несколько форм шифрования и дешифрования данных для их защиты.
- **Сеансовый уровень (пятый).** На этом уровне происходит *диалог*, или *сеанс связи*, между двумя компьютерами. Сеансовый уровень отвечает также за установление дуплексного (т.е. двунаправленного) или полудуплексного (т.е. однонаправленного) соединения, а также для корректного (т.е. не резкого и внезапного) разрыва связи между двумя хостами (т.е. сетевыми узлами).
- **Транспортный уровень (четвертый).** Основное назначение транспортного уровня — предоставить надежные транспортные услуги нижележащим уровням. Благодаря управлению потоком данных, их сегментации и десегментации, исправлению ошибок на транспортном уровне обеспечивается безошибочная доставка данных из одной точки сети в другую. Обеспечить надежную доставку данных крайне сложно, поэтому в модели OSI для этой цели выделен отдельный уровень. На транспортном уровне используются протоколы как с установлением соединения, так и без него. Именно на этом уровне и действуют определенные брандмауэры и промежуточные, так называемые прокси-серверы.
- **Сетевой уровень (третий).** Один из самых сложных уровней модели OSI, обеспечивающий маршрутизацию данных между физическими сетями и правильную адресацию сетевых узлов (например, по IP-адресу). На этом уровне происходит также разбиение потоков данных на более мелкие части, а иногда и обнаружение ошибок. Именно на этом уровне и действуют маршрутизаторы.
- **Канальный уровень (второй).** На этом уровне предоставляются средства для переноса данных по физической сети. Основное назначение данного уровня — предоставить схему адресации для обозначения

физических устройств (например, MAC-адреса). Именно на этом уровне и действуют такие физические устройства, как мосты и коммутаторы.

- **Физический уровень (первый).** Это самый нижний уровень модели OSI, где находится среда, по которой переносятся сетевые данные. На этом уровне определяется физические и электрические характеристики всего сетевого оборудования, включая уровни напряжений в сети, концентраторы, сетевые адаптеры, повторители и кабельную разводку. На физическом уровне устанавливаются и разрываются сетевые соединения, предоставляются средства для совместного использования общих сетевых ресурсов и преобразования сигналов из цифровой в аналоговую форму, и наоборот.

ПРИМЕЧАНИЕ *Для удобства запоминания уровней модели OSI на английском языке служит мнемоническая фраза "Please Do Not Throw Sausage Pizza Away" (Пожалуйста, не выбрасывайте пиццу с колбасой), где прописные буквы отдельных слов обозначают названия каждого уровня данной модели, начиная с первого*[1].

В табл. 1.1 перечислены некоторые из наиболее распространенных протоколов, употребляемых на каждом уровне модели OSI.

Таблица 1.1. Типичные протоколы, используемые на каждом уровне модели OSI

Уровень	Протоколы
Приложений	HTTP, SMTP, FTP, Telnet
Представления	ASCII, MPEG, JPEG, MIDI
Сеансовый	NetBIOS, SAP, SDP, NWLink
Транспортный	TCP, UDP, SPX
Сетевой	IP, IPX
Канальный	Ethernet, Token Ring, FDDI, AppleTalk
Физический	Проводной или беспроводный

Несмотря на то что модель OSI является всего лишь рекомендуемым стандартом, ее следует знать наизусть, поскольку она предоставляет удобный словарь терминов для осмысления и описания затруднений, возникающих в сетях. По мере чтения книги вы обнаружите, что вопросы маршрутизации относятся к третьему уровню данной модели, а вопросы программирования — к седьмому.

[1] В качестве варианта на русском языке предлагается мнемоническая фраза "Федя, Как Стать Тебе Самым Первым Парнем", естественно, на деревне. — *Примеч. ред.*

ПРИМЕЧАНИЕ *Однажды коллега рассказал мне о жалобе одного пользователя на невозможность доступа к сетевому ресурсу. Это затруднение возникло из-за того, что пользователь ввел неверный пароль. Мой коллега отнес данное затруднение к восьмому, неофициальному пользовательскому уровню модели OSI. И такое обозначение часто употребляется в среде тех, кто работает на уровне пакетов.*

Прохождение данных по модели OSI

Исходно передача данных по сети начинается на уровне приложений передающей системы. Данные проходят сверху вниз по всем уровням модели OSI до тех пор, пока не достигнут физического уровня, где находится точка, откуда данные отправляются из передающей системы в принимающую. А принимающая система получает данные на своем физическом уровне, откуда данные проходят снизу вверх по всем уровням модели OSI, достигая в конечном итоге уровня приложений.

Каждый уровень модели OSI может взаимодействовать только с уровнями, расположенными непосредственно выше и ниже его. Например, на уровне 2 данные можно передавать и принимать только с уровней 1 и 3.

Ни одна из услуг, предоставляемых различными протоколами на любом заданном уровне модели OSI, не должна быть избыточной. Так, если на одном уровне протокол предоставляет конкретную услугу, то ни один другой протокол на любом другом уровне не должен предоставлять такую же самую услугу. У протоколов на разных уровнях могут быть средства, служащие сходным целям, но их функционирование будет хотя бы немного, но все равно отличаться.

Протоколы на соответствующих уровнях передающего и приемного устройств дополняют друг друга. Так, если протокол на седьмом уровне передающего устройства отвечает за форматирование передаваемых данных, то соответствующий протокол на седьмом уровне приемного устройства должен отвечать за чтение полученных отформатированных данных.

На рис. 1.2 показано графическое представление модели OSI на двух связанных вместе устройствах. Как видите, передача данных происходит сверху вниз на одном устройстве, а прием — в обратном порядке на другом устройстве.

Инкапсуляция данных

Протоколы, действующие на разных уровнях модели OSI, обмениваются данными посредством *инкапсуляции данных*. Каждый уровень данной модели отвечает за добавление в начало и конец передаваемых данных соответствующего заголовка и концевика в виде дополнительных битов информации, предназначенных для взаимодействия между уровнями. Когда, например, данные получаются на транспортном уровне из сеансового уровня, на транспортном

уровне добавляется свой заголовок, содержащий требуемую информацию, в эти данные, перед передачей их дальше на сетевой уровень.

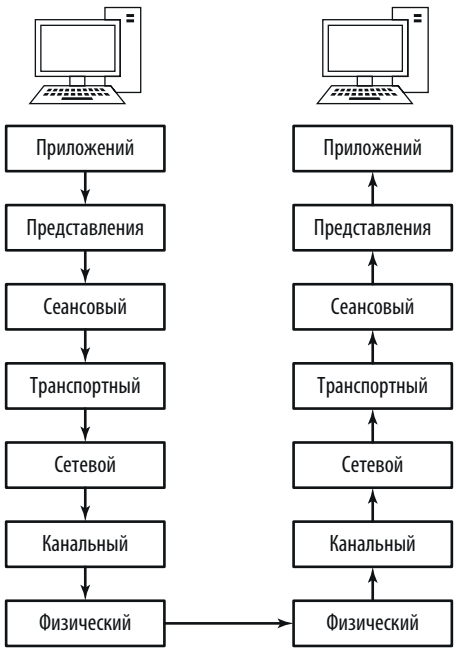

Рис. 1.2. Протоколы, действующие на одном и том же уровне как в передающей, так и в принимающей системе

В процессе инкапсуляции создается блок протокольных данных (PDU — protocol data unit), включающий в себя передаваемые данные и всю добавленную заголовочную и концевую информацию. По мере продвижения данных сверху вниз по модели OSI и добавления в нее заголовочной и концевой информации на разных уровнях блок PDU изменяется, постепенно разрастаясь. В окончательном виде блок PDU формируется, когда он достигает физического уровня, где он посылается приемному устройству по месту назначения. А на приемном устройстве заголовки и концевики извлекаются соответствующими протоколами из блока PDU в обратном их вводу порядке по мере прохождения данных снизу вверх по уровням модели OSI. И как только блок PDU достигнет верхнего уровня модели OSI, останутся лишь данные из исходного уровня приложений.

Чтобы продемонстрировать принцип действия инкапсуляции данных, рассмотрим упрощенный практический пример создания, передачи и приема пакета по отношению к модели OSI. Однако специалисты по анализу пакетов редко упоминают о сеансовом или уровне представлений модели OSI. Именно поэтому оба эти уровня отсутствуют в рассматриваемом здесь примере, как, впрочем, и в остальных примерах из данной книги.

ПРИМЕЧАНИЕ *Для описания данных, упакованных на каждом уровне, в модели OSI употребляются специальные термины. Так, на физическом уровне данные содержатся в виде отдельных битов, на канальном – в виде фреймов, на сетевом – в виде пакетов, на транспортном – в виде сегментов, а на трех верхних уровнях – просто в виде данных. Но на практике такая терминология не находит особого применения, и поэтому далее в книге мы будем пользоваться только термином пакет для обозначения полного или частичного блока PDU, включая заголовочную и концевую информацию из нескольких или всех уровней модели OSI.*

В данном случае предпринимается попытка просмотреть веб-страницу по адресу `http://www.google.com/`. Прежде всего необходимо сформировать пакет запроса, передаваемый из исходного клиентского компьютера на целевой серверный компьютер. В данном случае предполагается, что сеанс связи по сетевому протоколу TCP/IP уже установлен. Процесс инкапсуляции данных в рассматриваемом здесь примере наглядно показан на рис. 1.3.

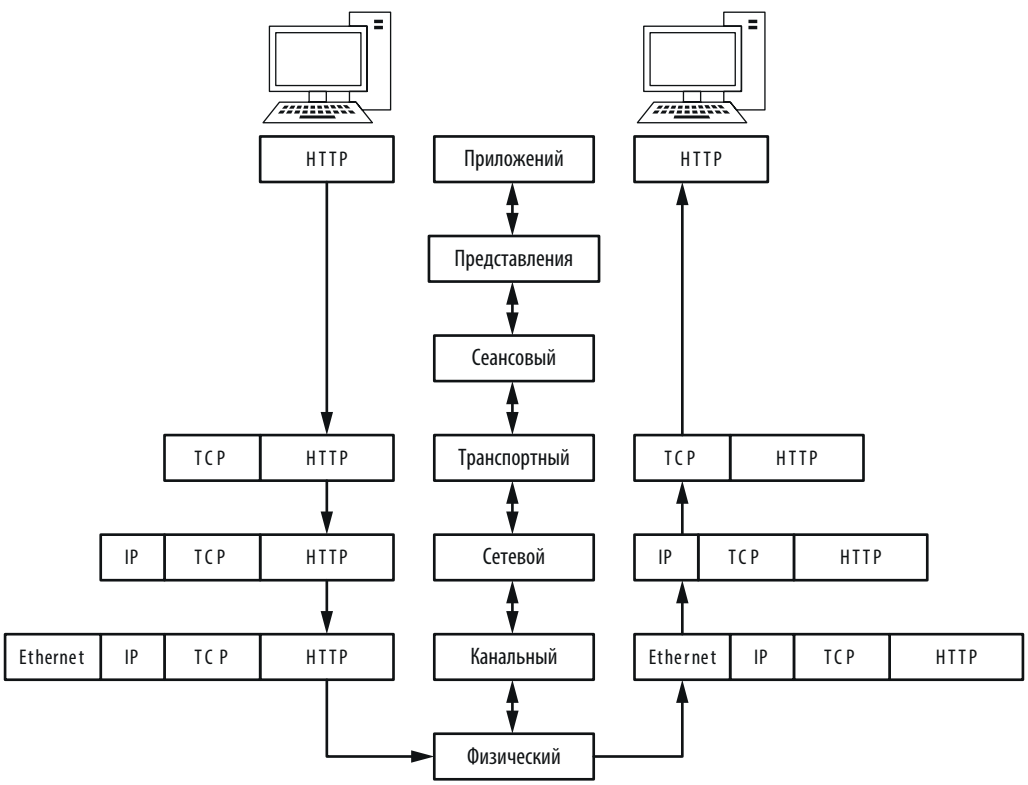

Рис. 1.3. Графическое представление инкапсуляции данных при их обмене между клиентом и сервером

Итак, начнем с клиентского компьютера на уровне приложений. Здесь речь идет о просмотре страницы веб-сайта, и поэтому применяется сетевой протокол HTTP. По этому протоколу выдается команда на загрузку файла `index.html` со страницы по указанному выше адресу.

ПРИМЕЧАНИЕ *На практике браузер запросит сначала у веб-сайта корневой каталог документов, обозначаемый косой чертой (/). Как только сервер получит этот запрос, он переадресует браузер к тому файлу, который сконфигурирован для обслуживания запросов на выдачу корневого каталога документов. Обычно это файл вроде* **`index.html`** *или* **`index.php`**. *Подробнее об этом речь пойдет в главе 9, "Распространенные протоколы верхнего уровня", при обсуждении сетевого протокола HTTP.*

Как только из протокола уровня приложений будет отправлена команда, останется лишь доставить пакет по месту его назначения. Данные из пакета передаются сверху вниз на транспортный уровень. В протоколе HTTP уровня приложений применяется протокол TCP, так как первый располагается в стеке протоколов *над* последним, а по существу, "сидит" на нем. Поэтому TCP служит в качестве протокола транспортного уровня, обеспечивая надежную доставку пакета. На транспортном уровне формируется TCP-заголовок, который добавляется в блок PDU, как показано на рис. 1.3. В этот TCP-заголовок входят порядковые номера и прочие данные, добавляемые к пакету и обеспечивающие надлежащую его доставку.

ПРИМЕЧАНИЕ *Специалисты по сетям нередко говорят, что один сетевой протокол "сидит" или "ездит" на другом протоколе из-за нисходящего характера архитектуры модели OSI. Так, протокол HTTP уровня приложений предоставляет определенные услуги, полагаясь на протокол TCP транспортного уровня для надежного оказания своих услуг. А услуги обоих этих протоколов опираются на протокол IP сетевого уровня для доставки данных по адресу. Таким образом, протокол HTTP "сидит" на протоколе TCP, а тот – на протоколе IP.*

Выполнив свое задание, протокол TCP передает пакет протоколу IP третьего уровня, отвечающему за логическую адресацию пакета. С этой целью в протоколе IP формируется информация о логической адресации, которая добавляется в блок PDU, после чего пакет передается протоколу Ethernet на канальном уровне, где физические адреса Ethernet сохраняются в Ethernet-заголовке. Таким образом, полностью собранный пакет переносится на физический уровень, где он передается в двоичном виде единиц и нулей по сети.

Полностью готовый пакет переносится по кабельной системе сети, достигая в конечном итоге веб-сервера компании Google. Этот веб-сервер начинает чтение полученного пакета снизу вверх, т.е. начиная с информации транспортного уровня, где содержатся сведения о физической адресации по протоколу Ethernet. Эта информация используется в сетевом адаптере с целью выяснить, предназначается ли полученный пакет для данного конкретного веб-сервера. После обработки этой информации из пакета удаляется информация второго уровня и обрабатывается информация третьего уровня.

Информация об адресации по протоколу IP третьего уровня читается с целью убедиться, что пакет адресуется правильно и не разбит на части. Эта информация также удаляется из пакета, чтобы обеспечить обработку информации следующего уровня.

Далее читается информация из протокола TCP четвертого уровня с целью убедиться, что пакет поступил в правильной последовательности. После этого информация четвертого уровня удаляется из пакета, где остаются лишь данные уровня приложений, которые могут быть переданы серверному приложению, развернутому на указанном веб-сервере. В ответ на этот пакет от клиента сервер должен передать сначала пакет подтверждения по протоколу TCP, чтобы уведомить клиента о благополучном получении его запроса, а затем запрашиваемый файл `index.html`.

Все пакеты создаются и обрабатываются описанным в данном примере способом независимо от применяемых протоколов. Но в то же время далеко не каждый пакет в сети формируется, начиная с протокола уровня приложений. В сети нередко можно обнаружить пакеты, содержащие информацию только из протоколов второго, третьего или четвертого уровня.

Сетевое оборудование

А теперь рассмотрим сетевое оборудование, где выполняется вся черновая работа по обмену данными в сети. В этом разделе основное внимание уделяется наиболее характерным компонентам сетевого оборудования, к которым относятся концентраторы, коммутаторы и маршрутизаторы.

Концентраторы

Обычный концентратор представляет собой прямоугольный корпус с целым рядом портов типа RJ-45 подобно модели NETGEAR, приведенной на рис. 1.4. Концентраторы разнятся от совсем небольших устройств на 4 порта до крупных устройств на 48 портов в стоечном исполнении для корпоративной среды.

Рис. 1.4. Типичный концентратор на четыре порта для сети Ethernet

Концентраторы могут формировать немало излишнего сетевого трафика и способны работать только в *полудуплексном режиме*, т.е. они не в состоянии одновременно передавать и принимать данные. Поэтому концентраторы, как правило, не применяются в большинстве современных сетей, а также в сетях с высоким трафиком, где вместо них используются коммутаторы, рассматриваемые далее. Тем не менее принцип действия концентраторов следует знать, поскольку им принадлежит важная роль в методике анализа пакетов, называемой "перехватом пакетов через концентратор" и обсуждаемой в главе 2, "Подключение к сети".

Концентратор — это всего лишь *повторитель*, работающий на физическом уровне модели OSI. Он принимает пакеты, посылаемые из одного порта, и передает их во все остальные порты, т.е. повторяет их, а обязанность приемного устройства — принять или отвергнуть каждый пакет. Так, если из компьютера, подключенного к порту 1 концентратора на 4 порта, требуется передать данные на компьютер, подключенный к порту 2, концентратор направит эти пакеты в порты 2, 3 и 4. Клиенты, подключенные в портам 3 и 4, проверят поле адреса получателя по стандарту Media Access Control (MAC — управление доступом к среде передачи данных), или просто MAC-адреса в Ethernet-заголовке пакета и обнаружат, что данный пакет предназначен не для них, и поэтому пропустят (т.е. отвергнут) его. На рис. 1.5 приведен пример, где компьютер А передает данные компьютеру Б. Когда компьютер А посылает эти данные, их получают все компьютеры, подключенные к концентратору. Но лишь компьютер Б фактически принимает отправленные данные, тогда как остальные компьютеры отвергают их.

В качестве аналогии допустим, что сообщение на тему "Внимание всем сотрудникам отдела маркетинга" рассылается по электронной почте всем сотрудникам в компании, а не только тем, кто работает в отделе маркетинга. Увидев данное сообщение, сотрудники этого отдела откроют его, а остальные сотрудники компании проигнорируют его, поскольку оно их не касается. Этот пример наглядно показывает, что такой подход к передаче данных по сети приводит к появлению излишнего трафика и напрасной трате времени. Но именно таким образом и действует концентратор. Наилучшим вариантом

замены концентраторов в производственных и сетях с высоким трафиком являются *коммутаторы — дуплексные* устройства, способные синхронно передавать и принимать данные.

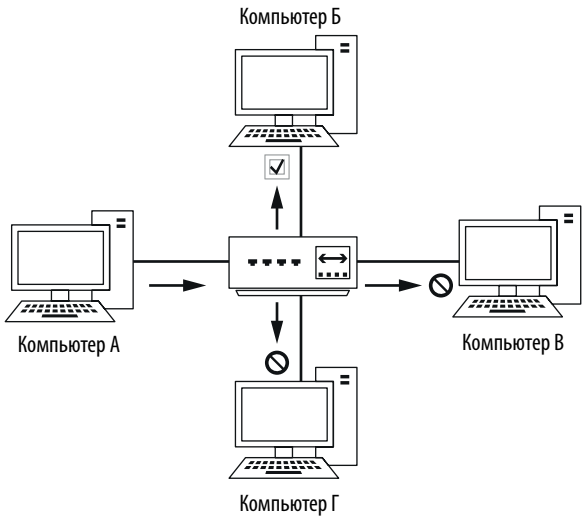

Рис. 1.5. Порядок прохождения трафика, когда компьютер А передает данные компьютеру Б через концентратор

Коммутаторы

Как и концентратор, коммутатор предназначен для повторения пакетов. Но в отличие от концентратора, коммутатор не рассылает данные в каждый порт, а посылает их только тому компьютеру, для которого они предназначены. Коммутаторы очень похожи на концентраторы, как показано на рис. 1.6.

Рис. 1.6. Коммутатор на 48 портов в стоечном исполнении для сети Ethernet

Некоторые крупные коммутаторы (например, компании Cisco) работают под управлением специализированного программного обеспечения или веб-интерфейсов, разработанных изготовителем оборудования. Такие коммутаторы обычно называются *управляемыми* и предоставляют ряд полезных возможностей для управления сетью, включая активизацию и дезактивизацию

портов, просмотр статистики портов, внесение корректив в конфигурацию и удаленную перезагрузку.

Коммутаторы предоставляют дополнительные функциональные возможности для обработки передаваемых пакетов. Для установления непосредственной связи с конкретными устройствами коммутаторы должны однозначно распознавать устройства по их MAC-адресам. Это означает, что они должны действовать на канальном уровне модели OSI.

Коммутаторы хранят адреса второго уровня каждого подключенного к ним устройства в *таблице ассоциативной памяти* (CAM –Content Addressable Memory), которая выполняет роль регулировщика дорожного движения. Когда передается пакет, коммутатор читает информацию из заголовка второго уровня в этом пакете и, используя таблицу ассоциативной памяти как справочник, определяет, в какие порты следует направить данный пакет. Коммутаторы посылают пакеты только в конкретные порты, тем самым значительно сокращая сетевой трафик.

На рис. 1.7 наглядно показан порядок прохождения сетевого трафика через коммутатор. В частности, компьютер А посылает данные только компьютеру Б как назначенному их получателю. Одновременно в сети может вестись множество диалогов, но данные передаются непосредственно между коммутатором и назначенным получателем, а не между коммутатором и всеми подключенными к нему компьютерами.

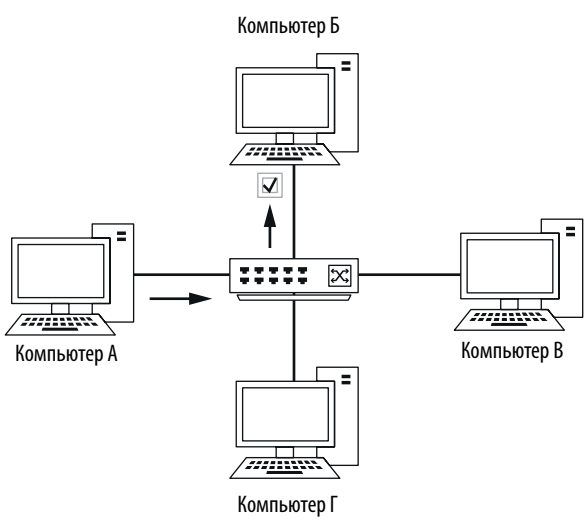

Рис. 1.7. Порядок прохождения трафика, когда компьютер А передает данные компьютеру Б через коммутатор

Маршрутизаторы

Маршрутизатор — это усовершенствованное сетевое устройство с намного более высоким уровнем функционирования, чем коммутатор или концентратор. Маршрутизатор может принимать самые разные виды и формы, но, как правило, на его передней панели установлено несколько светодиодных индикаторов, а на задней панели — ряд сетевых портов в зависимости от масштабов сети. Пример небольшого маршрутизатора приведен на рис. 1.8.

Рис. 1.8. Низкоуровневый маршрутизатор компании Enterasys, пригодный для применения в сетях малых и средних масштабов

Маршрутизаторы действуют на третьем уровне модели OSI, где они отвечают за пересылку пакетов между двумя или несколькими сетями. Процесс, в ходе которого маршрутизаторы направляют сетевой трафик между сетями, называется *маршрутизацией*. Имеется несколько видов протоколов маршрутизации, определяющих порядок, в котором разнотипные пакеты направляются в другие сети. Обычно маршрутизаторы пользуются адресами третьего уровня (например, IP-адресами) для однозначного распознавания устройств в сети.

Чтобы наглядно продемонстрировать понятие маршрутизации, обратимся к аналогии соседства городских улиц, представив дома с их адресами в качестве компьютеров, а каждую улицу — в качестве сетевого сегмента (рис. 1.9). Выйдя из своего дома, можно легко посетить своих соседей в других домах на той же самой улице, пройдя прямо по тротуару от парадной двери одного дома к другой. Аналогично коммутатор обеспечивает связь между всеми компьютерами в одном сегменте сети.

Но общение с соседями, проживающими на другой улице, подобно связи с компьютером в другом сегменте сети. Глядя на рис. 1.9, допустим, что вы проживаете на улице Виноградной, 502, и вам требуется добраться до дома по адресу Кизиловая аллея, 206. Чтобы сделать это, вы должны свернуть сначала на Дубравную улицу, а затем — на Кизиловую аллею. Это можно сравнить с пересечением разных сегментов сети. Так, если устройству, находящемуся

по адресу **192.168.0.3**, требуется связаться с устройством, расположенным по адресу **192.168.0.54**, оно должно сделать это сначала через свой маршрутизатор, чтобы достичь сети по адресу **10.100.1.x**, а затем через маршрутизатор целевого сегмента сети, прежде чем достичь этого сегмента.

Размеры и количество маршрутизаторов в сети, как правило, зависят от ее масштабов и функций. На границе личных, домашних и небольших учрежденческих сетей может быть установлен лишь один небольшой маршрутизатор. А в крупных корпоративных сетях может потребоваться несколько маршрутизаторов, установленных в разных отделах организации и подключенных к одному центральному маршрутизатору или коммутатору третьего уровня (т.е. к усовершенствованному типу коммутатора со встроенными возможностями функционирования в качестве маршрутизатора).

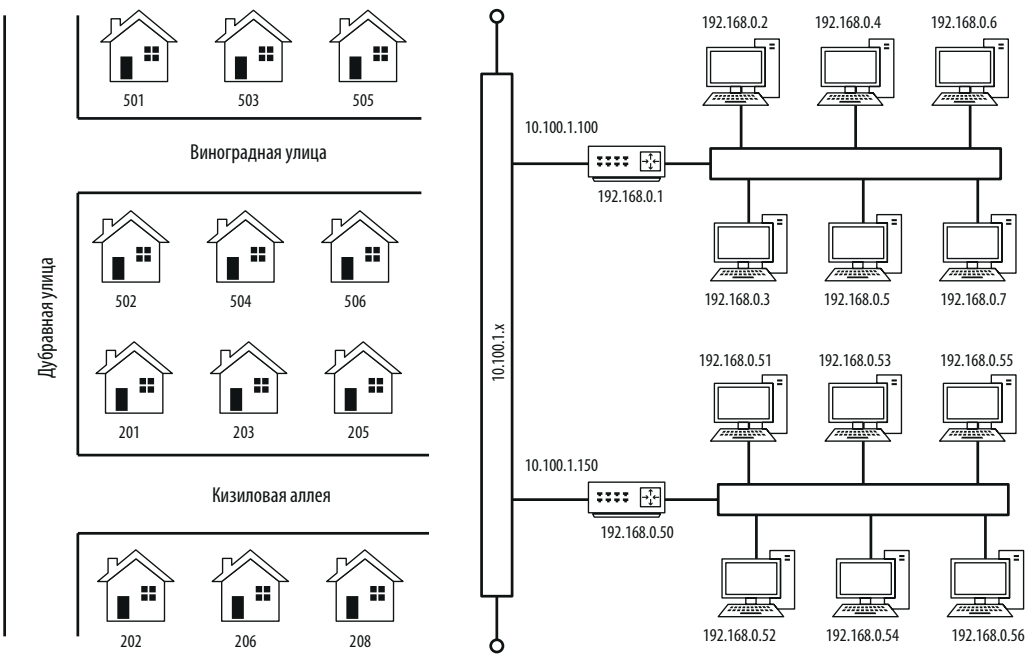

Рис. 1.9. Сравнение маршрутизируемой сети с соседством городских улиц

Рассматривая приведенные далее схемы сетей, вы в конечном итоге поймете, каким образом данные проходят через различные точки сети. Так, на рис. 1.10 приведена весьма распространенная форма компоновки маршрутизируемой сети. В данном случае две отдельные сети соединены через один маршрутизатор. Если компьютеру в сети А потребуется связаться с компьютером в сети Б, то передаваемые из него данные должны непременно пройти через маршрутизатор.

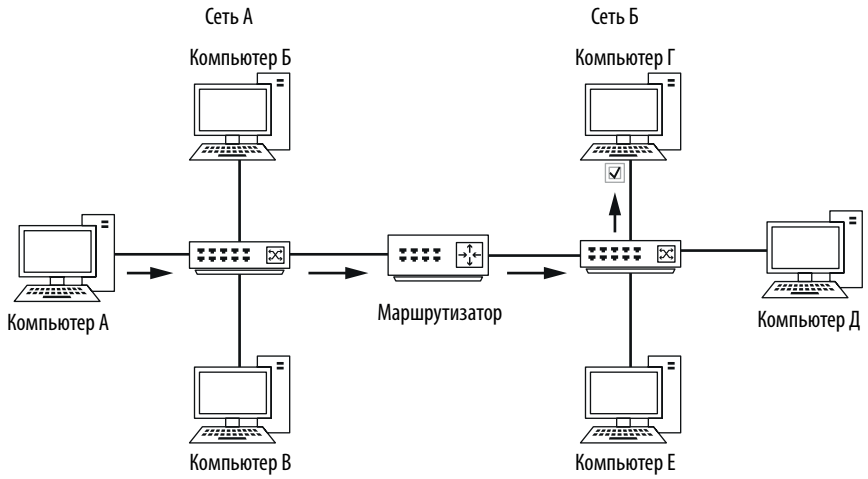

Рис. 1.10. Порядок прохождения сетевого трафика, когда компьютер А в одной сети передает данные компьютеру Г в другой сети через маршрутизатор

Классификация сетевого трафика

Сетевой трафик можно классифицировать по трем типам: широковещательный, много- и одноадресатный. Каждая из этих классификаций сетевого трафика имеет свои особенности, определяющие порядок обработки пакетов в сетевом оборудовании.

Широковещательный трафик

Широковещательным (broadcast) называется такой пакет, который посылается во все порты в сетевом сегменте независимо от того, где установлен данный порт: на концентраторе или коммутаторе. Имеются формы широковещательного трафика второго и третьего уровня. На втором уровне MAC-адрес `ff:ff:ff:ff:ff:ff` является зарезервированным широковещательным адресом, и любой трафик, направляемый по этому адресу, распространяется по всему сетевому сегменту. Широковещательный адрес имеется и на третьем уровне, но он меняется в зависимости от используемого диапазона сетевых адресов.

Самый большой из всех IP-адресов, допустимых в диапазоне адресов IP-сети, зарезервирован для применения в качестве широковещательного адреса. Так, если у компьютера имеется адрес `192.168.0.20` и маска подсети `255.255.255.0`, адрес `192.168.0.255` оказывается широковещательным (подробнее об IP-адресации речь пойдет в главе 7, “Протоколы сетевого уровня”).

Предел, до которого могут распространяться широковещательные пакеты, называется *широковещательным доменом*, который представляет собой сетевой сегмент, где любой компьютер может непосредственно передавать данные другому компьютеру без помощи маршрутизатора. В крупных сетях со многими концентраторами или коммутаторами, соединяемыми через разные средства связи и среды передачи данных, широковещательные пакеты, передаваемые от одного коммутатора, достигают всех портов на всех остальных коммутаторах в сети, поскольку пакеты повторяются, проходя от одного коммутатора к другому. На рис. 1.11 приведен пример двух широковещательных доменов в небольшой сети. Каждый широковещательный домен простирается до тех пор, пока он не достигнет маршрутизатора, и поэтому широковещательные пакеты циркулируют только в этом конкретном домене.

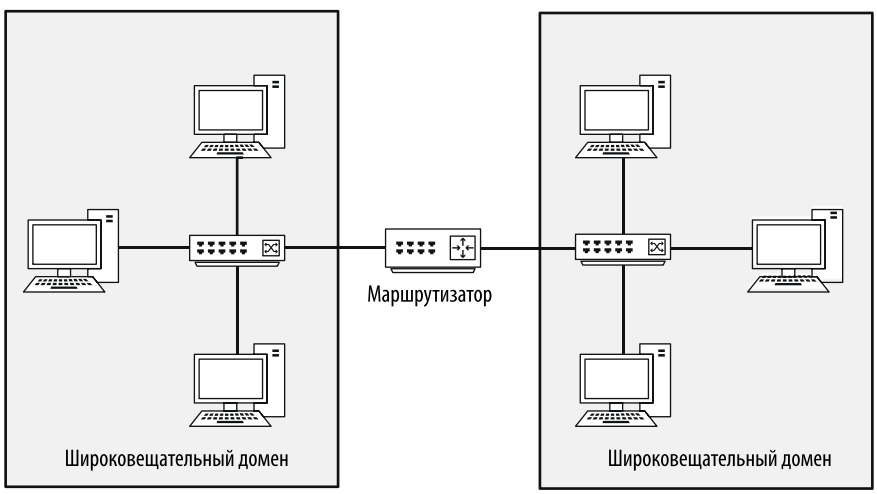

Рис. 1.11. Широковещательный домен охватывает весь текущий сегмент маршрутизируемой сети до маршрутизатора

Упоминавшаяся ранее аналогия с соседством городских улиц позволяет также наглядно показать принцип действия широковещательных доменов. В частности, широковещательный домен можно рассматривать как соседнюю улицу, где все соседи сидят на своем крыльце. Если вы станете на свое крыльцо и крикнете, то жильцы на вашей улице смогут вас услышать. Но если вам захочется пообщаться с кем-нибудь на другой улице, то придется найти какой-нибудь способ сделать это непосредственно, а не широко вещать (криком) со своего крыльца.

Многоадресатный трафик

Многоадресатной[2] *(multicast)* называется одновременная передача пакета из одного источника во многие места назначения. Цель многоадресатной передачи — использовать как можно меньше пропускной способности сети. Оптимизация многоадресатного трафика состоит в том, чтобы поменьше тиражировать поток данных по пути его следования к месту назначения. Конкретная обработка многоадресатного трафика зависит от его реализации в отдельных сетевых протоколах.

Многоадресатный трафик реализуется в основном через схему адресации, добавляющую получателей пакетов к многоадресатной группе. Именно так и действует многоадресатная передача по межсетевому протоколу IP. Такая схема адресации гарантирует от передачи пакетов тем компьютерам, для которых они не предназначены. Если вы обнаружите IP-адрес в пределах от `224.0.0.0` до `239.255.255.255`, то по нему, вероятнее всего, обрабатывается многоадресатный трафик, поскольку именно в этих пределах зарезервированы адреса для подобных целей.

Одноадресатный трафик

Одноадресатный (unicast) пакет передается непосредственно от одного компьютера к другому. Конкретное функционирование одноадресатной передачи зависит от применяемого сетевого протокола. В качестве примера можно привести устройство, которому требуется связаться с веб-сервером. Это одноточечное соединение, а следовательно, процесс передачи данных, должен быть начат клиентским устройством, передающим пакет только веб-серверу.

Заключительные соображения

В этой главе были представлены основы организации сетей, которые требуется знать для анализа пакетов. В частности, необходимо ясно понимать, что именно происходит на данном уровне передачи данных по сети, прежде чем приступить к диагностике сети. В главе 2, “Подключение к сети”, будет рассмотрен ряд методик перехвата пакетов, которые требуется проанализировать.

[2] Поскольку пакет, посланный по одному адресу, достигает многих получателей, или адресатов. — *Примеч. ред.*