

1.3. Ядро сети

Изучив периферию Интернета, давайте углубимся дальше и обратимся к ядру сети — набору коммутаторов пакетов и каналов связи, которые взаимодействуют с конечными системами Интернета. На рис. 1.10 элементы ядра сети выделены жирными линиями.

1.3.1. Коммутация пакетов

Конечные системы обмениваются друг с другом **сообщениями**, используя сетевые приложения. Сообщения могут содержать все что угодно, любую информацию, которую разработчик приложения пожелает туда поместить. Иногда они выполняют функции управления (например, сообщение «Привет» в нашем примере с рукопожатием на рис. 1.2), иногда содержат данные, например почтовое сообщение, рисунок в формате JPEG, либо аудиофайл в формате MP3. Для того чтобы отослать сообщение от конечной системы-источника в конечную систему-приемник, оно разбивается на более мелкие порции данных, называемые **пакетами**. На пути от источника к приемнику каждый пакет проходит через линии связи и **коммутаторы** (среди которых основными типами являются **маршрутизаторы** и **коммутаторы канального уровня**). Пакеты передаются по каждой линии связи с *максимальной* скоростью, которую может обеспечить данная линия. Поэтому, если исходная конечная система либо коммутатор отправляет пакет длиной L бит через соединение со скоростью R бит/с, то время передачи пакета равно L/R секунд.

Передача с промежуточным накоплением

Большинство коммутаторов пакетов используют так называемую **передачу с промежуточным накоплением**. Промежуточное накопление означает, что коммутатор пакетов должен сначала принять пакет целиком перед тем, как он начнет передавать в выходную линию связи его первый бит.

Чтобы изучить передачу с накоплением более детально, рассмотрим простую сеть, состоящую из двух конечных систем, соединенных одним маршрутизатором, как показано на рис 1.11.

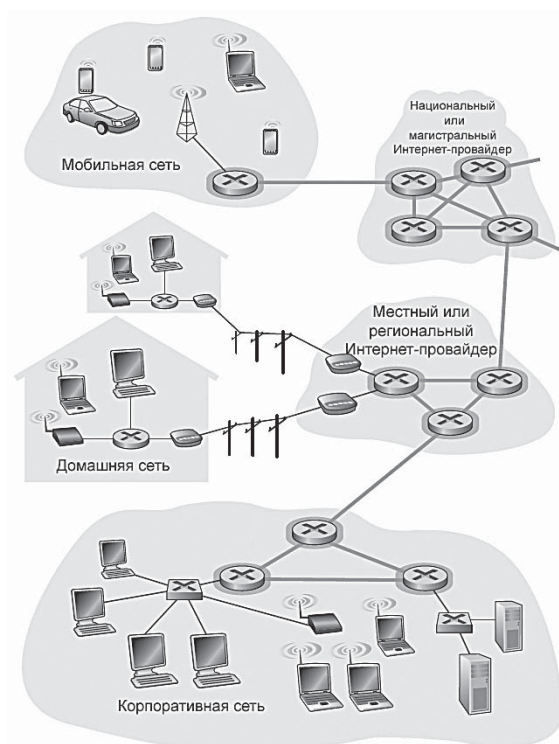


Рис. 1.10. Ядро сети



Рис. 1.11. Коммутация с промежуточным накоплением

Каждый маршрутизатор обычно имеет несколько соединений, так как его задача заключается в том, чтобы перенаправить входящий пакет в исходящее соединение; в данном примере задача маршрутизатора довольно проста: передача пакета из одной линии связи (входящей) в единственную исходящую. Здесь источник отправляет три пакета, каждый из них содержит L бит. В момент времени, показанный на рис. 1.11, источник отправил часть пакета 1, и она уже прибыла на маршрутизатор. Так как маршрутизатор работает по методу передачи с промежуточным накоплением, в данный момент он не может отправить биты, которые получил; вместо этого он должен сначала их сохранить (буферизовать). Только после того, как маршрутизатор получит все биты пакета, он может начать передачу (перенаправление) пакета в исходящую линию связи. Чтобы лучше разобраться, давайте подсчитаем количество времени, которое пройдет от того момента, когда источник начал отправку пакетов, до того момента, когда приемник получил весь пакет целиком. (Здесь мы пренебрегаем задержкой распространения — временем, которое требуется битам для прохождения по проводнику со скоростью, приблизительно равной скорости света, — что будет обсуждено в разделе 1.4). Источник начинает передачу в момент времени 0; в момент времени L/R секунд источник завершил передачу всего пакета, и пакет полностью получен и сохранен в маршрутизаторе (задержка распространения нулевая). В момент времени L/R , так как маршрутизатор только что получил весь пакет, он может начать его передачу в выходную линию связи в направлении адресата; в момент времени $2L/R$ маршрутизатор завершает передачу всего пакета, и тот полностью прибывает к месту назначения. Таким образом, общая задержка будет равна $2L/R$. Если коммутатор перенаправлял бы биты сразу же, как они прибывают, не дожидаясь получения всего пакета, общая задержка была бы равна L/R , так как не тратилось бы время на хранение битов в маршрутизаторе. Но маршрутизаторам необходимо получать, хранить и обрабатывать пакет перед тем, как его отправить.

Теперь давайте посчитаем время от того момента, когда источник начинает отсылать первый пакет, до того момента, когда приемник получит все три. Как и раньше, в момент времени L/R маршрутизатор начинает передавать первый пакет. Но в это же время L/R источник начинает отсылать второй пакет, так как первый он уже отправил целиком. Таким образом, в момент времени $2L/R$ приемник получит первый пакет, а маршрутизатор получит второй. Аналогично, в момент времени $3L/R$ приемник получит первые два пакета, а маршрутизатор получит третий. Наконец, в момент времени $4L/R$ приемник получит все четыре пакета!

Давайте теперь рассмотрим общий случай отправки одного пакета от источника к приемнику по пути, состоящему из N соединений, имеющих каждый скорость R (то есть между источником и приемником $N-1$ маршрутизатор). Применяя тот же метод, что и раньше, мы увидим, что общая (сквозная) задержка прохождения от источника к приемнику равна

$$d_{\text{сквозн}} = N \frac{L}{R} \quad (1.1)$$

Теперь вы можете сами посчитать время задержки, которое необходимо для передачи P пакетов через сеть из N соединений.

Задержки ожидания и потери пакетов

Каждый коммутатор пакетов может иметь несколько соединений. Для каждого соединения у коммутатора есть **выходной буфер** (также называемый **выходной очередью**), в котором будут храниться пакеты для отправки в данную линию связи. Выходные буферы играют ключевую роль в коммутации пакетов. Если, например, прибывающий пакет нужно отправить в линию связи, но она занята передачей другого пакета, то прибывающий пакет должен встать в очередь в выходном буфере. Таким образом, в дополнение к **задержкам накопления**, пакеты, вставая в очередь, испытывают задержки ожидания. Эти задержки являются переменными и зависят от степени перегруженности сети. Так как размер буфера маршрутизатора не бесконечен, может наступить момент, когда он полностью заполнен прибывшими пакетами, а пакеты все поступают и поступают. В таком случае происходит **потеря пакетов**. Отбрасывается либо один из прибывающих пакетов, либо один из тех, которые уже находятся в очереди.

На рис. 1.12 представлена простая сеть с коммутацией пакетов. Так же, как и на рис. 1.11, пакеты изображены в виде трехмерных плиток. Ширина плитки представляет число битов в пакете. На этом рисунке все пакеты одной ширины, то есть имеют один размер. Предположим, что хосты А и Б отсылают пакеты на хост Д. Хосты А и Б сначала отсылают пакеты по Ethernet-каналу (10 Мбит/с) на первый маршрутизатор. Маршрутизатор перенаправляет эти пакеты в канал со скоростью передачи 1,5 Мбит/с. Если в короткий промежуток времени скорость прибытия пакетов на маршрутизатор (в бит/с) превышает 1,5 Мбит/с, на нем происходит перегрузка, и пакеты встают в очередь перед тем, как их отправят в выходную линию связи. Например, если хосты А и Б каждый отправят серию из пяти пакетов в одно и то же время, то боль-

шинство из этих пакетов будут некоторое время ждать очереди. Ситуация на самом деле полностью аналогична многим каждодневным ситуациям из жизни, например, когда мы стоим в очереди перед банкоматом либо в кассе магазина. В разделе 1.4 мы подробнее рассмотрим задержку ожидания.

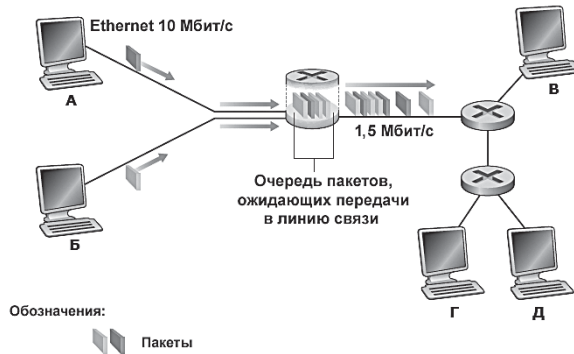


Рис. 1.12 Коммутация пакетов

Таблица маршрутизации и протоколы маршрутизации

Ранее мы упомянули, что маршрутизатор принимает пакеты, поступающие на одно из его соединений, и перенаправляет их на другое. Но как маршрутизатор определяет, куда направить пакет? На самом деле в различных видах компьютерных сетей перенаправление пакетов происходит различными методами. Здесь мы вкратце опишем, как это делается в Интернете.

Каждая конечная система в Интернете имеет свой адрес, называемый IP-адресом. Когда одна конечная система (источник) пытается отправить пакет на другую, то она включает в заголовок пакета IP-адрес места назначения. Как и в случае с почтовыми адресами, он имеет иерархическую структуру. Когда пакет прибывает на маршрутизатор, находящийся в сети, тот проверяет часть пакета, содержащую адрес места назначения и в соответствии с ним направляет пакет по необходимому пути. Если углубиться дальше, каждый маршрутизатор имеет **таблицу маршрутизации**, которая ставит в соответствие адреса места назначения (либо часть адресов места назначения) с исходящими соединениями маршрутизатора. Когда пакет прибывает на маршрутизатор, тот проверяет адрес и находит в таблице маршрутизации соответствующее исходящее соединение, куда и направляет данный пакет.

Процесс маршрутизации от источника к приемнику аналогичен ситуации, когда автолюбитель вместо того, чтобы воспользоваться картой, предпочитает спросить, как проехать. Предположим, например, Джо, который живет в Филадельфии, хочет навестить бабушку, живущую в другом штате по адресу 156, Лэйксайд-Драйв, Орландо, Флорида. Первым делом он едет на ближайшую заправочную станцию и спрашивает, как добраться до дома 156, Лэйксайд-Драйв, Орландо, Флорида. Сотрудник заправочной станции извлекает из адреса часть со словом «Флорида» и говорит Джо, что ему нужно на шоссе I-95 Юг, которое проходит рядом с АЗС. Также он сообщает Джо, что как только тот въедет во Флориду, ему следует там кого-нибудь спросить, куда направляться дальше. Джо едет по I-95 Юг, пока не добирается до местечка Джексонвилл во Флориде, где спрашивает еще одного работника заправочной станции, как проехать дальше. Работник выделяет слово «Орландо» из адреса и сообщает Джо, что ему нужно продолжать движение по шоссе I-95 до Дайтона-Бич, а там уже спросить еще кого-нибудь. Работник следующей АЗС в Дайтона-Бич говорит Джо, что ему нужно ехать по I-4, и он попадет прямо в Орландо. Джо едет по I-4, попадает в Орландо и опять направляется на заправочную станцию. На этот раз сотрудник выделяет часть адреса «Лэйксайд-Драйв» и указывает дорогу к шоссе «Лэйксайд-Драйв». Как только Джо выезжает на «Лэйксайд-Драйв», он спрашивает ребенка с велосипедом, как добраться до места назначения. Ребенок выделяет в адресе часть «156» и указывает Джо на нужный ему дом. Наконец Джо достигает места назначения. В указанной выше аналогии все работники заправочных станций, а также ребенок на велосипеде являются аналогами маршрутизаторов.

Мы только что узнали, что маршрутизатор использует адрес места назначения в пакете, чтобы найти его в таблице маршрутизации и определить подходящее выходное соединение. Но возникает вопрос: откуда берутся таблицы маршрутизации? Конфигурируются ли они вручную на каждом из маршрутизаторов либо в Интернете используются какие-то автоматические процедуры? Это мы изучим подробнее в главе 4. Но чтобы удовлетворить ваше любопытство прямо сейчас, мы отметим, что в Интернете существуют специальные **протоколы маршрутизации**, которые используются для автоматической генерации таблиц маршрутизации. Протокол маршрутизации может, например, определять кратчайший путь от маршрутизатора до любого места назначения и использовать эти результаты для конфигурации таблиц в маршрутизаторах.

Вы действительно хотите узнать маршрут, по которому пакеты проходят через Интернет от источника к приемнику? Тогда мы приглашаем вас попробовать поработать с программой Traceroute. Посетите сайт **www.traceroute.org**, выберите источник в определенной стране и проследите маршрут от этого источника до вашего компьютера. (Обсуждение Traceroute см. в разделе 1.4.)

1.3.2. Коммутация каналов

Существует два фундаментальных подхода к методам передачи данных по сетям: **коммутация каналов** и **коммутация пакетов**. Рассмотрев сети коммутации пакетов в предыдущем разделе, давайте обратим наше внимание на сети с коммутацией каналов.

В сетях с коммутацией каналов ресурсы, необходимые для обеспечения взаимодействия между конечными системами (буфер, скорость передачи), *резервируются* на время соединения между системами. В сетях с коммутацией пакетов эти ресурсы *не резервируются*; сеанс взаимодействия использует ресурсы по запросу и как следствие может ожидать (то есть вставать в очередь) доступа к освободившемуся соединению. В качестве простой аналогии, рассмотрим пример с двумя ресторанами, в одном из которых требуется предварительное резервирование столиков, а в другом приглашаются все желающие, без резервирования. Чтобы заказать обед в первом ресторане, мы должны предварительно договориться об этом по телефону, прежде чем туда идти. Но зато, когда мы прибудем в ресторан, мы можем немедленно занять резервированное место и заказать обед. Чтобы пообедать во втором ресторане, нам не нужно туда звонить, но придя, мы можем обнаружить, что все столики заняты, и нам придется ожидать освободившегося места.

Традиционные телефонные сети являются примером сетей с коммутацией каналов. Посмотрим, что происходит, когда один человек желает послать информацию (голосовое или факсимильное сообщение) другому через телефонную сеть. Перед тем как отправитель посылает информацию, в сети должно установиться соединение между получателем и отправителем. Это *добросовестное* соединение, которое поддерживается коммутаторами на всем его протяжении. На жаргоне телефонии такое соединение называется **каналом**. Когда устанавливается в сети канал, также резервируется постоянная скорость передачи в линии соединения (которая представляет собой часть пропускной способности этого соединения) на все время соединения. Так как данная скорость пе-

редачи резервируется для этого соединения между передатчиком и приемником, то в дальнейшем данные могут передаваться с *гарантированной* постоянной скоростью.

На рис.1.13 показана сеть с коммутацией каналов, в которой четыре коммутатора соединены между собой линиями связи. Каждая из этих линий связи имеет четыре канала, то есть поддерживает одновременно четыре соединения. Хосты (персональные компьютеры или рабочие станции) соединены напрямую с одним из коммутаторов. Когда два хоста хотят обмениваться информацией, между ними устанавливается выделенное **сквозное соединение**. Таким образом, для того чтобы хост А мог взаимодействовать с хостом Б, нужно зарезервировать один канал на каждой из двух линий связи. В данном примере выделенное соединение использует второй канал первой линии связи и четвертый канал второй линии связи. Так как в линиях связи содержится по четыре канала, то для каждой линии, используемой в выделенном соединении, используется полоса пропускания, равная полосе пропускания линии связи, на все время соединения. Таким образом, например, если каждая линия связи между соседними коммутаторами имеет скорость передачи 1 Мбит/с, то любому из сквозных коммутируемых соединений доступна скорость 250 Кбит/с.

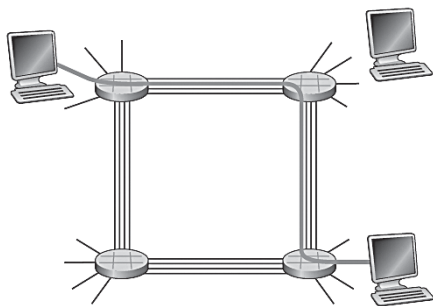


Рис. 1.13. Простейшая сеть с коммутацией каналов, состоящая из четырех линий связи и четырех коммутаторов

В противоположность этому рассмотрим, что происходит в сетях с коммутацией пакетов, таких как Интернет, когда один хост пытается отправить пакет другому хосту. Как и в случае с коммутацией каналов, пакет также проходит через последовательность линий связи. Но существенное отличие здесь в том, что пакет отсылается в сеть без резервирования каких-либо ресурсов линий соединения. Если одна из линий связи перегружена вследствие передачи по ней других пакетов в это же

время, то пакет вынужден ожидать в буфере на передающем конце соединения и испытывать, таким образом, задержку. Отсюда следует, Интернет прикладывает все усилия для своевременной доставки пакетов, но не может ее гарантировать.

Мультиплексирование в сетях с коммутацией каналов

Канал в линии связи организуется при помощи **мультиплексирования с частотным разделением** (frequency-division multiplexing, **FDM**), либо с **временным разделением** (time-division multiplexing, **TDM**). В случае с частотным разделением спектр частот всей линии делится между установленными каналами, то есть каждому каналу на все время соединения отводится определенная полоса частот. В телефонных сетях, например, эта полоса имеет ширину 4 кГц (то есть 4000 Гц или 4000 циклов в секунду). Эта ширина называется **полосой пропускания**. Радиостанции FM-диапазона тоже используют принцип частотного разделения и делят частоты от 88 МГц до 108 МГц.

В случае с временным разделением время разбивается на фиксированные промежутки, называемые *кадрами*, а кадры, в свою очередь, делятся на фиксированное число слотов. Когда в сети устанавливается соединение, то ему выделяется один временной слот в каждом кадре. Эти временные слоты используются для передачи данных только для одного соединения.

На рис. 1.14 показан принцип функционирования линий связи, поддерживающей до 4 каналов для случаев мультиплексирования с разделением по частоте и по времени. Для частотного мультиплексирования весь частотный диапазон разделен на четыре полосы по 4 кГц каждая. Для случая с временным разделением диапазон времени делится на кадры, содержащие по 4 слота; каждому каналу связи назначен один и тот же слот в сменяющихся кадрах. Скорость передачи данных в этом случае равна произведению частоты смены кадров и числа битов в слоте. Например, если по линии передается 8000 кадров в секунду, а слот содержит 8 бит, то скорость каждого канала связи составит 64 Кбит/с.

Сторонники коммутации пакетов всегда утверждали, что коммутация каналов является очень расточительной технологией, потому что выделенные каналы вынуждены простаивать во время так называемых **периодов тишины**. Например, когда в ходе телефонного разговора человек перестает говорить, то простаивающие сетевые ресурсы (частотная полоса или временные диапазоны) не могут быть использованы для

других соединений. В качестве еще одного примера представим себе врача-рентгенолога, который использует сеть с коммутацией каналов для удаленного доступа к базе рентгеновских снимков. Рентгенолог устанавливает соединение, запрашивает снимок, изучает его, затем запрашивает снова. Во время изучения снимка и размышлений сетевые ресурсы для соединения выделены, но не используются, а значит расходуются впустую. Также сторонники коммутации пакетов любят подчеркивать тот факт, что создание соединений и резервирование каналов — непростая задача, требующая сложного программного обеспечения для управления и синхронизации.

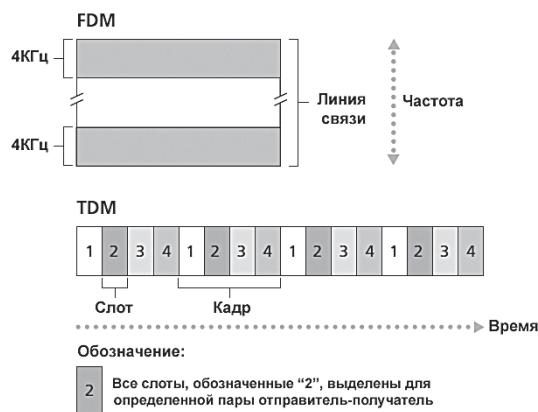


Рис. 1.14. При частотном разделении каждый канал постоянно занимает свою полосу частот. При временном канале периодически использует всю полосу пропускания (в выделенные ему временные слоты)

Перед тем как завершить обсуждение коммутации каналов, давайте рассмотрим числовой пример, который поможет еще лучше понять суть технологии коммутации пакетов. Давайте определим, сколько времени понадобится, чтобы отослать файл размером 640 000 бит с хоста А на хост Б в сети с коммутацией каналов. Предположим, что все линии связи в сети используют метод временного разделения с 24 слотами в кадре, и скорость передачи составляет 1,536 Мбит/с. Также предположим, что на установление соединения перед тем, как хост А начнет передавать файл, уйдет 500 мс. Каково тогда время отправки файла? Каждый канал имеет скорость передачи, равную 1,536 Мбит/с : 24 = 64 Кбит/с. Поэтому время передачи файла равно 640 000 бит : 64 Кбит/с = 10 с. К этим 10 с прибавляем время на установление соединения в канале и получаем 10,5 с — общее время передачи файла. Отметим, что время передачи не зависит от числа линий связи и бу-

дет равно примерно 10 с и в случае с одной линией, и в случае со ста линиями (на самом деле реальная задержка при передаче источник-приемник включает еще и задержку распространения, которую опишем в разделе 1.4).

Коммутация пакетов или коммутация каналов

Описав методы коммутации пакетов и коммутации каналов, давайте теперь сравним их. Противники коммутации пакетов часто утверждают, что она не подходит для приложений, работающих в реальном времени, например телефонии и видеоконференций, из-за того, что задержки передачи от источника к приемнику непостоянны и непредсказуемы (имеются в виду задержки ожидания). Сторонники же пакетной коммутации возражают и говорят, что данный метод дает возможность лучшего разделения емкости канала между пользователями, чем коммутация каналов. Также он проще, эффективнее и менее затратен. Очень интересное обсуждение коммутации пакетов против коммутации каналов можно посмотреть вот тут³⁴⁹. Вообще люди, которые не любят резервировать столики в ресторане, предпочитают коммутацию пакетов, а не коммутацию каналов.

Почему коммутация пакетов является более эффективной? Посмотрим на простой пример. Предположим, что пользователи делят канал 1 Мбит/с. Допустим также, что каждый пользователь чередует периоды активности, во время которых передает данные с постоянной скоростью 100 Кбит/с, и периоды бездействия, во время которых никаких данных не передает. Предположим также, что пользователь активен 10% всего времени, а все остальные 90%, например, пьет кофе. В случае с коммутацией каналов для *каждого* пользователя должно быть *зарезервировано* 100 Кбит/с на все время сеанса связи. Например, при мультиплексировании с временным разделением, если односекундный кадр делится на 10 временных слотов по 10 мс каждый, то каждому пользователю выделится 1 временной слот в каждом кадре.

Таким образом, соединение линии связи в сети с коммутацией каналов может поддерживать только 10 одновременных пользователей (равно 1 Мбит/с:100 Кбит/с). В случае с коммутацией пакетов вероятность того, что определенный пользователь активен, равна 0,1 (то есть 10%). Если в сети 35 пользователей, то вероятность того, что 11 из них одновременно активны, равна приблизительно 0,0004 (как получено данное значение, узнаете в упражнении 8). Когда одновременно активно 10 или меньше пользователей (что случается с вероятностью

0,9996), то общая скорость поступления данных меньше либо равна 1 Мбит/с, то есть не превышает пропускной способности линии связи, и в этом случае пользовательские пакеты проходят по ней без задержек, как в случае с коммутацией каналов. Когда в сети будет более чем 10 активных пользователей одновременно, то общая скорость поступления пакетов превысит пропускную способность линии, и начнет увеличиваться выходная очередь пакетов (она будет продолжать расти до тех пор, пока общая скорость входного потока не понизится до 1 Мбит/с). Поскольку вероятность одновременной активности 10 пользователей в данном примере очень мала, то метод с коммутацией пакетов покажет ту же производительность, что и метод с коммутацией каналов, *но при этом разрешит работать в сети втрое большему количеству пользователей.*

Теперь давайте рассмотрим еще один простой пример. Предположим, есть 10 пользователей в сети, и один из них внезапно генерирует 1000 1000-битных пакетов, в то время как другие пользователи не являются активными, не генерируя ничего. В сети с коммутацией каналов и временным разделением, когда каждый кадр делится на 10 слотов, и в каждом слоте содержится 1000 бит, активный пользователь может использовать один временной слот на кадр, чтобы передать данные, в то время как оставшиеся 9 временных слотов в каждом кадре будут простаивать. Тогда, чтобы передать весь миллион бит активного пользователя, уйдет 10 секунд. В случае с коммутацией пакетов активный пользователь сможет непрерывно посылать свои пакеты в канал связи с максимальной скоростью передачи в 1 Мбит/с, так как больше никто не передает пакеты в сеть. В этом случае все данные активного пользователя будут переданы за 1 с.

Указанная пара примеров демонстрирует нам преимущество в производительности метода коммутации пакетов перед методом коммутации каналов. Это также подчеркивает серьезную разницу между двумя формами того, как разделяется на несколько потоков данных скорость передачи в линии связи. Коммутация каналов резервирует емкость независимо от запросов, и поэтому зарезервированное, но неиспользуемое время остается потраченным впустую. Коммутация пакетов, с другой стороны, использует канал *по запросу*. Скорость передачи делится среди тех пользователей, кому нужно передать пакеты по каналу связи.

Хотя в нынешних телекоммуникационных системах работают обе технологии коммутации, тенденция направлена в сторону коммутации

пакетов. На нее постепенно переходят даже многие телефонные сети, и очень часто используют данный метод коммутации для межконтинентальных телефонных вызовов.

1.3.3. Сеть сетей

Мы видели ранее, что конечные системы (персональные компьютеры, смартфоны, веб-серверы, почтовые серверы и так далее) соединяются с Интернетом, пользуясь услугами Интернет-провайдеров. Интернет-провайдер обеспечивает либо проводной, либо беспроводной доступ, используя ряд технологий, включающих DSL, кабельный доступ, FTTH, Wi-Fi, а также мобильные сотовые сети. Заметим, что Интернет-провайдером необязательно бывают телефонные компании либо компании кабельных сетей; им может быть, например, университет (предоставляющий доступ в Интернет своим студентам и персоналу факультетов) либо коммерческая компания (обеспечивающая доступ в Интернет своим сотрудникам). Но соединение конечных пользователей, провайдеров контента, сетей доступа, Интернет-провайдеров — это только малая часть сложной мозаики, состоящей из миллионов конечных систем, которые образуют Интернет. Чтобы закончить эту мозаику, сети доступа Интернет-провайдеров должны быть сами взаимосвязаны между собой. Это делается путем создания *сети сетей* — понимание этой фразы является ключом к пониманию Интернета.

По прошествии многих лет сеть сетей, которая сформировала Интернет, развилась в очень сложную структуру. Развитие это во многом определялось не фактором производительности, а экономикой и национальной политикой. Для того чтобы лучше понять структуру сегодняшнего Интернета, давайте построим по шагам серию сетевых структур, на каждом шаге приближаясь к сложному Интернету, который мы имеем сегодня. Вспомним, что главной целью является связать все сети доступа таким образом, что все конечные системы могли отсылать пакеты друг другу. Самым простым и достаточно наивным подходом было бы соединить каждую сеть доступа *напрямую* со всеми другими сетями доступа. Получившаяся в результате сеть была бы очень дорогая, так как потребовала бы отдельной линии связи для каждого соединения сети доступа с сотнями тысяч других таких же сетей по всему миру.

Первый вариант, *Сетевая Структура 1*, объединяет все сети доступа Интернет-провайдеров с *одним магистральным Интернет-провайдером*.

Наш (воображаемый) магистральный провайдер — это сеть маршрутизаторов и коммуникационных каналов связи, которая охватывает не только весь земной шар, но и имеет по крайней мере по одному маршрутизатору для каждой из сотен тысяч сетей доступа. Конечно, было бы очень затратно для провайдера построить такую широкую сеть. Чтобы получать прибыль, провайдер должен брать плату за соединение с каждой из сетей доступа, и ее размер должен отражать (но необязательно пропорционально) объем трафика, которым сеть доступа обменивается с провайдером. Так как сеть доступа оплачивает прохождение трафика провайдеру, то сеть доступа можно назвать **заказчиком**, а провайдера **поставщиком**.

Если какая-то компания строит такую глобальную сеть, которая приносит прибыль, вполне естественно, что другие компании захотят тоже стать магистральными провайдерами. Мы приходим к *Сетевой Структуре 2*, состоящей из сотен тысяч сетей доступа и *нескольких* магистральных провайдеров. Естественно, сети доступа как заказчики будут предпочитать Сетевую Структуру 2, а не 1, так как в этом случае у них есть выбор между провайдерами, которых они могут предпочесть и посчитать более выгодными по цене и предлагаемым услугам. Отметим, что сети провайдеров в данном случае должны быть связаны между собой, а иначе клиенты разных магистральных провайдеров не смогли бы взаимодействовать друг с другом.

Сетевая Структура 2 представляет собой двухуровневую иерархию, в которой магистральные провайдеры занимают верхний уровень, а сети доступа лежат на нижнем. Такая структура предполагает, что магистральный провайдер не только имеет возможность соединяться со всеми сетями доступа, но и находит экономически выгодным делать это. Несмотря на то, что магистральные провайдеры имеют очень внушительное сетевое покрытие и связаны с очень многими сетями доступа, не все из них присутствуют в любом городе на планете. Вместо этого существуют **региональные Интернет-провайдеры**, которые осуществляют подключение сетей доступа в каждом регионе, а те, в свою очередь, соединены с **Интернет-провайдерами первого уровня**. Существует приблизительно дюжина провайдеров первого уровня, среди которых Level 3 Communications, ATMT, Sprint и NTT. Интересно отметить, что ни один из провайдеров прямо не заявляет о своей принадлежности к первому уровню. Как говорится, если вы сомневаетесь, входите ли в узкий круг — то, вероятно, вы в него не входите.

Возвращаясь к этой сети сетей, заметим, что конкурируют не только провайдеры первого уровня, но и многочисленные местные Интернет-провайдеры в своем регионе. В такой иерархической структуре каждая сеть доступа вынуждена платить за соединение местному Интернет-провайдеру, а каждый местный Интернет-провайдер — провайдеру первого уровня (сеть доступа может быть соединена напрямую с провайдером первого уровня и в этом случае платить за подключение непосредственно ему). Таким образом, на каждом из уровней иерархии существуют отношения заказчик-поставщик. Заметим также, что провайдеры первого уровня не платят никому, так как они находятся на вершине иерархии. На самом деле структура может быть еще сложнее, например, в некоторых регионах существуют более крупные местные Интернет-провайдеры (иногда охватывающие целую страну), к которым подсоединяются более мелкие местные провайдеры, а крупные соединяются с провайдерами первого уровня. Например, в Китае в каждом городе есть сети доступа, которые соединяются с провайдерами провинции, а те, в свою очередь, с национальными Интернет-провайдерами, соединяющимися с провайдерами первого уровня⁶³⁶. Таким образом, мы пришли к *Сетевой Структуре 3*, которая является все еще неточным приближением сегодняшнего Интернета.

Для построения сетевой структуры, которая наиболее близко отражает сегодняшний Интернет, мы должны добавить еще точки присутствия (Points of Presence или PoP), пиринг, точки обмена трафиком (IXP), а также возможность использования многоинтерфейсного режима. Точки присутствия существуют на всех уровнях иерархии, исключая нижний (сети доступа). **Точка присутствия** — это группа из одного или нескольких маршрутизаторов (расположенных в одном и том же месте) сети провайдера, к которым могут подключаться маршрутизаторы сети заказчика. Чтобы подключиться к Интернет-провайдеру, заказчик может арендовать высокоскоростные линии связи какой-нибудь телекоммуникационной компании для соединения своих маршрутизаторов с маршрутизаторами провайдера в точке присутствия. Любой Интернет-провайдер (за исключением провайдера первого уровня) может выбрать так называемое **многоинтерфейсное** подключение, то есть соединение с двумя или более провайдерами верхнего уровня. Так, например, сеть доступа может иметь многоинтерфейсное подключение к двум местным Интернет-провайдерам или, как вариант, к двум местным и к одному провайдеру первого уровня. Аналогично, местный Интернет-провайдер может иметь многоинтерфейсное подключение к нескольким провайдерам первого уровня. Используя такое подключение, провайдер гаранти-

рует для себя отправку и получение пакетов, если у одного из его провайдеров проблемы на линии связи.

Как мы уже выяснили, заказчики платят Интернет-провайдерам за доступ в сеть, и размер этой платы, как правило, отражает объем трафика, которым заказчик обменивается с поставщиком. Для уменьшения затрат пара соседствующих Интернет-провайдеров одного уровня иерархии может установить между собой, так называемое **пиринговое** соединение, то есть соединить свои сети напрямую таким образом, чтобы трафик между ними не шел через промежуточные каналы связи. Обычно по соглашению сторон такое соединение является бесплатным. Как уже упоминалось, провайдеры первого уровня также устанавливают между собой пиринговые соединения. Обсудить пиринг и взаимоотношения провайдеров, заказчиков Интернет-услуг можно здесь⁶⁴³. Компании, не являющиеся Интернет-провайдерами (так называемые третьи стороны), могут создавать **точки обмена Интернет-трафиком** (Internet Exchange Point, **IXP**), обычно в отдельно стоящем здании, своими собственными коммутаторами. Через такие точки провайдеры могут устанавливать пиринговое взаимодействие между собой. На сегодняшний день в Интернете существует примерно 300 точек обмена трафиком³¹. Таким образом, мы приходим к нашей Сетевой Структуре 4, состоящей из точек доступа, региональных провайдеров, провайдеров первого уровня, точек присутствия, многоинтерфейсного режима, пиринга и точек обмена трафиком (IXP).

Наконец, мы пришли к *Сетевой Структуре 5*, которая описывает сегодняшний Интернет (по состоянию 2012 года). Сетевая структура, показанная на рис. 1.15, происходит из Структуры 4 с добавлением **сетей провайдеров контента**. Одним из ярких представителей таких провайдеров контента является компания Google. На момент написания данной книги, по приблизительным оценкам, Google имела от 30 до 50 центров обработки данных, размещенных по всей Северной Америке, Европе, Азии, Южной Америке и Австралии. Многие из этих центров включали в себя сотни серверов, а некоторые и до сотни тысяч. Все центры обработки данных компании Google взаимосвязаны частной сетью Google (ТСР/IP сетью), которая охватывает весь земной шар, но, тем не менее, отделена от публичного Интернета. Важно заметить, что трафик в частной сети Google идет только между серверами Google. Как мы видим на рис. 1.15, частная сеть Google пытается выступать в качестве провайдера первого уровня, устанавливая пиринговое соединение (бесплатное) с Интернет-провайдерами нижнего уровня, либо связываясь с ними на-

прямую, либо через точки обмена трафиком³⁰³. Однако из-за того, что подключиться ко многим сетям доступа можно только через провайдера первого уровня, Google также устанавливает соединение с провайдерами первого уровня и платит им за трафик, которым с ними обменивается. Создавая собственные сети, провайдер контента не только сокращает расходы, связанные с подключением провайдера более высокого уровня, но также получает возможность оптимизировать управление своими сервисами, предоставляемыми конечным пользователям. Более подробно инфраструктуру сети Google мы опишем в разделе 7.2.4.

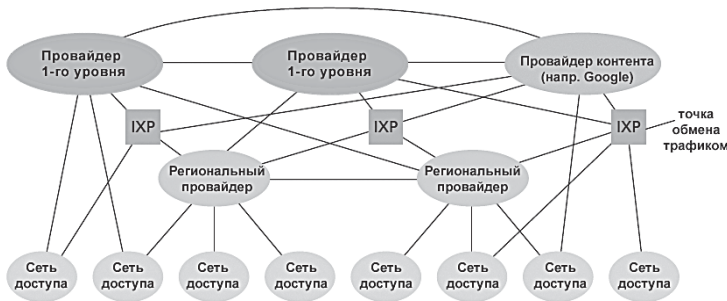


Рис. 1.15. Взаимодействие сетей

Резюмируя все вышесказанное, заметим, что сегодняшний Интернет — сеть сетей — это сложная структура, состоящая примерно из дюжины Интернет-провайдеров первого уровня и сотен тысяч провайдеров нижних уровней. Некоторые из них охватывают небольшие географические области, а некоторые — многие континенты и океаны. Провайдеры нижнего уровня соединяются с провайдерами верхнего уровня, а также взаимодействуют друг с другом. Пользователи и провайдеры контента являются заказчиками провайдеров нижних уровней, а те, в свою очередь, заказчиками провайдеров верхних уровней. В последние годы большинство провайдеров контента создали свои собственные сети, которые соединяются с сетями провайдеров нижних уровней там, где это возможно.