Sams Teach Yourself Cisco® Routers in 21 Days

Second Edition

Jerome F. DiMarzio

SAMS

HIGH TECH

Маршрутизаторы Cisco®

Пособие
для самостоятельного изучения

Второе издание

Джером Ф. Димарцио

Санкт-Петербург — Москва
2003

Серия «High tech»

Джером Ф. Димарцио

Маршрутизаторы Cisco

Пособие для самостоятельного изучения

Перевод П. Шера

Главный редактор	*А. Галунов*
Зав. редакцией	*Н. Макарова*
Научный редактор	*А. Лапин*
Редактор	*М. Буйневич*
Художник	*В. Гренда*
Корректура	*С. Беляева*
Верстка	*Н. Гриценко*

Димарцио Д. Ф.

Маршрутизаторы Cisco. Пособие для самостоятельного изучения. – Пер. с англ. – СПб: Символ-Плюс, 2003. – 512 с., ил.

ISBN 5-93286-048-0

Книга Дж. Димарцио существенно отличается от книг по Cisco, посвященных подготовке к сертификационным экзаменам. Она охватывает все вопросы, необходимые для изучения основ маршрутизации Cisco. Вы получите глубокие знания о Cisco IOS и пользовательском интерфейсе, сможете настроить маршрутизатор Cisco «с нуля» и сконфигурировать его для работы практически в любой локальной или глобальной сети. Вы познакомитесь с командами, используемыми для настройки протоколов IP, IPX, RIP, IGRP, EIGRP, OSPF и BGP. Отдельная глава посвящена методам обеспечения безопасности: спискам доступа IP и трансляции сетевых адресов (NAT). Богатый опыт автора в сочетании с умением просто и доступно говорить о сложных проблемах делают эту книгу бесценной для новичков, изучающих сети и технологии маршрутизации Cisco.

ISBN 5-93286-048-0
ISBN 0-672-32296-X (англ)

Издательство «Символ-Плюс». 199034, Санкт-Петербург, 16 линия, 7.
тел. (812) 324-5353, edit@symbol.ru. Лицензия ЛП N 000054 от 25.12.98.
Налоговая льгота – общероссийский классификатор продукции
ОК 005-93, том 2; 953000 – книги и брошюры.
Подписано в печать 25.02.2003. Формат 70x100 1/16 . Печать офсетная.
Объем 32 печ. л. Тираж 2000 экз. Заказ N
Отпечатано с диапозитивов в Академической типографии «Наука» РАН
199034, Санкт-Петербург, 9 линия, 12.

Оглавление

Об авторе

Джером Ф. Димарцио (Jerome F. DiMarzio) – специалист по сетям, имеющий десятилетний опыт работы по проектированию и администрированию систем. С 1991 года работал консультантом в Walt Disney Company и Министерстве обороны США.

В настоящее время Дж. Ф. Димарцио работает техническим консультантом крупной финансовой организации в центре Массачусетса. Он обладатель сертификатов MCP, MCP+I, MCSE и CCNA. Имея статус GOLD члена IEEE, Димарцио входит в состав специальных групп компьютерного сообщества по изучению искусственного интеллекта и по изучению информационных технологий для бизнес-приложений.

О рецензенте

Мишель Труман (Michelle Truman), сертифицированный эксперт по межсетевому взаимодействию (CCIE # 8098), имеет ученую степень в области телекоммуникаций, одновременно являясь бакалавром гуманитарных наук (колледж св. Екатерины в Сент-Поле, Миннесота). Мишель 8 лет руководила разработками и проектами в банковской отрасли и пищевой промышленности для нескольких крупных компаний, в том числе AgriBank, Pillsbury, Burger King, Guinness Brewing и Haagen Dazs. Она вела курсы по LAN- и WAN-технологиям и протоколам маршрутизации BGP и OSPF. Сейчас она главный консультант AT&T Business Internet Services.

О научном редакторе

Андре Паре-Хафф (Andre Paree-Huff) (сертификаты CCNP, CCDA, MCSE+I, ASE, A+, Network+, I-Network+, Server+) имеет десятилетний стаж в области вычислительной техники. В настоящее время он работает инженером по сетевой поддержке третьего уровня для Compaq Computer Corporation в Северо-американском центре обслуживания клиентов, находящемся в Колорадо Спрингс, Колорадо. Андре занимается выявлением неисправностей сетевого оборудования, специализируясь на втором и третьем уровнях модели OSI. Он является соавтором пяти книг на сетевую тематику, а также научным редактором более двух десятков изданий. Сейчас он готовится к сдаче на сертификаты CCDE и CCIE.

Посвящение

Моей удивительной семье: Сюзанне, Кристиану и Софии.

Благодарности

Прежде всего, я хотел бы поблагодарить Вильяма Е. Брауна (William E. Brown), Марка Ренфроу (Mark Renfrow), Мэтта Персела (Matt Purcell) и коллектив SAMS Publishing, а также Молли Реденбоу (Molly Redenbaugh), Джерома Колберна (Jerome Colburn), Дэйва Мейсона (Dave Mason) и всех остальных сотрудников редакции.

Я также хотел бы выразить свою благодарность (порядок упоминания не имеет значения) Уолту Адамсу (Walt Adams) из Double Eagle Services в центральной Флориде, Лу Веллу (Lou Vella) и коллективу Allmerica Financial, Хизер Остерло (Heather Osterloh) и всем, кто помог этой книге появиться на свет.

Большое спасибо моей жене Сюзанне и двум моим замечательным детям, Кристиану и Софии, за то, что они научили меня ценить каждую минуту бесконечной рутинной работы и поисков. Хотелось бы также поблагодарить Диану Митчел (Diana Mitchell). Наконец, я, наверное, провалился бы сквозь землю со стыда, если бы не поблагодарил своих родителей, Джерома и Агнес, и моего брата Мэтта.

Я благодарю всех, кто помог в создании этой книги, и если я кого-то случайно забыл или пропустил, то приношу свои искренние извинения.

Пишите нам, что вы думаете!

Именно читатель является для нас самым ценным критиком и комментатором. Нам необходимо ваше мнение, чтобы знать, что мы делаем хорошо, что могли бы делать лучше, какие области вы хотели бы видеть охваченными в наших публикациях; интересны и любые другие мудрые слова, которые вы захотите нам написать.

Будучи помощником издателя Sams Publishing, я с удовольствием приму ваши замечания и предложения. Можете отправить факс, электронное письмо или написать лично мне, чтобы поделиться тем, что вам понравилось или не понравилось в данной книге, а также сообщить, что мы можем сделать для того, чтобы наши книги стали лучше.

Пожалуйста, обратите внимание, что я не смогу помочь вам с техническими проблемами, связанными с предметом рассмотрения этой книги, а также, возможно, не смогу ответить на все письма из-за большого объема получаемой корреспонденции.

Адресуя нам сообщение, пожалуйста, убедитесь, что вы назвали в нем имя автора и название книги, а также ваше имя и номер факса или контактного телефона. Я внимательно изучу ваши комментарии и обсужу их с автором и редакторами, работавшими над изданием.

Факс: 317-581-4770

E-mail: *feedback@samspublishing.com*

Адрес: Jeff Koch
Sams Publishing
201 West 103rd Street
Indianapolis, IN 46290 USA

Предисловие

Цель книги «Маршрутизаторы Cisco» заключается в обеспечении читателей основными «кирпичиками» знаний, необходимых для успешного изучения маршрутизации Cisco. В этом издании предложен независимый от особенностей программ сертификационных экзаменов взгляд на наиболее важные понятия, которые следует знать каждому начинающему пользователю Cisco.

Круг вопросов, обсуждаемых в данной книге, и приведенные в ней примеры основываются на более чем десятилетнем опыте автора в области сетевых решений и технологий маршрутизации. Джером Ф. Димарцио построил книгу как введение, дающее основные знания тем, кто нуждается в фундаменте, на котором будет базироваться их будущий опыт маршрутизации.

Структура книги

Издание содержит всю информацию, необходимую для изучения маршрутизаторов Cisco. Книга состоит из глав, разбитых на три части, при этом сложность тем возрастает с каждой частью.

Первая часть знакомит с основными элементами Cisco:

Глава 1 «Введение в маршрутизацию Cisco: технология и компания». В этой главе вводятся основные понятия, кратко описывается история Cisco и те проблемы, для решения которых были созданы продукты компании. Задается тональность дальнейшего изложения.

Глава 2 «Введение в аппаратную часть Cisco». Материал данной главы поможет читателю осознать различия, имеющиеся в аппаратных средствах маршрутизации, и идентифицировать части маршрутизаторов, упоминающиеся на протяжении книги.

Глава 3 «Введение в Cisco IOS». В этой главе исследуется и поясняется операционная система маршрутизатора Cisco, а также рассказывается, как ее запустить и как установить обновления.

Глава 4 «Пользовательский интерфейс Cisco IOS». В данной главе объясняется, как ориентироваться в пользовательском интерфейсе Cisco IOS и как с его помощью конфигурировать маршрутизатор.

Глава 5 «Перемещение данных маршрутизаторами». Отклоняясь слегка от тем, непосредственно связанных с Cisco, эта глава охватывает технические аспекты переноса данных с одного места в другое.

Глава 6 «Запуск маршрутизатора и работа с ним». В данной главе шаг за шагом объясняется, как настроить маршрутизатор Cisco (при первом включении) для выполнения основных операций.

Глава 7 «Резервное копирование маршрутизаторов Cisco». В последней главе первой части описываются действия, необходимые для резервного копирования и сохранения конфигураций, созданных в предыдущей главе.

Вторая часть посвящена изучению основных протоколов, применяемых в маршрутизации:

Глава 8 «Введение в маршрутизируемые протоколы». В данной главе объясняется функционирование маршрутизируемых протоколов в терминах, не привязанных к какому-либо конкретному производителю сетевых продуктов.

Глава 9 «Изучение основ IP». Глава знакомит с наиболее популярным в настоящее время маршрутизируемым протоколом Интернета – протоколом IP.

Глава 10 «Настройка протокола IP на маршрутизаторе Cisco». Из этой главы читатель узнает, как запустить на маршрутизаторе протокол IP и связанные с ним утилиты.

Глава 11 «Введение в сегментированные сети». Глава, также связанная с IP, посвящена разделению сетей на подсети и маршрутизации в подсетях и сегментированных сетях.

Глава 12 «Настройка протокола IPX». В данной главе изучается процесс конфигурирования маршрутизатора Cisco для работы с протоколом IPX (Internetwork Packet Exchange).

Глава 13 «Введение в протоколы глобальных сетей». На страницах этой главы рассматривается функционирование протоколов, используемых в глобальных сетях (WAN), таких как Frame Relay.

Глава 14 «Введение в протоколы маршрутизации». Последняя глава второй части знакомит с протоколами маршрутизации – основной темой следующей части.

В последней части обсуждаются более сложные темы, такие как безопасность и протоколы маршрутизации:

Глава 15 «Конфигурирование RIP». В данной главе изучаются команды, необходимые для настройки протокола маршрутизации RIP на маршрутизаторе Cisco.

Глава 16 «Использование IGRP и EIGRP». В главе объясняется, как настроить на маршрутизаторе Cisco протокол маршрутизации внутреннего шлюза (IGRP) и его усовершенствованный вариант – EIGRP.

Глава 17 «Конфигурирование OSPF». В этой главе рассматривается еще один популярный протокол маршрутизации – протокол первоочередного открытия кратчайших маршрутов (OSPF).

Глава 18 «Введение в BGP». Здесь рассмотрены команды, применяемые для настройки протокола граничного шлюза (BGP).

Глава 19 «Изучение IS-IS». Данная глава посвящена нечасто упоминаемому протоколу, обычно считающемуся сложной темой. Рассмотрены команды настройки протокола взаимодействия промежуточных систем (IS-IS).

Глава 20 «Основы обеспечения безопасности Cisco». Эта глава знакомит с трансляцией сетевых адресов (NAT) и списками доступа IP.

Глава 21 «Основы маршрутизации на коммутаторе Cisco Catalyst и PNNI». Заключительная глава этой книги посвящена еще одному сложному вопросу: настройке коммутатора Cisco Catalyst для маршрутизации PNNI (Private Network-to-Network Interface).

Для кого написана эта книга

Эта книга не охватывает всех концепций и тем, необходимых для свободного владения бесчисленными командами и функциями Cisco. Тем не менее, каждый, кто собирается изучить маршрутизаторы Cisco, должен начать с этой книги. Здесь приведены все основные сведения, необходимые IT-профессионалам, а также читателям, не имеющим опыта работы с маршрутизаторами. Каждый, кто собрался проложить себе дорогу в мир сетей и маршрутизации, получит неоценимую помощь от изучения того практического опыта, который положен в основу этой книги.

Используемые обозначения

В тексте вам встретятся следующие обозначения:

Примечание

Содержит комментарии и дополнительные сведения по рассматриваемым темам.

В конце каждой главы вы найдете разделы «Резюме» и «Вопросы и ответы». Многие из глав содержат упражнения, которые позволят вам проверить приобретенные знания.

Кроме того, в книге приняты следующие типографские соглашения:

- Команды, имена переменных, каталогов и файлов напечатаны специальным моноширинным шрифтом.
- Команды, вводимые пользователем, выделены **полужирным моноширинным шрифтом**.
- Заполнители в описании синтаксиса выделены *моноширинным шрифтом в курсивном начертании*. Это означает, что вы должны заменить заполнитель действительным именем файла, параметром или другим элементом, который он представляет.

I

Основы маршрутизации Cisco

1

Введение в маршрутизацию Cisco: технология и компания

Настоящее издание содержит материал, необходимый для понимания процесса маршрутизации, выполняемого маршрутизаторами Cisco. Технические специалисты сегодня весьма востребованы, и нигде они не нужны так, как в области маршрутизации. Маршрутизация информации – это то, что заставляет «вращаться мир» в нашем изменчивом обществе, по мере того как оно становится все более и более зависимым от информации и от ее доставки из одного места в другое.

Вспомните о технологических новинках последнего десятилетия. Большинство из них связано с Интернетом (либо с другой похожей технологией повсеместного доступа). Теперь представьте себе эти достижения в условиях отсутствия возможности регулировать потоки информации. Многие услуги, ставшие привычными, такие как электронная почта, электронная торговля, интернет-телефония, станут практически невозможными. Большинство людей не сознает, как важна маршрутизация в технологическом обществе. Глобальная экономика находится в зависимости от способности доставлять информацию из одной системы в другую.

Компания Cisco давно лидирует в технологии маршрутизации, благодаря чему инженеры, имеющие опыт работы с ее оборудованием, высоко ценятся на корпоративном рынке. Однако тут всегда приходится решать очень сложную задачу. Оборудование довольно дорогое, а от обрабатываемых им потоков данных зачастую зависит существование компаний, поэтому многие предпочитают нанимать только опытных специалистов, чтобы сократить как время их обучения, так и риск повреждения маршрутизатора стоимостью 50 000 долларов. Многие

из тех, кто решил сделать карьеру сетевого инженера, сталкиваются с вопросом: как получить необходимый опыт, не имея работы, и как получить работу, не имея достаточного опыта?

Раньше единственным ответом были дорогостоящие курсы, дающие сертификат по маршрутизации. Почему вы должны посещать их, если все, что вам нужно – это изучить основы технологии? Ответ прост – такие курсы были единственной возможностью. В настоящее время не существует вводных курсов по маршрутизаторам Cisco. Следовательно, если вы хотите изучить основы, вам придется пройти Cisco Certified Network Associate (CCNA).

Примечание

Сертификаты Cisco Career Certifications – ценный капитал для любого сетевого инженера. Однако лекции и материалы курса могут оказаться слишком дорогими, кроме того, по рекомендации самой Cisco вы должны иметь опыт не менее двух лет в области маршрутизации (предпочтительно на оборудовании Cisco), прежде чем сможете претендовать на сертификат. Опять-таки, как вы сможете получить опыт, необходимый для начала выбранной карьеры?

В последнее время все больше людей выбирают самостоятельную форму обучения. К сожалению, ощущается значительная нехватка учебных материалов (книг или компакт-дисков), не нацеленных на получение сертификата. Этому тоже есть очень убедительное объяснение. Поскольку нанимателям требуются инженеры с опытом, то что может наилучшим образом подтвердить широту ваших знаний, если не сертификат Cisco Career Certification? Поэтому значительная часть спроса, а следовательно, и денег, которые на этом можно заработать, приходится на сертификацию, и большинство книг написаны с этим уклоном. Что же в этом плохого? Многие книги для подготовки к сертификации имеют два существенных недостатка.

Первый заключается в том, что эти книги предполагают, что читатель уже имеет некоторый опыт, связанный с Cisco. (Повторим, Cisco требует, чтобы претендент на сертификат имел не менее чем двухлетний опыт работы. Эти материалы рассчитаны на людей, желающих расширить свои знания в области маршрутизации, а не на начинающих.) Отсюда следует, что читатель, имеющий небольшой опыт или не имеющий его вовсе, не сможет даже начать изучение. Многие базовые концепции, такие как конфигурация аппаратной части маршрутизатора и его интерфейсы, выпадают из рассмотрения.

Примечание

Многие книги по подготовке к сертификационным экзаменам Cisco написаны настолько хорошо, что не содержат ничего лишнего, что не потребуется для сдачи экзамена. Если у вас нет предварительных знаний о маршрутизации или хотя бы доступа к маршрутизатору Cisco, вы можете почувствовать себя обделенным.

Второй недостаток многих книг для подготовки к сертификации состоит (по определению) в том, что они учат вас лишь тому, что необходимо при прохождении конкретного теста, для которого и была написана такая книга. За кадром остаются многие основополагающие понятия и более сложные вопросы. Поскольку система сертификации Cisco построена иерархически, специфические вопросы, необходимые при прохождении первого теста (CCNA), уже не повторятся в более сложных тестах CCNP (Cisco Certified Network Professional, сертифицированный эксперт Cisco по сетевым комплексам) и CCIE (Cisco Certified Internetwork Engineer, сертифицированный специалист Cisco по межсетевому взаимодействию). И, наоборот, в книге, предназначенной для подготовки к CCNA, не встретятся вопросы, относящиеся к CCIE. Таким образом, книги, нацеленные на сертификацию, не могут охватить всех сведений, необходимых начинающему инженеру для освоения концепций маршрутизации Cisco.

Книга «Маршрутизаторы Cisco» охватывает все вопросы, необходимые специалисту для изучения основ маршрутизации Cisco. К концу этой книги вы сможете настроить маршрутизатор Cisco «с нуля» и сконфигурировать его для работы практически в любой локальной (LAN) или глобальной (WAN) сети. Будут детально изучены темы, обычно остающиеся вне рассмотрения, включая интерфейсы маршрутизаторов, обновление IOS, работу с различными аппаратными конфигурациями.

Примечание

Хотя в этой книге и не представлен в явном виде материал, необходимый для получения сертификата Cisco Career Certification (например, CCNA), многое из того, что вы узнаете, пригодится на экзамене. Но все же, если вы планируете сдавать экзамен CCNA, не полагайтесь только на эту книгу. Она ориентирована на изучение технологии, а не на прохождение тестов.

Материал книги поделен на три части (по семь глав в каждой). Каждая глава посвящена определенной теме; многие из глав основаны на материале предыдущих. Это позволит успешно изучить маршрутизацию Cisco в последовательной и логичной манере.

Формат книги таков, что позволяет изучать принципы маршрутизации так, как вы изучали бы их в процессе работы – от простых вопросов к более сложным. Даже если ваш опыт в области маршрутизации минимален, расположение глав позволяет легко следовать за излагаемым материалом. Новички смогут изучить основы, начав с глав «Введение в маршрутизацию Cisco: технология и компания» и «Введение в аппаратную часть Cisco» (главы 1 и 2) и закончив достаточно сложной темой «Основы маршрутизации на коммутаторе Cisco Catalyst и PNNI» (глава 21).

Примечание

Сценарии работы и советы по решению проблем делают эту книгу более практичной и полезной. Многие примеры, приведенные в книге, основаны на реальном опыте.

Не рекомендуется пропускать главы, однако читатели, имеющие опыт, смогут легко перейти к более сложным темам. Независимо от того, будете ли вы читать всю книгу от начала до конца или же решите пропустить некоторые главы, в каждой из них вы найдете полное изложение соответствующей темы.

Первая глава этой книги познакомит вас с компанией Cisco, с историей, предшествовавшей созданию ее продуктов, и с принципами маршрутизации.

История маршрутизаторов Cisco

До 1984 года не существовало общепринятого разумного способа доставки цифровой информации из одного места в другое. Несмотря на существование метода коммутации пакетов, появившегося в середине 60-х, требования новых компьютерных сетей и коммуникационных протоколов (таких как TCP/IP) заставляли искать новые технологические решения. Компьютерные сети, которым к тому времени едва исполнилось десять лет, только начинали становиться общедоступными.

И в 60-х, и в 70-х годах отсутствие надежных средств маршрутизации не представляло проблемы (для тех случаев, когда требовалось объединение сетей, вполне подходила коммутация пакетов). Большинство компьютерных сетей в те времена состояло из пяти-шести больших машин с десятком терминалов у каждой. Эти мэйнфреймы обычно располагались в одном месте, благодаря чему не возникало потребности в длинных кабельных соединениях. Близость расположения значительно облегчала их объединение в сети. Сеть, обслуживающая 60 пользователей, требовала лишь шести сетевых соединений, расположенных в ограниченном пространстве; для соединения терминалов с мэйнфреймами сетевые соединения не требовались. На рис. 1.1 изображена сеть на основе мэйнфреймов.

Примечание

Несмотря на то что многие склонны рассматривать последовательные соединения терминалов с мэйнфреймом как «сети», с точки зрения нашего обсуждения они таковыми не являются. Мы имеем дело с сетями, используемыми *и разделяемыми* мэйнфреймами – другими словами, с разделяемыми сетевыми соединениями; последовательные соединения к ним не относятся.

Распространение больших коммутируемых сетей в 60-х и 70-х годах ограничивалось организациями, которые могли позволить себе содержать соответствующее оборудование. Это характерно для всех новых техно-

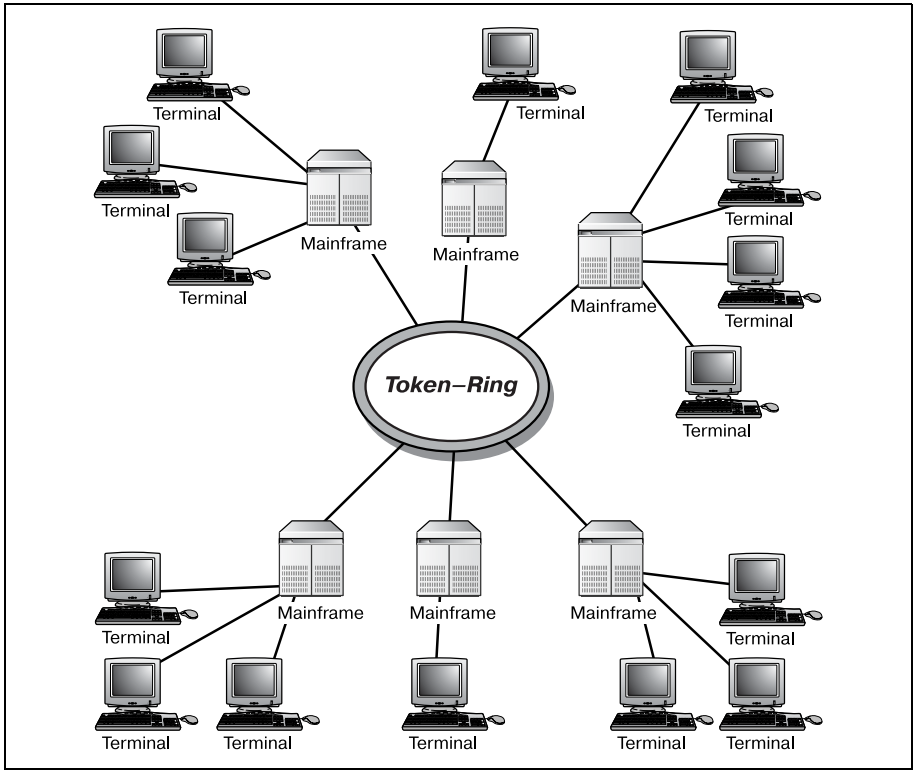

Рис. 1.1. *Сеть на основе мэйнфреймов*

логий: чем раньше вы вкладываете деньги в продукт, тем дороже он вам обходится. Например, правительственная сеть США ARPAnet (позднее превратившаяся в DARPAnet) была одной из крупнейших в мире и одной из первых. Оборудование этой знаменитой крупномасштабной сети, использовавшее прогрессивную технологию коммутации пакетов и сделавшее ARPAnet реальностью, из-за своей дороговизны так и осталось недоступным для большинства американских компаний.

В начале 1980-х началась революция персональных компьютеров (ПК). Компьютеры уменьшались в размерах и становились все более самодостаточными. Жесткие диски, вычислительные и сетевые мощности, ранее бывшие привилегией больших вычислительных машин, теперь могли уместиться на рабочем столе и обслуживать потребности единственного пользователя. Эти качества персональных компьютеров сделали их привлекательными для использования в бизнесе.

Пришествие персонального компьютера

По мере того как персональные компьютеры занимали места на рабочих столах (и в домах), администраторы начали сталкиваться с новы-

ми проблемами. Та же сеть на 60 рабочих мест, которая раньше требовала шести сетевых соединений на сравнительно небольшой площади, теперь потребовала уже 66 соединений (60 ПК и 6 мэйнфреймов) значительно большей длины.

Для эффективного использования ПК недостаточно соединить его через последовательный интерфейс с мэйнфреймом, как терминал. Каждому ПК потребуется такое же сетевое соединение, как и у мэйнфрейма. Например, если мэйнфреймы были соединены сетью Token Ring, то и всем персональным компьютерам потребуется соединение Token Ring. В результате можно столкнуться не только с физическими ограничениями технологии, такими как MAU (Media Access Units, модули подключения к среде) и длина кабелей, но и с ограничениями пропускной способности. Сеть, обслуживавшая передачу данных между шестью устройствами, вдруг столкнулась с более чем десятикратным приростом трафика. Этот скачок был наиболее заметен в сетях Token Ring, имевших к тому времени пропускную способность 4 Мбит/с.

Персональные компьютеры требуют использования того же протокола и системы адресации, что и их старшие братья – мэйнфреймы. То есть каждый ПК, присутствующий в сети, требует собственного адресного пространства (в соответствии с используемым протоколом). Однако не это оказалось самой сложной проблемой для администраторов.

Следующая трудность, с которой столкнулись сетевые администраторы и которая потребовала применения маршрутизаторов, была вызвана низкой стоимостью ПК. Новые персональные компьютеры, готовые к работе в сети, оказались намного дешевле своих больших предшественников. В результате компании начали наращивать количество ПК в своих сетях. Администраторы столкнулись с опасностью неограниченного роста числа подключаемых к сети устройств.

Наиболее популярные в то время сетевые топологии, Token Ring и Ethernet, были разработаны еще до появления ПК. Дополнительная нагрузка на сеть, которую создали персональные компьютеры, помогла определить границы применимости существовавших сетевых решений. На рис. 1.2 изображена сеть, достигшая своего предела.

Надо было что-то делать со все возрастающей сетевой нагрузкой. Существовавшие ранее проекты сетей уже не годились. Небольшие, ориентированные на подразделения сети пришли на смену большим и открытым сетям. Единая сетевая среда должна была разделиться (на логическом уровне) на меньшие и более простые в управлении подсети. Рисунок 1.3 иллюстрирует структуру сегментированной сети.

Если вы знакомы с проектированием сетей, то уже знаете, что разделение большой сети на более мелкие подсети дает ряд преимуществ. Уменьшение размеров сети позволяет избавиться от перегрузок и упрощает управление ею. Однако такие маленькие, легко управляемые сети создают другую, более серьезную проблему. Как пользователи од-

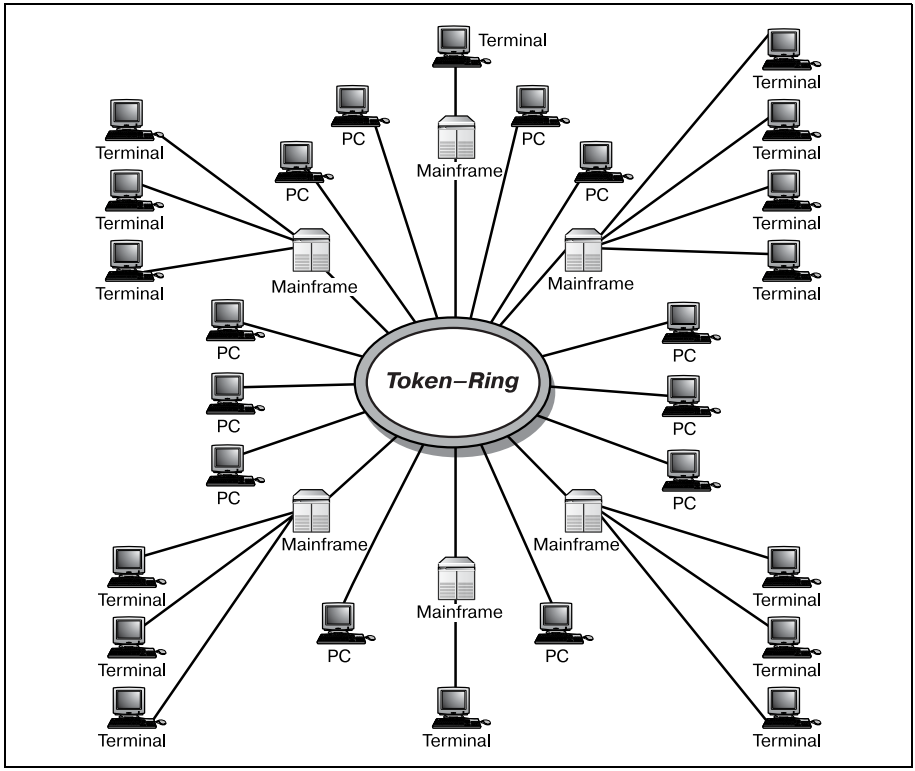

Рис. 1.2. *Большая вычислительная сеть*

ного сегмента сети смогут получить доступ к ресурсам, расположенным в другом сегменте? Ответом стала маршрутизация.

Практически без привлечения стороннего финансирования Лернер и Босак на собственные средства провели исследования и разработку одного из возможных решений задачи маршрутизации. Процесс проектирования оказался сложнее, чем виделось вначале. Но все же через два года они разработали свой первый маршрутизатор и начали прокладывать сеть в кампусе Стенфорда.

Примечание

Два сотрудника Стенфордского университета, Сэнди Лернер (Sandy Lerner), руководитель компьютерной службы факультета вычислительной техники, и Леонард Босак (Leonard Bosack), руководитель компьютерной службы Высшей школы бизнеса, начали работать над решением проблем растущей университетской сети. По ходу дела Лернер и Босак также начали работу над другой, намного более сложной задачей: проблемой маршрутизации межсистемных протоколов.

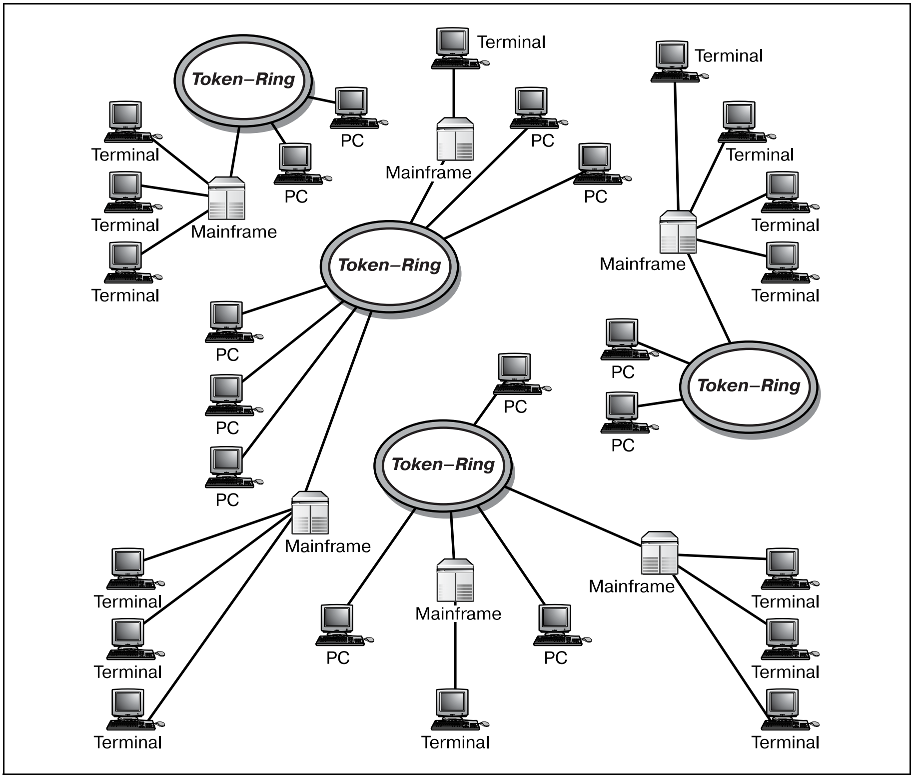

Рис. 1.3. *Сегментированная сеть*

В конце концов, их труд и новые идеи принесли результат. Они успешно объединили оборудование Стенфордского университета при помощи своих маршрутизаторов, создав одну из крупнейших на то время университетских вычислительных сетей.

Осознавая технологический (и коммерческий) потенциал своих разработок, Лернер и Босак обратились к руководству университета с предложением коммерческого использования их продукта. (В то время в Стенфорде также искали способы коммерциализации своих технологий.) Казалось бы, каждая из сторон нашла идеальное решение своих проблем: Лернер и Босак могли получить поддержку, необходимую для вывода своих маршрутизаторов на рынок, а университет принял бы участие в коммерческом проекте.

Однако университет отказался от продвижения новой технологии маршрутизации, так что Лернеру и Босаку пришлось выходить на рынок самостоятельно. Они продолжили выпуск своих маршрутизаторов и основали компанию Cisco.

Рождение Cisco

Компания Cisco представила свой первый маршрутизатор широкой публике в октябре 1984 года. Лернер и Босак продолжали финансировать компанию из частных источников и доходов от продаж. Но поставки продукта не поспевали за растущим спросом. Начиналась эра сетевых вычислений. Компании, которые не интересовались технологией коммутации пакетов (либо не могли ее себе позволить), начали присматриваться к маршрутизаторам Cisco. Объединение сетей любых размеров выполнялось сравнительно легко. Закладывался фундамент Интернета (такого, каким мы его знаем сейчас).

Чтобы удовлетворить высокий спрос на маршрутизаторы, Лернер и Босак обратились к группе венчурных предпринимателей с предложением о финансировании, необходимом для роста компании в избранном направлении. Проблема в том, что многие (если не все) венчурные предприниматели стремятся войти в правление проинвестированной ими компании, чтобы сохранить контроль над расходованием денег. В случае с Cisco венчурные предприниматели в конце концов прибрали компанию к рукам, и Лернер и Босак ушли из созданной ими фирмы.

В 1991 году Джон Чамберс (John Chambers) из компании Wang занял в Cisco пост главного вице-президента, а в 1995 получил статус президента и CEO. Под управлением Чамберса годовой доход Cisco вырос до 20 миллиардов долларов. Несмотря на то что компания изменилась с тех времен, когда Сэнди Лернер и Леонард Босак основали Cisco в Стенфордском университете, цель осталась прежней: выпуск лучшего в мире оборудования для маршрутизации.

Маршрутизаторы Cisco единодушно признаны лучшими в индустрии. На сегодняшний день им нет равных по сочетанию мощности, надежности и масштабируемости. Нигде эти характеристики не проявляются так хорошо, как в изменчивой среде Интернета.

Маршрутизация и Интернет

Одной из наиболее важных технологических новинок последнего времени стал Интернет. То, что начиналось как один из способов связи между военными учреждениями США, превратилось в индустрию с многомиллиардным годовым оборотом. Теперь Интернет используется везде, начиная с распространения новостей и другой информации, торговли и игр и заканчивая телевещанием и телефонией. Несомненно, Интернет стал неотъемлемой частью современного общества.

Первая инкарнация Интернета имела мало общего с современным киберпространством. В младенческие годы Интернет был совершенно другим. В начале 1980-х годов, когда в магазинах появились первые персональные компьютеры, по всей стране стали возникать небольшие сети серверов электронных досок объявлений (BBS). Эти серверы

обычно располагались в крупных университетских городках, то есть там, где была доступна соответствующая технология и где пользователи могли обращаться к ним из дома при помощи модема.

Пользователи BBS могли позвонить на сервер и получить доступ к информации, новостям, чатам и службам сообщений (предшественникам электронной почты). Эти небольшие (по сегодняшним меркам) сообщества пользователей, соединявшихся по модему с большими BBS, стали той структурой, из которой впоследствии возник Интернет. Однако чудо электронной почты, ставшее началом массового компьютерного взаимодействия, было еще далеко от совершенства.

Главным недостатком систем BBS была их разобщенность. На протяжении более чем пяти лет (целая жизнь в мире технологий) пользователям всего мира были доступны только радости набора телефонного номера их локальной университетской BBS и «разговора» с незнакомцем из соседнего округа. Люди общались друг с другом, но те, кто не знал, какой номер следует набирать и как войти в систему, не могли вырваться из среды их каждодневного общения.

Со временем, по мере появления продуктов маршрутизации Cisco как в коммерческих организациях, так и в университетах, различные системы BBS начали общаться друг с другом. Теперь человек, звонящий на BBS Массачусетского технологического института (МТИ), мог не только узнать температуру газировки в научном корпусе, но и установить связь с университетом Вашингтона и получать оттуда последние геосейсмические данные.

80-е годы Cisco провела за стратегическим закладыванием основ Интернета. Именно маршрутизаторы Cisco сделали возможной и простой «бесшовную» стыковку сотен различных систем, начиная с самых первых BBS. Сотни невидимых пользователям BBS маршрутизаторов Cisco передавали сигналы МТИ в Университет Вашингтона. Рассвет Интернета был уже совсем близок.

В конце восьмидесятых годов Университет Миннесоты начал работу над системой Gopher. Являясь расширением таких технологий, как Lynx и Trumpet, она должна была дать пользователям BBS возможность просматривать графику на тех компьютерах, где когда-то доминировал текст. Фактически произошло рождение Интернета. Ради возможности побродить по Интернету люди тысячами подписывались на такие онлайн-службы, как Genie, Prodigy и CompuServe. Но никто из них не знал (или не интересовался), как информация из Интернета попадала на их экран. А все это стало возможным благодаря маршрутизации.

На заре появления Интернета функция маршрутизатора была ясно и четко определена. Маршрутизаторы были необходимы для эффективного перемещения данных из одной системы в другую. Ранние маршрутизаторы работали во многом так же, как и нынешние. С технической точки зрения они проверяют пакеты, рассчитывают пути и принимают «разумные» решения в отношении маршрутизации. Эти процессы поз-

воляют маршрутизатору определить, откуда поступают данные, куда они направляются и как доставить их по назначению. Будет справедливо сказать, что существование Интернета в том виде, который доступен сегодня, было бы невозможно без маршрутизаторов и их способностей.

Отличие же маршрутизаторов Cisco, имеющихся на рынке в настоящее время, от их предшественников состоит в богатом наборе функций, накопленных ими за прошедшие годы. Изменялась технология, вслед за ней изменялись и маршрутизаторы. Многие нынешние усовершенствования и достижения в технологии маршрутизации исходят от Cisco. Другие производители быстро копируют многие элементы и функции, встроенные в оборудование маршрутизации Cisco, что подтверждает лидерство Cisco в данной отрасли информационных технологий.

Современные маршрутизаторы Cisco умеют транслировать внешние IP-адреса, выполнять функции межсетевых экранов, или брандмауэров, предоставлять удаленный доступ к сети, отфильтровывать лишний трафик и многое другое. Но основная обязанность маршрутизатора не изменилась за прошедшие годы – это рациональная передача данных из одной системы в другую.

Давайте посмотрим на то, как маршрутизатор Cisco работает в Интернете. На рис. 1.4 представлен пользователь, просматривающий вебсайт в Интернете.

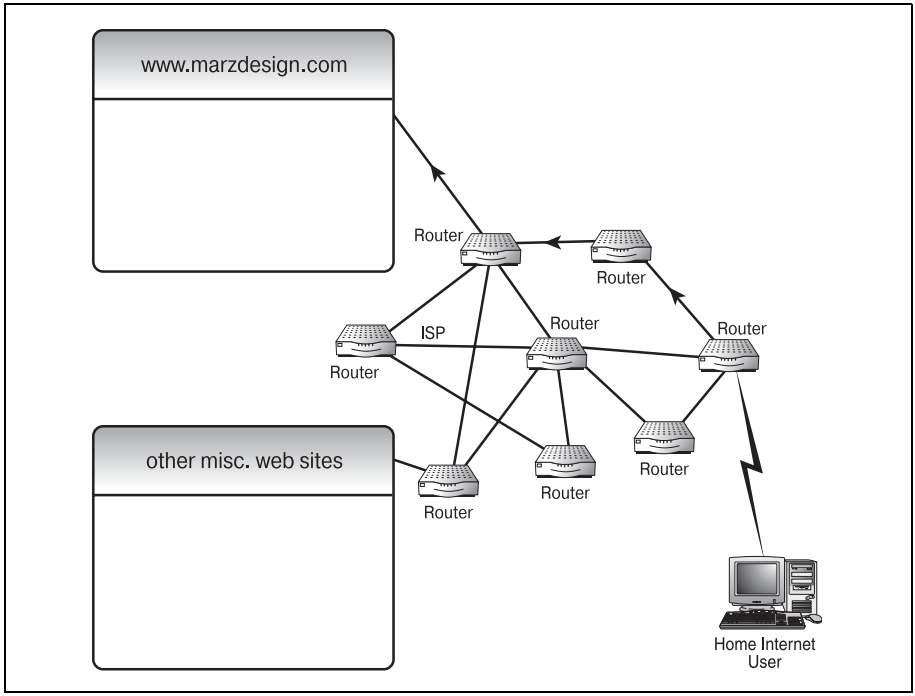

Рис. 1.4. *Процесс маршрутизации в Интернете*

Впервые пользователь может столкнуться с маршрутизатором Cisco в процессе регистрации, особенно если соединение с Интернетом осуществляется по широкополосному каналу.

Примечание

Маршрутизаторы Cisco умеют работать с интерфейсами многих типов – от WAN-интерфейсов, таких как ISDN (Integrated Services Data Network, цифровая сеть с комплексными услугами), DSL (Digital Subscriber Lines, цифровые абонентские линии) и T-каналы, до LAN-соединений, таких как Ethernet; один маршрутизатор Cisco может управлять несколькими соединениями в разных средах. Поэтому вполне возможно, что первой точкой соприкосновения с сетью провайдера будет маршрутизатор Cisco, выполняющий функцию межсетевого экрана.

В зависимости от установок окружения интернет-провайдера межсетевой экран может быть расположен перед сетевым устройством аутентификации, после него или и там, и там.

Для большинства интернет-провайдеров маршрутизатор Cisco можно поместить на внешний край сети. Он будет первой линией обороны в системе обеспечения безопасности. Любой маршрутизатор, расположенный на границе сети, подобно рассматриваемому (пользователи могут подсоединяться через него), скорее всего, будет отправлять все данные с межсетевого экрана к NAT (Network Address Translation, трансляция сетевых адресов).

Примечание

Обычно, говоря о «граничном маршрутизаторе», имеют в виду маршрутизатор, который использует BGP (Border Gateway Protocol – протокол граничного шлюза). Однако в этом сценарии пользователь услуг провайдера не имеет возможности подсоединиться к сети через граничный маршрутизатор BGP.

В данном примере на маршрутизаторе, находящемся на границе сети провайдера, работает межсетевой экран, защищающий устройство аутентификации сети (Network Authentication Device). После того как пользователь идентифицирован в сети провайдера, он может начать бродить по Интернету.

Примечание

Приведенное ниже описание работы маршрутизаторов в Интернете является упрощенным. Дочитав книгу до конца, вы получите полное представление о данном вопросе.

Предположим, что пользователь хочет посетить сайт *www.marzdesign.com*. Броузер пользователя оповещает провайдера интернет-услуг об извлечении информации с IP-адреса 207.217.96.36, соответствующего доменному имени *www.marzdesign.com*. Запрос перемещается с одного маршрутизатора провайдера на другой. Каждый маршрутизатор, исследуя запрос, определяет, какой IP-адрес запрошен и кто яв-

ляется инициатором запроса, затем сопоставляет эти данные со своей собственной *таблицей маршрутов*. Таблица маршрутов указывает, где искать запрашиваемый IP-адрес, или же (если таблица маршрутов не знает, где найти такой адрес) сообщает адрес другого маршрутизатора, который может это знать (своего рода соглашение типа «знакомый моего знакомого»). Процесс продолжается до тех пор, пока запрос не достигнет *www.marzdesign.com*. Тогда таблицы маршрутов (которые следили за тем, кто сделал запрос) посылают необходимую информацию обратно пользователю. Этот сценарий повторяется миллионы раз в день, а пользователи в большинстве случаев даже не знают об этом.

Маршрутизация в повседневной жизни

К середине 1990-х годов состав оборудования, формирующего технологическую основу Интернета, в целом определился. Большая часть аппаратных средств маршрутизации, приводящих в действие Интернет, принадлежит семейству Cisco. Их маршрутизаторы, коммутаторы и другие продукты, обеспечивающие возможность соединения, делают Интернет более надежным, более масштабируемым и более быстрым, чем когда бы то ни было ранее.

Как же способность Cisco приводить в действие Интернет реализуется в повседневной жизни? Вскоре после того как Cisco начала искать новые пути по организации связи между компаниями, последние начали думать о расширении и разветвлении. Многие предприятия стали экспериментировать, разрешив сотрудникам выполнять свои служебные обязанности, находясь дома.

Чтобы соответствовать современным технологическим требованиям бизнеса, Cisco в середине девяностых выпустила линию оборудования SOHO (small office/home office – малый офис/домашний офис). Линия маршрутизаторов SOHO обеспечивала полную функциональность предприятия в гораздо более мелком масштабе. Такие возможности, как списки доступа, каналы WAN и межсетевые экраны, которые обычно реализуются в старших моделях оборудования, теперь могут быть приобретены домашним пользователем. Маршрутизатор Cisco может обеспечить необходимую маршрутизацию и в случае домашней сети с интернет-доступом для нескольких компьютеров, и если дом используется в качестве вспомогательного офиса.

Во многих компаниях есть сотрудники, которые некоторое время работают дома. Существует несколько способов обеспечить такой процесс, и в любом из них можно использовать маршрутизаторы Cisco. Можно соединить удаленный офис с главным с помощью канала WAN. Характерный домашний офис представлен на рис. 1.5.

Домашний офис связан с главным офисом посредством ISDN. Больший маршрутизатор (например, серии 7500) расположен в главном офисе. Модем ISDN установлен как встроенное периферийное устройство на

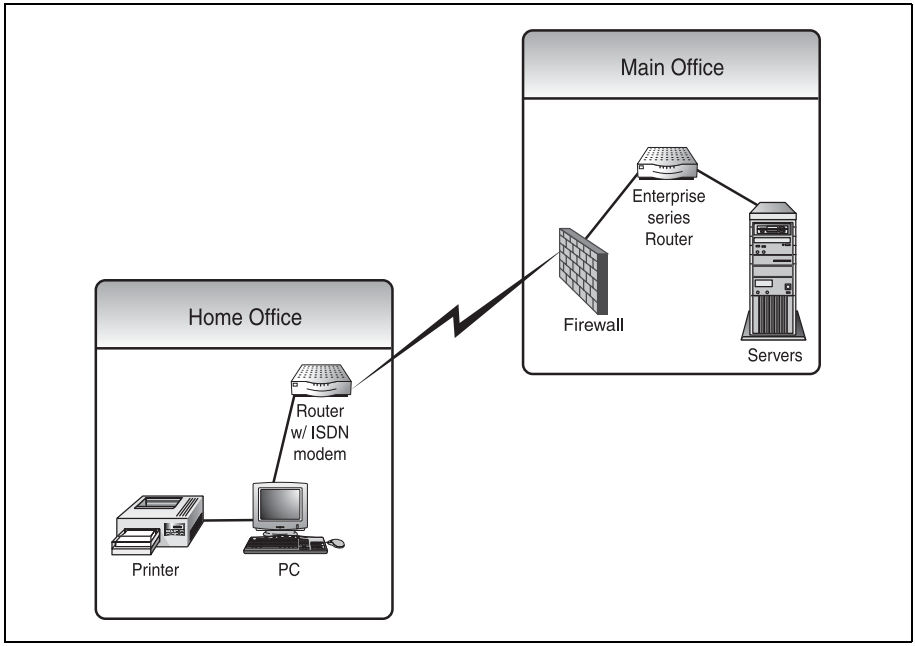

Рис. 1.5. *Типичный домашний офис*

большом маршрутизаторе, а маленький маршрутизатор SOHO (скажем, Cisco серии 1600) находится дома у пользователя и снабжен другим периферийным устройством ISDN, принимающим второй конец ISDN-соединения. Теперь два удаленных друг от друга маршрутизатора соединены таким образом, что могут взаимодействовать так, как если бы находились в одном здании.

Примечание

Cisco проделала очень большую работу для гарантии того, что ее «домашние» продукты во всех отношениях так же соответствуют существующим технологиям, как и продукты уровня предприятия. Большая часть не только работает с тем же программным обеспечением (Cisco IOS), но и может использовать одинаковые аппаратные модули, такие как энергозависимая и энергонезависимая память и платы расширения.

Несмотря на то что WAN-технологии, такие как ISDN и Frame Relay, являются общепринятым решением для удаленного доступа, существует и более популярный вариант. Для предприятий с большим количеством сотрудников, работающих не выходя из дома, WAN-технологии могут быть слишком дорогостоящими. Поэтому многие компании предпочитают технологию VPN (Virtual Private Network, виртуальная частная сеть).

На рис. 1.6 изображена сеть «дом–работа», реализующая VPN-технологию. Если вы приглядитесь, то заметите, что отличие данной сети от сети, изображенной на рис. 1.5, невелико. В сети на рис. 1.6 нет домашнего устройства маршрутизации. Для установления соединения с Интернетом пользователю не требуется иметь дома устройство, подобное Cisco серии 1600.

Однако если посмотреть со стороны компании, можно заметить и более интересную особенность этой конкретной сети. Тот же самый маршрутизатор Cisco серии 7500, который использовался для удаленного доступа, может использоваться и для организации виртуальной частной сети. Фактически один маршрутизатор может одновременно обеспечивать работу обоих сценариев. Следовательно, один маршрутизатор Cisco может обслуживать распределенную сеть для практически любого вида бизнеса. Едва ли можно проектировать сеть, работать в сети или же каким-либо другим образом взаимодействовать с сетью, не столкнувшись с маршрутизатором Cisco. Компания занимает сильную позицию на рынке и имеет хорошую репутацию благодаря производству продукта маршрутизации высшего качества.

Теперь, после того как вы приняли решение освоить маршрутизацию Cisco и познакомились с историей продукта, давайте поговорим об оборудовании, обеспечивающем всю эту работу. В следующей главе будет представлено аппаратное обеспечение, применяемое для маршрутизации.

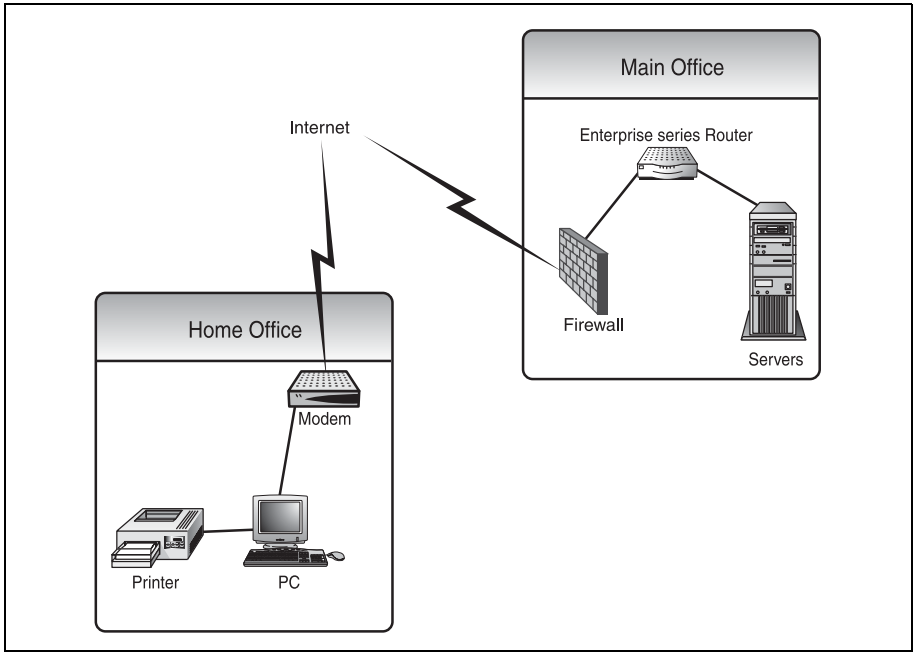

Рис. 1.6. *Виртуальная частная сеть «дом–работа»*

Резюме

Итак, вы получили базовые знания, необходимые для полного понимания истоков межкомпьютерной маршрутизации. Представленные в этой главе идеи, такие как использование протоколов и отличия между ПК и специальным оборудованием для маршрутизации, будут становиться более очевидными по мере прочтения книги.

- Долгие годы единственной возможностью изучить маршрутизацию Cisco было терпеливое посещение занятий, готовящих к получению Cisco Career Certification. Однако по стандартам самой компании Cisco не следует пытаться получить сертификат, если вы не обладаете хотя бы двухгодичным опытом в области маршрутизации.
- Прочитав книгу до конца, вы овладеете навыками, необходимыми для того, чтобы ворваться в этот удивительный мир маршрутизации.
- Cisco появился в результате технологического «эксперимента» в Стенфордском университете.
- После того как университетские чиновники приняли решение не заниматься коммерциализацией новой технологии маршрутизации, Cisco стала продавать свои маршрутизаторы частным компаниям.
- Cisco является мировым лидером в области маршрутизации, и сегодня маршрутизаторы этой компании пропускают через себя большую часть трафика Интернета.

Вопросы-ответы

Вопрос Если сети существовали и до возникновения маршрутизаторов, как же информация попадала из сети в сеть?

Ответ Основным способом организации взаимодействия сетей было использование коммутаторов. Технология коммутации уже применялась в телефонных сетях общего пользования. Кроме того, протоколы маршрутизации были реализованы на компьютерах, что позволяло таким устройствам работать в качестве маршрутизаторов.

Тест

Вопросы

1. Каково назначение маршрутизатора?
2. Где зародились маршрутизаторы Cisco?

Ответы

1. Перемещение данных из одной сети в другую.
2. В Стенфордском университете.

2

Введение в аппаратную часть Cisco

В этой главе мы рассмотрим оборудование, на котором работают маршрутизаторы Cisco. Прежде чем приступить непосредственно к изучению маршрутизации, вам необходимо познакомиться с аппаратной частью маршрутизаторов. Состав оборудования, в частности объем памяти, возможность расширения, конфигурация портов, может различаться у разных производителей и даже в пределах линейки продуктов одного производителя. Поэтому прежде всего рассмотрим разнообразное аппаратное обеспечение, предлагаемое компанией Cisco.

Примечание

Имейте в виду, что программное обеспечение, работающее в маршрутизаторах Cisco – Cisco IOS, – одинаково для всех моделей. Для некоторых серий существуют дополнительные возможности, но базовая IOS остается той же. Таким образом, различие между маршрутизаторами Cisco определяется в основном их аппаратной частью. Исключением из этого правила стали маршрутизаторы серии 700. Они работают под управлением модифицированной IOS, известной под названием IOS-700. Только эти маршрутизаторы, имеющие фиксированную конфигурацию, используют модифицированную операционную систему.

Маршрутизаторы Cisco объединены в серии. Аналогично тому как два автомобиля разных серий одного производителя отличаются друг от друга, например Ford Explorer и Ford Expedition, маршрутизаторы Cisco имеют различия между сериями. Различия в оборудовании могут быть небольшими (например, дополнительный порт Ethernet) или весьма существенными (возможность добавления десятков портов в слоты расширения). В любом случае вам необходимо иметь представление об этих различиях, чтобы понимать, с каким из маршрутизаторов вы имеете дело. Например, не стоит пытаться сконфигурировать ISDN-порт, если он отсутствует физически.

Примечание

В этой главе дан обзор аппаратного обеспечения, которое вы встретите, работая с маршрутизаторами Cisco. Мы не станем углубляться в детали устройства маршрутизаторов. Познакомимся только с общими характеристиками широкой гаммы оборудования, применяемого в маршрутизации.

При рассмотрении маршрутизаторов Cisco можно выделить две категории оборудования: оборудование общего назначения и оборудование, специфичное для данной серии. Оборудование общего назначения включает в себя элементы, используемые во всех маршрутизаторах, такие как оперативная память (RAM), порты и флэш-память. Специфичное для серии оборудование включает элементы шасси, конфигурацию и расположение портов и модулей расширения.

Выбор определенной модели маршрутизатора для сети, возможно, является одним из наиболее важных решений проектировщика (по крайней мере, с точки зрения последствий). Знание характеристик оборудования каждой из серий Cisco и конкретных моделей поможет администратору выбрать правильный маршрутизатор для любой задачи.

Две темы, рассмотренные в этой главе, помогут в выборе решения:

- Компоненты общего назначения в маршрутизаторах Cisco.
- Оборудование, специфичное для серий.

Изучая приведенные ниже спецификации оборудования, обращайте внимание на возможности и ограничения определенных моделей. Сравнение этих спецификаций с требованиями, определяемыми условиями эксплуатации, поможет правильно выбрать устройство.

Компоненты общего назначения маршрутизаторов Cisco

Каждый маршрутизатор Cisco содержит компоненты, общие для всех серий. Это означает, что некоторые узлы встречаются во всех устройствах независимо от того, к какой серии они принадлежат.

Такие компоненты, как флэш-память и порты, одинаковы во всех маршрутизаторах Cisco. Эти детали знакомы каждому, кто имеет опыт работы с компьютерным оборудованием (ПК или другим). Чтобы систематизировать изучение этих компонентов, будем обсуждать их по категориям. Таких категорий для компонентов общего назначения можно выделить две: внешнее оборудование и внутреннее оборудование.

Внутренние устройства общего назначения включают оперативную память и другие узлы, обычно скрытые внутри шасси. К внешним устройствам относятся порты, блоки питания и модули расширения. Да-

вайте рассмотрим внешние компоненты, общие для всех маршрутизаторов Cisco.

Внешнее оборудование

Рассуждения о характеристиках внешних устройств маршрутизаторов Cisco могут показаться очевидными, но на самом деле за простотой их использования стоят немалые усилия проектировщиков. Эти характеристики включают в себя такое расположение портов, при котором они наиболее доступны, и размещение маркировки маршрутизатора таким образом, что она всегда может быть легко прочитана. По характеристикам внешних компонентов можно определить, с каким типом маршрутизатора вы имеете дело. Поэтому начнем с обсуждения корпуса, в котором размещается маршрутизатор.

Корпус

Обсуждая внешние характеристики маршрутизаторов Cisco, следует начать с корпуса. Корпус – одна из наиболее характерных деталей маршрутизатора. Голубой и серый цвета (в сочетании с логотипом Cisco) говорят вам о происхождении устройства. Но помимо этого корпус имеет и другие отличительные черты, позволяющие определить серию, модель и другие сведения, определяющие производительность маршрутизатора.

Логотип Cisco и номер серии обычно выделяются на передней панели. У большинства маршрутизаторов Cisco, как у серии 2500, логотип и обозначение серии располагаются соответственно в левой и правой частях передней панели.

Примечание

Имейте в виду, что в составе серии может быть несколько моделей. Например, серия маршрутизаторов 2500 включает модели 2501, 2502, 2503, 2504, 2505, 2507, 2509 и еще 18 других. Обозначение серии на передней панели говорит только о том, к какой серии принадлежит данный маршрутизатор, а не о конкретной модели, так как различные модели одной серии могут использовать корпуса одинаковой конструкции.

Маршрутизаторы, у которых логотип расположен на передней панели, часто предназначены для монтажа в стойке (особенно это относится к старшим сериям). Устанавливаемые в стойку устройства обычно доступны только со стороны передней панели, и в большинстве ситуаций вы обнаружите, что все необходимые порты также расположены на передней панели.

У некоторых маршрутизаторов Cisco, предназначенных для малого бизнеса, таких как серия 1600, логотип и обозначение серии находятся на верхней крышке корпуса. Устройства серий для малого бизнеса обычно располагаются на столе или монтируются на стену.

Примечание

При монтаже на вертикальную поверхность маршрутизаторы Cisco обычно подвешиваются так же, как картины, то есть нижняя поверхность (дно) прикрепляется к вертикальной стойке или другой подходящей поверхности.

В большинстве маршрутизаторов Cisco, не предназначенных для монтажа в стойку, логотип и обозначение серии – единственное, что расположено на передней панели. Однако некоторые маршрутизаторы, например серии 1600, имеют панель состояния. Эта панель обычно расположена в передней или верхней части устройства. Панель состояния содержит набор светодиодных индикаторов, соответствующих различным функциям: наличие питания, целостность системы, статус порта. По индикаторам состояния вы можете судить о функционировании порта или об аномальном количестве коллизий.

Состав индикаторов зависит от типа маршрутизатора. В модели 1605 имеются две пары индикаторов для портов Ethernet (ETH0 и ETH1) и одна пара для адаптера WAN-интерфейса (WIC). В то же время, в модели 1603 используется один индикатор для порта ISDN (BRI0) и один – для адаптера WIC.

В маршрутизаторах, не имеющих индикаторов состояния на передней панели, например серии 2500, индикаторы располагаются на задней панели. Такие индикаторы расположены непосредственно у каждого порта и, как и на панели состояния 1600 серии, сигнализируют о состоянии порта. Обычно состояние порта определяется индикатором соединения и либо индикатором коллизий, либо индикатором ошибки.

Индикатор соединения порта сигнализирует о том, работает ли присоединенный к нему коммуникационный кабель. Индикаторы коллизий и ошибок показывают текущее значение уровня ошибок в канале (в сети Ethernet индицируется общий уровень коллизий в линии).

В отличие от индикаторов, которые могут располагаться и на передней, и на верхней, и на задней панелях, есть узлы, которые всегда занимают одно и то же место. Посмотрим, какие из внешних общих компонентов находятся на задней панели маршрутизатора.

Компоненты, расположенные на задней панели

Самые распространенные (и легко узнаваемые) элементы, расположенные на задней панели, – разъем и выключатель питания.

В старших моделях маршрутизаторов Cisco может использоваться питание от источника постоянного тока. Такие маршрутизаторы требуют несколько более сложной установки, чем простое подключение кабеля питания. Многие крупные компании используют возможность питания от сети постоянного тока для обеспечения резервного питания маршрутизаторов.

Сам блок питания выполнен в виде модуля, то есть блок питания переменного тока может быть вынут и заменен блоком питания постоянного тока (в тех моделях, которые это допускают).

Наряду с блоком питания, общим для всех маршрутизаторов Cisco элементом является консольный порт (типы портов могут различаться).

Консольный порт

Еще один элемент, который всегда находится на задней панели, – это консольный порт. Этот порт предназначен для первоначального установления соединения с маршрутизатором. Консольный порт выглядит, как обычный порт RJ-45. Однако если вы попытаетесь использовать стандартный кабель RJ-45, у вас ничего не получится.

Примечание

Если вы посмотрите на старшие модели маршрутизаторов (предназначенные для сетей уровня предприятия и провайдеров), то обнаружите последовательный консольный порт вместо RJ-45. Маршрутизаторы с последовательным консольным портом могут работать со стандартным терминалом, не требуя соединения с ПК. Однако не все последовательные порты, в отличие от портов RJ-45, поддерживают модемное соединение. В таких маршрутизаторах обычно предусмотрен отдельный последовательный порт для прямого модемного соединения.

Примечание

Использование консольных портов мы рассмотрим в главе 3 «Введение в Cisco IOS». Сейчас вам следует знать, что хотя консольный порт выглядит вполне невинно, он создает инженерам множество проблем. В комплекте к каждому маршрутизатору Cisco прилагается специальный консольный кабель (обычно голубого цвета) – только этот кабель можно использовать для соединения с консольным портом.

Доступ к маршрутизаторам Cisco и их администрирование могут осуществляться через различные интерфейсы. Возможна работа (при полностью сконфигурированном маршрутизаторе) через консольный порт, порт Ethernet или по протоколу Telnet через IP-порт. Однако единственный интерфейс, инициализированный по умолчанию, – консольный. Поэтому важно уметь найти консольный порт и правильно к нему подключиться.

Не будем пока вдаваться в детали, но вам следует знать, что для подключения к консольному порту ПК потребуется специальный кабель RJ-45 (несколько отличающийся как от стандартного, так и от перекрестного Ethernet-кабелей). Поскольку такие кабели трудно найти (и еще труднее правильно изготовить), всегда держите под рукой стандартный консольный кабель Cisco. Каждый маршрутизатор комплектуется по крайней мере одним таким кабелем.

Устройство специальных консольных кабелей Cisco будет рассмотрено в главе 6 «Запуск маршрутизатора с работа с ним».

Трудно переоценить важность постоянного наличия консольного кабеля Cisco. Но поверьте, в тот момент, когда он вам понадобится, поблизости не окажется ни одного. Если с маршрутизатором что-то не в порядке, консольный порт обеспечивает единственный гарантированный способ доступа к нему.

На задней панели маршрутизатора Cisco можно обнаружить еще два элемента, которые присутствуют в большинстве моделей. Это порт расширения WAN и сменный модуль флэш-памяти.

Порт расширения WAN

Порт расширения WAN, расположенный в задней части корпуса, закрыт заглушкой, сняв которую, можно получить доступ к одной из наиболее интересных возможностей маршрутизаторов Cisco. Чтобы удалить заглушку WAN-порта, просто отверните винты, которыми она закреплена.

Примечание

Некоторые маршрутизаторы Cisco поставляются с установленным в WAN-порт модулем расширения. Даже если в ваш маршрутизатор установлена плата WIC, вы можете заменить ее другой, имеющей дополнительные возможности.

Типы каналов связи, с которыми работают многие модули WAN, включая ISDN и T1, будут рассмотрены в следующих главах.

Типы интерфейсных плат WIC, которые могут быть установлены в маршрутизатор, зависят от его серии и модели (список для каждой модели приведен в разделе «Специфичное для серий оборудование» данной главы). Порт расширения WAN позволяет добавить маршрутизаторам такие элементы, как порт ISDN и интерфейс T1, дающие возможность работы с распределенными сетями.

Всегда сверяйтесь с документацией по конкретной модели маршрутизатора, чтобы удостовериться в том, что устанавливаемый модуль WIC совместим с данной моделью. Следует также убедиться в том, что ваша версия IOS поддерживает функции устанавливаемого модуля.

Расширение возможностей маршрутизатора путем установки модуля WIC – хороший способ полностью использовать заложенные в оборудование возможности. Несмотря на то что многие недооценивают возможности небольших устройств, даже такой маленький маршрутизатор, как Cisco 1605, может одновременно выполнять маршрутизацию распределенной сети (через модуль WIC) и нескольких локальных сетей (через два Ethernet-порта). Функциональность маршрутизатора может быть удвоена простым добавлением модуля WIC. Квалифицированный инженер должен знать потенциальные возможности маршрутизатора.

Сменная флэш-память предоставляет еще одну возможность модернизации, способную продлить маршрутизатору жизнь (и увеличить его производительность).

Флэш-память

Большинство маршрутизаторов Cisco комплектуются сменным (а значит, обновляемым) модулем флэш-памяти. Емкость такого модуля варьируется в зависимости от маршрутизатора (в пределах от 4 Мбайт до 24 Мбайт), но его всегда можно заменить.

Маршрутизатор использует флэш-память для хранения файлов конфигурации и образа IOS. Поэтому очень важно удостовериться в том, что все файлы сохранены, перед тем как модуль будет заменен. В противном случае при загрузке маршрутизатора можно обнаружить, что все ваши настройки пропали (не лучший вариант). Копирование файлов из флэш-памяти (и другие аспекты ее использования) мы обсудим в третьей главе при изучении Cisco IOS.

Примечание

Некоторые модели маршрутизаторов Cisco работают непосредственно с флэш-памятью, другие же работают с собственной оперативной памятью, а флэш используют только для хранения данных. В большинстве случаем это отличие отражено в названии модели. Маршрутизаторы с буквой R в конце названия модели (например, Cisco 1605 R) имеют оперативную память.

Оба типа используют флэш-память для хранения файлов.

Теперь вы уже достаточно хорошо знакомы с внешним оборудованием, общим для всех маршрутизаторов Cisco. Прежде чем приступать к рассмотрению компонентов, предназначенных для определенных серий, обсудим внутреннее оборудование маршрутизаторов.

Внутреннее оборудование

Если вы знакомы с аппаратной частью ПК, то обнаружите, что маршрутизаторы Cisco имеют с ними много общего. Все привычные компоненты – процессор, ОЗУ и другие внутренние цепи – присутствуют во всех маршрутизаторах.

В большинстве маршрутизаторов Cisco используется процессор семейства Motorola 68000 (обычно 68360 или 68030 с тактовой частотой 28 МГц). Эти процессоры очень хорошо зарекомендовали себя в сложных вычислениях, необходимых при выборе маршрута. Еще одним преимуществом современных процессоров серии 68000 является их малое энергопотребление. Благодаря низкой потребляемой мощности выделение тепла остается незначительным. Это важная характеристика, так как низкое тепловыделение позволяет некоторым маршрутизаторам (в основном ориентированным на малый бизнес) работать без охлаждающих вентиляторов. Отсутствие вентиляторов делает эти маршрутизаторы тихими и экономичными.

Большинство маршрутизаторов Cisco используют динамическое ОЗУ (DRAM) на одном модуле SIMM, устанавливаемом в слот расширения,

как и в большинстве компьютеров. Благодаря этому память заменяется так же легко, как и в ПК. (Вероятно, вам придется хотя бы раз столкнуться с заменой оперативной и флэш-памяти за время жизни маршрутизатора.)

Примечание

Динамическая память нуждается в постоянном *обновлении* содержимого, то есть в «напоминании» о своем содержании по сигналам чтения, записи или по специальному периодическому сигналу обновления. Если содержимое должным образом не обновляется, данные могут быть потеряны. Этот недостаток окупается большой емкостью и быстродействием. С другой стороны, статическая память (SRAM) не нуждается в обновлении, но имеет меньшую емкость в пересчете как на один кристалл, так и на единицу стоимости.

Динамическая память DRAM прекрасно работает в маршрутизаторах, поскольку информация в памяти маршрутизатора изменяется настолько часто, что проблем с обновлением не возникает. Кроме того, скорость работы динамической памяти дает ей дополнительное преимущество.

Мы рассмотрели все аппаратные компоненты, общие для большинства маршрутизаторов Cisco. Обратимся теперь к оборудованию, предназначенному для определенных серий.

Специфичное для серий оборудование

Каждая серия маршрутизаторов имеет предназначенное только для нее оборудование. (В противном случае не было бы необходимости в создании отдельной серии.) Серии отличаются возможностями входящего в них оборудования. Другими словами, маршрутизаторы серии, предназначенной для сетей масштаба предприятия, имеют большие возможности по управлению трафиком и расчету маршрутов, чем устройства, предназначенные для малого бизнеса.

Примечание

Третья группа маршрутизаторов, так называемые Service Provider Series, выпускается специально для нужд интернет-провайдеров (ISP). Маршрутизаторы такого высокого уровня выходят за рамки обсуждения в данной книге.

В этом разделе описано оборудование (доступные порты, их производительность, возможность обновления), специфичное для серий, используемых как в малом бизнесе, так и на крупных предприятиях. Это те маршрутизаторы, которые вы наверняка встретите в действующих сетях. Данный раздел послужит хорошим справочником для ваших будущих проектов (в сочетании с остальными знаниями, которые вы извлечете из этой книги, и вашим опытом).

Маршрутизаторы для малого и среднего бизнеса

Маршрутизаторы Cisco для малого бизнеса прекрасно подходят для фирм, не нуждающихся в многопортовых соединениях, характерных для сетей масштаба предприятия. Однако не спешите сбрасывать их со счетов только из-за того, что они не предназначены для обработки трафика, создаваемого большим предприятием.

Маршрутизаторы этого типа столь же надежны, как и их старшие братья. На практике, благодаря их низкой стоимости, многие инженеры приобретают их для собственного использования и получения практики. Такие маршрутизаторы имеют практически ту же функциональность, что и многие устройства уровня предприятия при существенно меньшей стоимости. Давайте рассмотрим наиболее популярные маршрутизаторы для малого бизнеса.

Серия 800

Маршрутизаторы серии 800 наиболее популярны в небольших, в том числе домашних, сетях. Их простая конструкция позволяет выполнять эффективную маршрутизацию между LAN- и WAN-каналами. Домашние пользователи выбирают их за простой интерфейс, в то время как для малого бизнеса они привлекательны своей универсальностью.

В этой серии наиболее популярна модель Cisco 805. Эта модель комплектуется всем необходимым для подключения к небольшой локальной сети через WAN- или PPP-соединение. Это важно для тех пользователей, которым необходимо устройство, требующее минимального (послепродажного) обслуживания и модификации.

Порт Ethernet маршрутизатора Cisco 805 имеет особенность (отсутствующую у большинства других моделей), которая делает его очень удобным для домашнего использования. Переключатель с маркировкой «HUB/NO HUB», расположенный рядом с портом, дает возможность пользователю применять обычный кабель вместо перекрестного при подключении к единственному домашнему компьютеру. Для подключения маршрутизатора к небольшой сети (другими словами, для подключения к концентратору) достаточно перевести переключатель «HUB/NO HUB» в соответствующее положение.

Последовательный порт маршрутизатора Cisco 805 используется для модемного соединения. Порт может быть установлен в асинхронный режим для работы по протоколу PPP (Point-to-Point Protocol – протокол соединения точка-точка) через модем либо в синхронный режим для работы с каналом Frame Relay через устройство CSU/DSU (channel service unit/data service unit – модуль обслуживания канала/данных). Режим работы последовательного порта устанавливается из IOS.

Характеристики модели 805 приведены в табл. 2.1.

Таблица 2.1. Характеристики Cisco 805

Компонент	Стандартное значение	Расширение
ОЗУ	8 Мбайт	16 Мбайт
Флэш-память	4 Мбайт	12 Мбайт
Процессор	Motorola 68000	нет
Порты		
Ethernet	1 (10BaseT)	нет
Последовательный	1 (синхр./асинхр.)	нет
WAN	нет	нет

Серия 1600

Маршрутизаторы Cisco серии 1600 – это превосходные маршрутизаторы начального уровня для компаний среднего размера по разумной цене, допускающие замену модулей флэш-памяти и установку дополнительных портов подключения к глобальным сетям. В серию 1600 входят пять моделей, поддерживающих различные функции. Варианты используемого в серии 1600 оборудования представлены в табл. 2.2.

Таблица 2.2. Модели серии Cisco 1600

Маршрутизатор	Ethernet-порты	Последовательные порты	Порты WAN
1601	1	1	1 (свободный слот)
1602	1	1 последовательный с интегрированным 56Кбит/с DSU/CSU	1 (свободный слот)
1603	1	0	1 ISDN и 1 свободный слот
1604	1	0	1 s-bus для линии ISDN и 1 свободный слот
1605	2	0	1 (свободный слот)

Модель 1603 популярна в небольших фирмах, использующих ISDN-соединение с удаленными офисами. Однако в этой серии наибольшую популярность заслужила модель 1605 во многом благодаря наличию двух портов Ethernet, что позволяет организовать эффективную маршрутизацию между двумя локальными и глобальной сетями. Стандартное оборудование маршрутизатора 1605 и возможности его обновления отражены в табл. 2.3. (Перечисленные порты WIC могут быть установлены в любой маршрутизатор серии 1600, имеющий соответствующий слот расширения.)

Таблица 2.3. Характеристики Cisco 1605

Компонент	Стандартное значение	Расширение
ОЗУ	8 Мбайт	24 Мбайт
Флэш-память	4 Мбайт (сменный модуль)	16 Мбайт
Процессор	Motorola 68360 33 МГц	нет
Порты		
Ethernet	2 (10BaseT)	нет
Последовательные	0	нет
WAN	Интерфейс WIC	Последовательный (синхр/асинхр.)
		56Кбит/с CSU/DSU
		Дробный T1 CSU/DSU
		ISDN BRI с интерфейсом S/T
		ISDN BRI со встроенным NT1, U-интерфейс

Серия 2500

Уже упоминавшиеся маршрутизаторы для средних предприятий серии 2500, монтируемые в стойку, могут использоваться в любой локальной сети. Многие крупные предприятия также используют серию 2500 для сегментирования больших сетей.

В эту серию входят более 20 моделей маршрутизаторов, и в этом разделе не хватило бы места для того, чтобы описать их все. Наиболее популярны модели, используемые для соединения удаленных офисов. Такое разнообразие моделей позволяет использовать серию 2500 практически в любых сетевых конфигурациях.

Один из самых популярных маршрутизаторов этой серии – модель Cisco 2520, оборудованная несколькими последовательными портами и портами Ethernet. Последовательные порты могут работать с модемами, оборудованием CSD/DSU и другими последовательными интерфейсами глобальной сети. Следует, однако, отметить, что только модели 2525 и 2524 этой серии позволяют использовать сменные модули WIC. Остальные модели серии 2500 имеют встроенные модули доступа к WAN. Характеристики маршрутизатора Cisco 2520 представлены в табл. 2.4.

Таблица 2.4. Характеристики Cisco 2520

Компонент	Стандартное значение	Расширение
ОЗУ	8 Мбайт	24 Мбайт
Флэш-память	4 Мбайт EPROM	12 Мбайт
Процессор	Motorola 68EC030 20 МГц	нет

Таблица 2.4 (продолжение)

Компонент	Стандартное значение	Расширение
Порты		
Ethernet	1 (AUI или 10BaseT)	нет
Последовательный	1 (синхр./асинхр.), 2 синхр.	нет
WAN	Встроенный	1 порт ISDN BRI

Серии маршрутизаторов масштаба предприятия

Маршрутизаторы масштаба предприятия – это основные «рабочие лошадки» Cisco. В них реализовано множество функций, выходящих за пределы рассмотрения данной книги, например, в сериях 7000 и 12000 применяются такие технологии, как VoIP (голос по IP) и сложная маршрутизация PPP.

Одна из серий уровня предприятия, заслуживающая упоминания в этой книге, – Cisco 4000. Высокотехнологичные маршрутизаторы этой серии достаточно широко распространены, так что вы можете встретить их в самых различных сетях. Причина их популярности кроется в поддержке оптоволоконных соединений по стандарту FDDI.

Примечание

Стандарт FDDI (Fiber Distributed Data Interface – распределенный интерфейс передачи данных по волоконно-оптическим каналам) имеет высокую пропускную способность и использует топологию «кольца». Избыточность, возникающая благодаря использованию двойного кольца, делает этот стандарт весьма привлекательным для компаний, нуждающихся в широкой полосе пропускания и высокой надежности.

Маршрутизаторы серии Cisco 4000 имеют модульное исполнение. Это означает, что в них могут использоваться сменные модули для маршрутизации. Один из наиболее популярных – это модуль FDDI-интерфейса. Характеристики оборудования серии Cisco 4000 представлены в табл.2.5.

Таблица 2.5. Характеристики серии Cisco 4000

Компонент	Стандартное значение	Расширение
ОЗУ	4 Мбайт	16 Мбайт
Флэш-память	4 Мбайт EPROM	8 Мбайт
Процессор	Motorola 68EC030 40 МГц	нет
Порты		
Ethernet	0	модуль
Последовательный	1	модуль

Компонент	Стандартное значение	Расширение
WAN	Сетевые модули	Ethernet Последовательный Token Ring FDDI

Резюме

В этой главе представлен обзор аппаратной части маршрутизаторов Cisco. Знакомство с оборудованием, на котором реализуется технология маршрутизации, поможет вам в усвоении материала последующих глав. В следующей главе рассматривается операционная система, работающая на маршрутизаторах Cisco. Знание аппаратной структуры маршрутизаторов позволит вам лучше понять решения, использованные в Cisco IOS.

Cisco IOS – это операционная система, управляющая всеми маршрутизаторами Cisco. В следующей главе дан обзор Cisco IOS и показано, как выполняются некоторые простые настройки. Мы обсудим пользовательский интерфейс системы, структуру команд и систему помощи.

Вопросы и ответы

Вопрос Влияет ли выбор модели маршрутизатора Cisco на общую производительность сети?

Ответ Безусловно. В зависимости от функциональных характеристик каждый маршрутизатор принадлежит к определенной серии. Выбор маршрутизатора, слишком мощного для вашей сети, приведет к неэффективному расходованию средств, а недостаточно производительный маршрутизатор не сможет адекватно обработать трафик.

Тест

Вопросы

1. ОЗУ какого типа обычно устанавливается в маршрутизаторы Cisco?
2. Как называется модуль, используемый для организации интерфейса с распределенными сетями?

Ответы

1. DRAM.
2. WIC (WAN Interface Card).

3

Введение в Cisco IOS

В этой главе мы рассмотрим Cisco IOS (Internetwork Operating System) – операционную систему, работающую на большинстве маршрутизаторов Cisco (за исключением серии 700). Прежде чем приступать к изучению таких вопросов, как настройка и эксплуатация маршрутизаторов Cisco, необходимо разобраться в основных элементах IOS. Поэтому в данной главе мы постараемся дать вам базовые знания, необходимые для понимания остального материала книги и успешного изучения маршрутизаторов Cisco.

IOS содержит полный набор инструментов, необходимых администратору для настройки маршрутизаторов Cisco и управления ими. Один из этих инструментов, командный интерпретатор Exec, служит основой, на которой выполняется Cisco IOS.

Командный интерпретатор является ядром IOS. Каждая введенная в Cisco IOS команда нуждается в управляющем процессе, выполняющем обработку и возвращающем результат. Этим и занимается интерпретатор Exec.

Командный интерпретатор Cisco состоит из двух модулей, или уровней. Нижний уровень, выполняющий основную интерпретацию команд, называется командным интерпретатором *пользовательского режима*. Верхний уровень – это командный *интерпретатор привилегированного режима*. Мы рассмотрим, как используются оба эти режима работы Exec.

Примечание

Если в вашем распоряжении есть маршрутизатор Cisco, на котором вы будете осваивать материал данной книги, не торопитесь начинать прямо сейчас. Эта глава служит введением в Cisco IOS. Поэтому лучше просто прочитать ее и усвоить инфор-

мацию, а не пытаться сразу воспроизводить примеры. (В конце каждой главы, начиная со следующей, будут приведены упражнения, которые позволят вам попрактиковаться на изученном материале.)

Если у вас есть маршрутизатор Cisco, но вы его еще не включали (или включали, но не поняли того, что увидели), отложите это до главы 6 «Запуск маршрутизатора и работа с ним». Все маршрутизаторы Cisco при первом включении входят в режим конфигурирования. Этот режим позволяет администратору установить начальную конфигурацию устройства. Если вы еще не знакомы с основами Cisco IOS, этот процесс может поставить вас в тупик. Именно поэтому мы рассматриваем основные принципы Cisco IOS здесь, в главе 3.

Эта глава, посвященная основным элементам Cisco IOS, тесно связана с главой 4 «Пользовательский интерфейс Cisco IOS». Вместе они составляют полное введение в операционную систему маршрутизаторов – Cisco IOS.

Основные темы, рассмотренные в этой главе:

- Что такое Cisco IOS и как ее обновить
- Командный интерпретатор Cisco IOS

Основы Cisco IOS

Как и любому компьютеру (и большинству других электронных устройств), маршрутизаторам Cisco для выполнения своих функций необходима операционная система. Сердцем каждого маршрутизатора Cisco является Cisco IOS, в состав которой входит код, необходимый для конфигурирования, управления и успешной работы маршрутизаторов Cisco в любых сетях.

Маршрутизаторы, как и ПК, представляют собой сложные устройства, выполняющие интенсивные вычисления. Маршрутизаторы Cisco применяют сложные формулы к наборам критериев (представляющих собой маршруты и параметры), находя, в конце концов, желаемый результат в виде «наилучшего пути» для прохождения информации. Такие вычисления требуют от операционной системы одновременно надежности и «дружественности» по отношению к пользователю. В то же время, эта операционная система должна быть компактной и быстрой, чтобы справляться с нагрузкой в современных сетях.

Благодаря хорошо продуманной структуре Cisco IOS сочетает в себе устойчивость и функциональную полноту. Система, работающая на всех маршрутизаторах Cisco, содержит функции поддержки самых сложных конфигураций, благодаря чему маршрутизаторы могут работать практически во всех мыслимых ситуациях. При этом в ней нет избыточных или «редко используемых» функций. IOS – это очень рационально собранный пакет команд и функций, предназначенных для управления маршрутизаторами.

Вам предстоит узнать, что Cisco IOS – больше, чем набор команд, выполняемых маршрутизаторами. Система предоставляет средства хранения файлов, управления памятью и другие услуги, необходимые администратору для работы в сетевой среде.

Вы можете подумать, что с таким обширным набором команд (и при такой сложности выполняемых задач) Cisco IOS должна быть сложной и трудной в использовании. На самом деле Cisco IOS чрезвычайно «дружелюбна» и легка в изучении. Каждый, кто знаком с основами работы и терминологией ПК, легко сможет изучить эту систему.

Однако прежде чем погрузиться в изучение Cisco IOS, обратимся к таким важным для операционной системы вопросам, как ее приобретение и установка.

Выяснив это, займемся техническими вопросами функционирования Cisco IOS. К концу главы вы познакомитесь с общей архитектурой этой операционной системы.

Получение обновлений IOS

Операционным системам, независимо от платформы, для достижения успеха нужна определенная гибкость. То есть чем лучше вы как администратор можете контролировать операционную систему, тем большую отдачу сможете от нее получить. Частью понятия гибкости является способность замены или модификации.

Сегодня практически каждая операционная система, присутствующая на рынке, может считаться гибкой. Если, например, у вас есть ПК, вы можете установить на него Microsoft Windows 2000, затем заменить ее на Microsoft Windows XP либо продолжать использовать, устанавливая пакеты обновления.

Система Cisco IOS настолько же гибка. Вы можете установить на маршрутизатор IOS версии 12.0(3), обновить ее до версии 12.2 или продолжать пользоваться прежней версией, устанавливая функциональные пакеты.

Примечание

Функциональный пакет Cisco IOS реализует функции, не включенные в базовую версию системы, такие как межсетевой экран или поддержка виртуальных частных сетей (VPN).

Каждый маршрутизатор поставляется (если это не оговорено особо) с базовой версией Cisco IOS. В базовом варианте IOS обычно поддерживает маршрутизацию одного протокола, такого как IP или IPX. Установка функционального пакета для IP/IPX позволит ему обрабатывать оба протокола.

В отличие от пакетов обновления большинства операционных систем для ПК, функциональные пакеты Cisco IOS полностью включают в се-

бя операционную систему, а не только модули, добавляющие новую функциональность. Поэтому, устанавливая на свой маршрутизатор функциональный пакет, необходимо убедиться в том, что он поддерживает все те возможности, которые вы уже используете.

Предположим, например, что маршрутизатор работает в базовой версии IOS с IP-маршрутизацией, а администратор решает добавить маршрутизацию IPX и с этой целью устанавливает функциональный пакет Cisco IPX IOS. Ошибка этого администратора заключается в том, что при установке нового пакета его функции не объединяются с уже имеющимися. В результате таких действий имеющийся экземпляр IOS с поддержкой IP-маршрутизации буден заменен на другой, поддерживающий маршрутизацию IPX (и только IPX), а поддержка IP будет утеряна. Для работы с обоими протоколами администратору необходимо установить функциональный пакет Cisco IP/IPX IOS. Поэтому будьте внимательны, выбирая пакет обновления для маршрутизатора.

После того как решение об обновлении Cisco IOS принято, необходимо получить требуемое программное обеспечение.

В отличие от многих других операционных систем, Cisco IOS может быть приобретена только у компании Cisco – просто потому, что она может работать только на оборудовании Cisco. Маршрутизаторы Cisco продаются с предустановленной IOS последней версии. Этого вполне достаточно для работы маршрутизатора. Однако благодаря надежности оборудования Cisco и приверженности пользователей многие желают обновлять имеющуюся у них IOS по мере выпуска новых версий. Кроме того, потребности сети могут меняться со временем, вызывая необходимость установки новых функциональных пакетов. Где же найти эти обновления?

Рекомендуемый способ получения «самого свежего и лучшего», предлагаемого Cisco, заключается в регистрации вашего маршрутизатора и заключении сервисного соглашения. В этой книге не рассматриваются детали различных предлагаемых Cisco сервисных соглашений, но большинство из них включают доступ к защищенному веб-сайту Cisco, показанному на рис. 3.1.

На защищенном сайте Cisco вы можете бесплатно получить (в соответствии с вашим планом обслуживания) последние обновления Cisco IOS и функциональные пакеты. Доступны образы IOS (файлы, из которых устанавливается операционная система) для текущей версии и для большинства предшествующих. Следовательно, администраторы, обслуживающие более старое оборудование, тоже могут получить IOS, необходимую для поддержания сетей в рабочем состоянии.

Помимо новых версий IOS на сайте CCO (Cisco Connection Online) можно найти множество технической документации.

Получение Cisco IOS – это только первый шаг. Следующим шагом станет обновление системы (или, в зависимости от ситуации, установка функционального пакета).

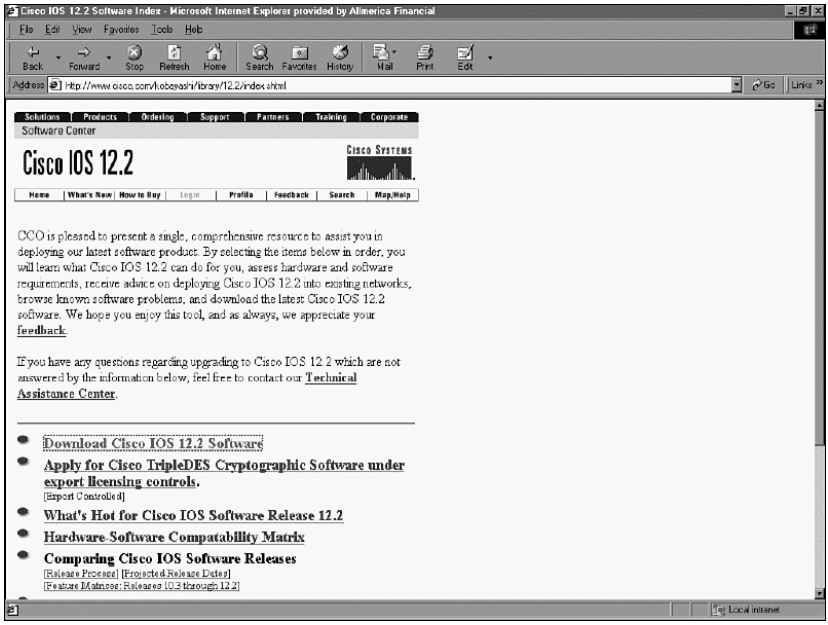

Рис. 3.1. *Защищенная страница Cisco для загрузки обновлений IOS*

Организация памяти маршрутизатора

Прежде чем обновлять IOS, необходимо разобраться с устройством памяти маршрутизатора. На разных моделях маршрутизаторов IOS выполняется в разных местах. Знание того, откуда запускается IOS, позволит успешно осуществить обновление, не повредив при этом образ системы.

По способу запуска операционной системы маршрутизаторы Cisco делятся на две категории. Устройства, загружающие IOS в ОЗУ, относятся к типу RFR (Run-from-RAM), а устройства, исполняющие IOS непосредственно во флэш-памяти, – к типу RFF (Run-from-Flash).

Примечание

Индекс R в конце номера модели означает, что маршрутизатор относится к типу RFR – например, Cisco 1605R.

Можно считать, что все маршрутизаторы без индекса R относятся к типу RFF.

Метод обновления IOS выбирается в зависимости от того, как организована память маршрутизатора. Процедура обновления системы для маршрутизаторов типа RFR не годится для устройств типа RFF. Поэтому необходимо разобраться, в чем заключается различие этих двух архитектур.

Маршрутизаторы типа RFF

Основное свойство флэш-памяти заключается в способности сохранять данные вне зависимости от того, включено или нет устройство, в котором эта память установлена. Такая энергонезависимость – способность хранить информацию при выключенном питании – главная причина, по которой Cisco использует флэш-память в качестве основного устройства хранения информации в большинстве своих маршрутизаторов.

Все файлы, конфигурации и образы IOS хранятся во флэш-памяти маршрутизатора независимо от способа организации памяти. В обоих типах, RFR и RFF, образ IOS хранится во флэше. Различие между двумя типами архитектур заключается в том, куда помещается образ IOS после загрузки маршрутизатора.

Устройства типа RFF загружаются так же, как и все остальные маршрутизаторы Cisco. Загрузчик вызывает и запускает исполняемые образы непосредственно во флэш-памяти. Рисунок 3.2 иллюстрирует процесс загрузки маршрутизатора RFF.

Загрузочный процесс исполняет распакованный образ IOS непосредственно во флэш-памяти. Образ IOS выполняется во флэш-памяти, а рабочие файлы копируются в ОЗУ.

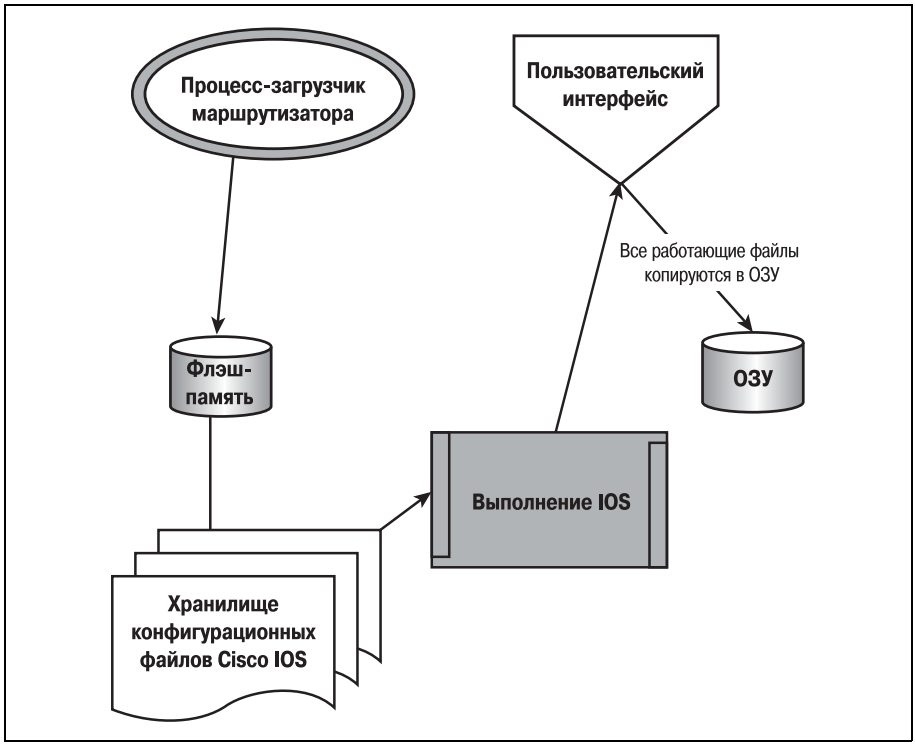

Рис. 3.2. *Загрузка образа IOS, исполняемого во флэш-памяти*

Такой подход имеет некоторые преимущества перед другими способами организации памяти. Первое заключается в быстроте загрузки. Поскольку образ IOS не нуждается в распаковке, маршрутизатор стартует быстрее. Это может быть важно в условиях, когда на счету каждая минута. (Если маршрутизатор типа RFR требует обычно от 2 до 3 минут для загрузки, то маршрутизатор RFF запускается менее чем за минуту.)

Прямая работа с флэш-памятью также позволяет освободить ОЗУ. Благодаря тому что оперативная память полностью отдана под размещение рабочих файлов IOS (текущей конфигурации, таблиц маршрутов и других необходимых в процессе работы файлов), доступ к ним значительно ускоряется. Это дает дополнительное преимущество над другими типами архитектур.

Последнее преимущество архитектуры RFF заключается в меньшем риске повреждения. Образ IOS не подвергается модификациям с той частотой, с какой это имеет место при других конфигурациях памяти.

Благодаря тому что образ IOS, хранимый во флэш-памяти, не подвергается преобразованиям на двоичном уровне (то есть не упаковывается и не распаковывается) и остается неизменным, маршрутизатор работает более стабильно и вероятность повреждения файлов значительно уменьшается.

Однако архитектура RFF имеет и свои недостатки. Главный из них состоит в том, что образ IOS выполняется в той же области памяти, где и хранится. В результате систему невозможно обновить в процессе работы. Несмотря на то что такое желание может показаться противоестественным, в нем есть смысл. Но прежде чем рассматривать процесс обновления системы, обсудим архитектуру памяти маршрутизаторов типа RFR.

Маршрутизаторы типа RFR

Другой подход к загрузке маршрутизаторов реализован в архитектуре памяти RFR. В них, как и в устройствах RFF-типа, образ IOS хранится во флэш-памяти. Однако на этом сходство заканчивается. Образ IOS хранится во флэш-памяти этих маршрутизаторов в сжатом виде – в отличие от RFF-маршрутизаторов.

Примечание

Независимо от архитектуры памяти все маршрутизаторы Cisco используют флэш-память для хранения файлов. Все образы IOS, и сжатые и несжатые, хранятся во флэше.

В процессе загрузки сжатый образ IOS извлекается из флэш-памяти и распаковывается. В результате распаковки получаются исполняемые файлы операционной системы. Процесс загрузки маршрутизатора RFR показан на рис. 3.3.

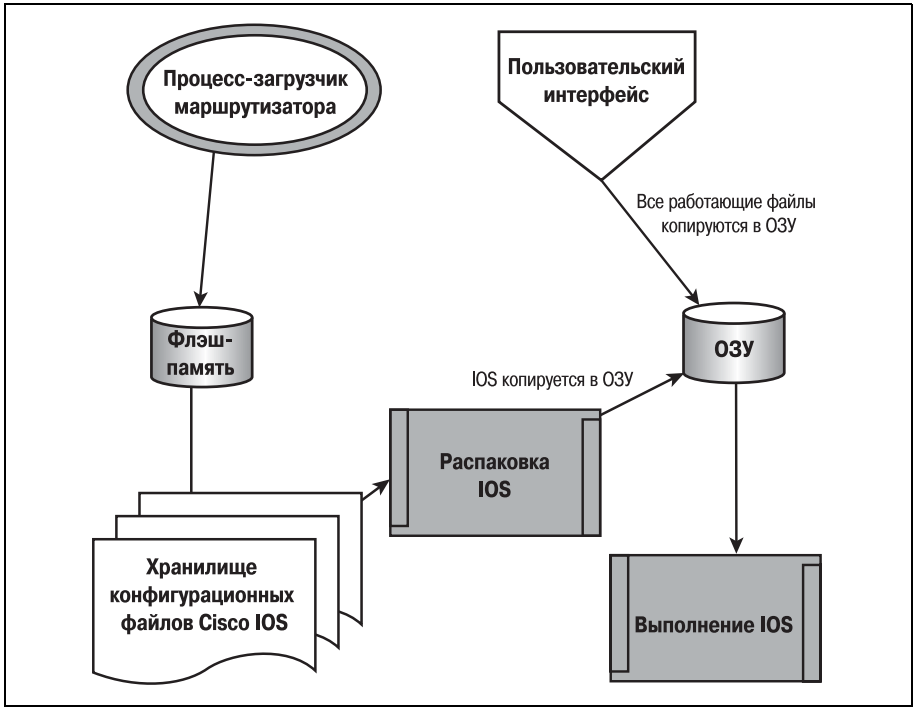

Рис. 3.3. *Загрузка образа IOS, исполняемого в ОЗУ*

Распакованный образ IOS копируется в ОЗУ маршрутизатора. Вследствие этого пространство ОЗУ должно быть поделено между рабочими файлами и операционной системой. Поэтому маршрутизаторы с памятью типа RFR обычно комплектуются ОЗУ большего размера, чем RFF-устройства.

Примечание

Вспомните вторую главу: большинство маршрутизаторов Cisco имеют ОЗУ динамического типа.

После того как образ IOS распакован и скопирован в ОЗУ, маршрутизатор начинает выполнять его. Так стартует операционная система маршрутизатора. Все стандартные операции по маршрутизации выполняются в ОЗУ, а флэш-память может использоваться для долговременного хранения данных.

Архитектура RFR, как и RFF, имеет свои преимущества. Основное из них – освобождение флэш-памяти. В силу того что IOS выполняется в ОЗУ, флэш-память остается свободной и в нее можно загрузить новую версию системы, не прерывая работу существующей. В результате обновление IOS на таких маршрутизаторах выполняется значительно проще, чем на RFF.

К недостаткам маршрутизаторов RFR можно отнести повышенный риск повреждения IOS. Поскольку над системой выполняются манипуляции значительно более сложные, чем в маршрутизаторах типа RFF, увеличивается шанс повредить систему при двоичных преобразованиях.

Поэтому в устройствах RFR очень важно поддерживать актуальное состояние резервных копий конфигурации и файлов IOS. (Методы и процедуры резервного копирования будут рассмотрены в главе 7 «Резервное копирование маршрутизаторов Cisco».)

Обновление IOS, исполняемой во флэш-памяти

Есть два основных метода обновления IOS в маршрутизаторах типа RFF. Наиболее распространенный путь заключается в использовании программы Flash Load Helper (FLH), содержащей набор команд, позволяющих получить доступ к флэш-памяти в обход IOS.

Другой путь предполагает использование двойного банка флэш-памяти. При этом используется модуль флэш-памяти (SIMM) большой емкости, который может быть разделен на две области, что позволяет администратору получить доступ к неиспользуемой части независимо от того, выполняется ли в данный момент IOS. Давайте рассмотрим процедуры, использующие оба метода.

Примечание

Не волнуйтесь, если какие-то функции IOS в приведенных ниже примерах вам покажутся непонятными. Основы пользовательского интерфейса Cisco IOS и используемые здесь функции будут более подробно рассмотрены в главе 4.

Использование двойного банка флэш-памяти

Несмотря на простоту и очевидность использования двойного банка флэш-памяти для обновления IOS этот процесс требует значительного времени на подготовку. Правда в большинстве случаев эту процедуру приходится проделывать лишь единожды.

Имейте в виду, что не все маршрутизаторы Cisco могут работать с двойным банком флэш-памяти. Чтобы определить, можно ли установить маршрутизатор в такой режим, изучите документацию либо вскройте корпус и рассмотрите SIMM-модуль флэш-памяти. Если на нем расположено более одного набора из четырех микросхем памяти, то маршрутизатор может быть настроен на использование двух банков.

Примечание

Сегмент флэш-памяти может состоять не менее чем из четырех микросхем.

Разделение флэш-памяти на сегменты позволяет создать две логические области для хранения и выборки данных. Это позволяет адми-

нистратору записывать образ IOS в один раздел, в то время как отдельный образ исполняется в другом разделе.

Преимущество такого метода в том, что загрузка нового образа IOS не требует остановки маршрутизатора, в то время как при использовании программы Flash Load Helper маршрутизатор должен быть ненадолго выведен из рабочего состояния.

После того как вы определили, что маршрутизатор допускает разделение флэш-памяти на два банка, проделайте следующие шаги.

Во-первых, войдите в привилегированный режим командного интерпретатора.

```
Router>enable
Router#
```

В привилегированном режиме войдите в программу глобальной конфигурации.

```
Router#configure
Configuring from terminal, memory, or network [terminal]? terminal
```

Вводите команды конфигурации по одной в каждой строке. В конце нажмите <Ctrl>+<Z>.

```
Router(config)#
```

Находясь в программе глобальной конфигурации, вы можете вводить команды сегментирования флэш-памяти.

```
Router(config)# partition flash
```

Эта команда принимает параметры, определяющие количество и размер создаваемых разделов.

```
Router(config)#partition flash ?
 <1-8> Number of partitions in device

Router(config)#partition flash 2 ?
 <1-64> Size of partition 1
```

После того как разделение флэш-памяти закончено, обновление IOS не вызовет затруднений. Используйте команду `copy` для копирования образа IOS в неиспользуемый раздел памяти.

Примечание

Не волнуйтесь, если вам не понятен смысл некоторых из использованных нами команд. В дальнейшем, по мере продвижения вперед, вы поймете их назначение и функционирование.

Команда `copy` применяется для копирования файлов из одного места в другое независимо от архитектуры памяти и метода обновления систе-

мы. Поэтому вам придется довольно часто ее использовать в процессе работы с маршрутизатором.

```
Router#copy tftp flash
```

Команда такого вида копирует неуказанный файл с сервера TFTP (Trivial File Transfer Protocol – простой протокол передачи файлов) во флэш-память. При вводе этой команды IOS попросит вас ввести еще несколько параметров, прежде чем начнется собственно копирование. Вы должны будете ввести следующие параметры:

- Имя TFTP-сервера
- Имя файла, содержащего образ IOS
- Раздел флэш-памяти, куда копируется файл
- Имя файла-копии

Примечание

Программу TFTP-сервера (обычно бесплатную) можно найти практически для любой операционной системы. Учтите, однако, что программы FTP – это не то же самое, что TFTP. Большинство FTP-клиентов не могут работать с TFTP.

Когда образ IOS скопирован в требуемый раздел, следует указать маршрутизатору, какой из образов он должен использовать. То есть в данный момент в вашем маршрутизаторе есть две области памяти, в каждой из которых находится пригодная для работы операционная система. Поэтому необходимо сообщить ему, какую из них следует использовать.

Чтобы указать маршрутизатору, с какого раздела флэш-памяти он должен загружаться, используйте команду `boot` в режиме глобальной конфигурации.

```
Router(config)#boot system flash 2
```

Эта команда означает, что маршрутизатор должен загружать систему из второго раздела флэш-памяти.

Теперь маршрутизатор полностью настроен для запуска новой IOS. Однако если ваш маршрутизатор типа RFF не может работать с несколькими разделами памяти, вам придется использовать Flash Load Helper.

Использование Flash Load Helper

Значительно более простым способом обновления системы в маршрутизаторах типа RFF является использование Flash Load Helper. FLH – это небольшая предустановленная утилита, автоматизирующая процесс обновления системы. Однако она может быть использована не на всех RFF-маршрутизаторах.

Чтобы воспользоваться программой Flash Load Helper, просто введите ту же самую команду, которую вы использовали для копирования

файла. Если маршрутизатор сконфигурирован для использования Flash Load Helper, то после ввода команды появится сообщение.

```
Router# copy tftp flash
**************** NOTICE ***************************
Flash load helper v1.0
This process will accept the TFTP copy options and then terminate the current
system image to use the ROM based image for the copy. Router functionality will
not be available during that time. If you are logged in via telnet, this
connection will terminate. Users with console access can see the results of
the copy operation.
*****************************************************
```

Это сообщение говорит о том, что данный маршрутизатор готов использовать Flash Load Helper, и вы можете продолжать процесс обновления IOS.

После сообщения FLH появится приглашение на ввод еще нескольких параметров. Программа спросит вас адрес TFTP-сервера, с которого следует получить образ IOS, имя файла, содержащего образ, и имя файла, в который он будет скопирован. Затем FLH попросит подтвердить указанные действия, и процесс обновления начнется.

Единственный серьезный недостаток использования FLH заключается в том, что когда процесс завершен, программа выполняет перезагрузку. Маршрутизатор должен быть перезагружен с новым образом системы (так как старый больше не существует). Поэтому при использовании Flash Load Helper возникает небольшой перерыв в работе.

Обновление IOS, исполняемой в ОЗУ

Существует два различных способа обновления Cisco IOS на маршрутизаторе типа RFR. В первом используется поставляемая Cisco программа RSL (Router Software Loader). Эта программа устанавливает соединение с маршрутизатором, определяет текущую версию IOS и позволяет обновить ее. Разумеется, для обновления IOS вам понадобится файл с новым образом системы, как было рассказано в предыдущем разделе. Рисунок 3.4 иллюстрирует использование RSL.

После успешного соединения с маршрутизатором и проверки его флэш-памяти RSL позволит скопировать существующую IOS либо установить вместо нее новую. Однако здесь неизбежно возникнет проблема: вы получите сообщение о том, что устанавливаемая IOS не является официальным релизом. Эта ошибка показана на рис. 3.5.

Причина появления этого сообщения в том, что файл, который вы пытаетесь установить на маршрутизатор, хоть и является корректным образом IOS, но имеет неправильное имя. Например, файл базовой IP-версии IOS 12.0(3) для серии Cisco 1600, загруженный с сайта, имеет имя `c1600-y-mz.120-3.bin`. Но RSL считает, что файл должен называться `aaa0269.bin`. Откуда такое несоответствие? Автору неизвестно. Так

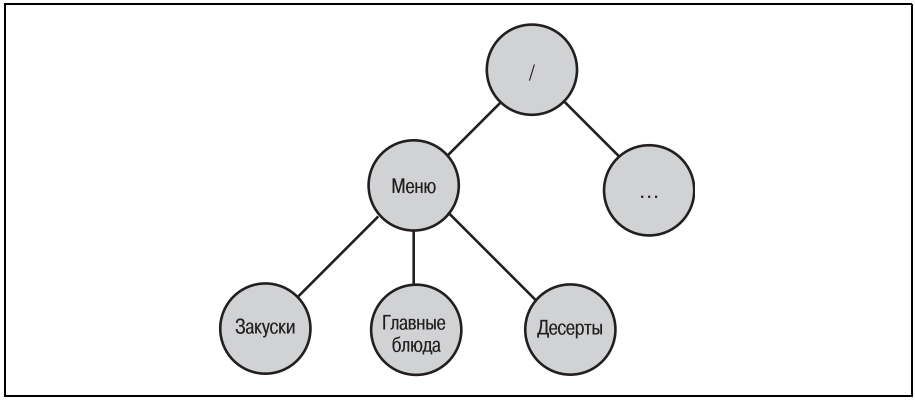

Рис. 3.4. *Программа Router Software Loader*

или иначе, многие инженеры предпочитают выполнять обновление IOS вручную.

Приведенные ниже инструкции помогут вам в обновлении Cisco IOS. Даже если большинство команд пока останется для вас непонятным, вы получите представление об обслуживании маршрутизатора. Вы сможете вернуться к этому разделу позже, когда ближе познакомитесь с функционированием маршрутизатора. (Как и при любых действиях по изменению конфигурации, рекомендуется перед началом работы сделать резервную копию содержимого флэш-памяти. Вы можете скопировать флэш-память на TFTP-сервер, выполнив команду `copy flash TFTP`.)

Для обновления IOS следует войти в привилегированный режим

```
Router>enable
```

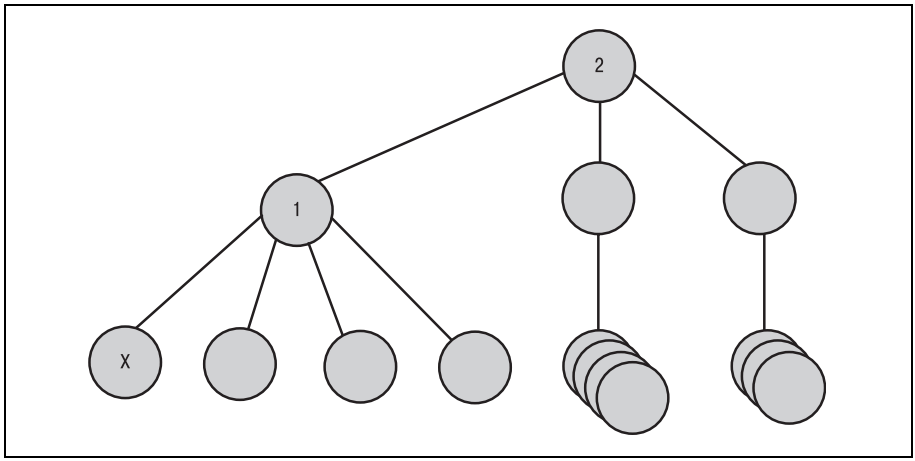

Рис. 3.5. *Ошибка RSL*

Находясь в привилегированном режиме, используйте команду `copy`, чтобы инициировать диалог. Эта команда констатирует, что вы собираетесь что-либо копировать с TFTP-сервера во флэш-память.

```
Router#copy tftp flash
```

Маршрутизатор предложит вам ввести три параметра: IP-адрес (или DNS-имя) TFTP-сервера, имя файла, содержащего образ IOS, и имя файла, в который выполняется копирование.

```
Address or name of remote host []? 10.16.4.152
Source filename []? c1600-y-mz.120-3.bin
Destination filename [c1600-y-mz.120-3.bin]?
```

Вы заметите, что имя файла-приемника (имя, которое будет иметь файл образа во флэш-памяти) по умолчанию совпадает с именем исходного файла. После начала копирования маршрутизатор спросит, следует ли удалить файл текущего образа IOS. Ответьте утвердительно и продолжайте:

```
%Warning: Run from flash images are recommended for this platform.
Erase flash: before copying? [confirm]
Erasing the flash filesystem will remove all files! Continue? [confirm]
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Erase of flash: complete
Loading c1600-y-mz.121-8a.bin from 10.16.4.152 (via Ethernet0):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
[OK - 2868332/5736448 bytes]

Verifying checksum... OK (0xD6C8)
2868332 bytes copied in 106.936 secs (27059 bytes/sec)
```

Наконец, когда новый образ IOS скопирован на маршрутизатор, для перезагрузки устройства запустите команду `reload`.

```
Router#reload
```

Когда маршрутизатор перезагрузится, у вас будет новенькая IOS «для игр». Но не думайте, что на этом все закончилось. При перезапуске после установки IOS через TFTP часто возникает проблема. Маршрутизатор может вывести сообщение о том, что образ IOS не может быть загружен. Не расстраивайтесь, маршрутизатор продолжит загрузку, а ошибку легко исправить.

Обычно в результате загрузки образа IOS с TFTP-сервера Boot Loader, загрузчик Cisco IOS, продолжает искать файлы на том же сервере. Не

обнаружив их по этому адресу, маршрутизатор просмотрит флэш-память и загрузит образ оттуда. Чтобы избежать этого, отредактируйте файл startup-config, войдя в привилегированный режим.

```
Router#configure terminal
Router (config)# no boot tftp 1: <имя файла образа>
Router (config)#^Z
Router#reload
```

Примечание

В предыдущем примере замените фрагмент *<имя файла образа>* именем того файла образа IOS, к которому пытается обратиться загрузчик.

Эти действия исключат ошибку при попытке загрузить IOS по TFTP. Есть еще одна особенность, о которой следует знать, устанавливая новую версию Cisco IOS.

Обязательно обратите внимание на требования к объему памяти устанавливаемой версии IOS. Вполне вероятно, что после копирования на маршрутизатор образа IOS не останется места для его исполнения. Имейте в виду, что команда copy не интересуется тем, что копирует, она просто перемещает двоичные данные с места на место.

Объем флэш-памяти может быть более чем достаточен для хранения образа IOS. Но если объем ОЗУ не позволит запустить систему, загрузчик зациклится. Загрузив образ, он попытается его запустить, обнаружит, что запуск не удался, и попытается повторить загрузку, в результате чего попадет в бесконечный цикл.

Эта проблема также имеет решение, хотя и несколько более сложное. Вам понадобятся следующие вещи:

- Терминальная программа на ПК, работающая по протоколу Xmodem и способная послать код разрыва соединения (break).
- Резервная копия исходного образа IOS (или другого подходящего образа). Этот образ должен находиться на ПК.
- Ангельское терпение.

Примечание

Основным условием для успешного решения проблемы является наличие терминальной программы, способной послать сигнал прерывания в линии. Многие пользуются программой HyperTerminal, поставляемой с большинством операционных систем от Microsoft. Однако HyperTerminal на Windows NT4 не может послать сигнал прерывания. Поэтому вам, возможно, придется поискать подходящий эмулятор терминала для вашей платформы.

Первым делом для решения проблемы с зацикливанием загрузчика необходимо перезапустить маршрутизатор. В момент включения пошлите ему сигнал прерывания. Этим вы переведете маршрутизатор в ре-

жим управления. Данный режим можно узнать по характерному приглашению:

```
rommon 1 >
```

В ответ на это приглашение запустите на маршрутизаторе протокол XModem:

```
rmonitor 1 >xmodem -r -s9600 aaa0269.bin
```

Эта команда означает, что вы хотите: запустить на маршрутизаторе протокол Xmodem (`xmodem`), скопировать файл образа непосредственно в динамическое ОЗУ и запустить его (`-r`), копировать со скоростью 9600 бит/с (`-s9600`) и назвать файл aaa0269.bin (`aaa0269.bin`). (Укажите здесь имя файла, посылаемого на маршрутизатор.)

Есть две причины для копирования файла в ОЗУ, а не во флэш. Во-первых, вы не можете удалить образ, находящийся во флэш-памяти, поэтому вам наверняка не хватит места для еще одного. Во-вторых, если бы это вам и удалось, маршрутизатор все равно будет пытаться загрузить старый образ. Поэтому надо копировать новый образ прямо в ОЗУ.

Когда маршрутизатор сообщит, что готов к приему файла, запустите протокол XModem на эмуляторе терминала. Перешлите на маршрутизатор резервную копию исходного образа IOS. После того как передача закончится (что займет некоторое время при использовании последовательной линии), маршрутизатор перезапустится с той версией IOS, которую вы ему отправили. Но это еще не все.

Оставшуюся часть данной главы мы посвятим введению в команды и функции IOS – эти знания пригодятся для понимания и усвоения сведений, приведенных в последующих главах. Прежде чем изучать набор команд Cisco IOS, разберемся в том, как организован пользовательский интерфейс.

Командный интерпретатор Cisco IOS

Теперь, когда мы установили (или обновили) IOS, настало время заглянуть внутрь и понять, как эта система работает. Основной компонент Cisco IOS – командный интерпретатор Exec, различные режимы работы которого описаны в данном разделе.

Как уже упоминалось во «Введении» в начале главы, Cisco IOS предоставляет два уровня доступа к структуре команд системы. Эти уровни известны как режимы командного интерпретатора, или режимы Exec. Если вы обратитесь к примерам обновления IOS из предыдущего раздела, то заметите, что команды вводятся в ответ на два различных приглашения. Эти приглашения представляют два режима, в которых находится командный интерпретатор.

Первый (и основной) известен как пользовательский режим и обозначается знаком > в приглашении.

```
Router>
```

В этом режиме пользователь может выполнить большинство основных команд, таких как просмотр характеристик маршрутизатора или временное изменение настроек терминала. Доступ к маршрутизатору в пользовательском режиме не требует пароля.

Вот наиболее часто используемые команды пользовательского режима:

- `ping`
- `rlogin`
- `telnet`
- `show`

Хотя эти команды не могут изменить глобальные настройки или внести постоянные изменения в функционирование маршрутизатора, их возможностей вполне достаточно для сбора информации и наблюдения за исправностью устройства.

Второй (более сложный) уровень доступа – это привилегированный режим Exec. В этом режиме в приглашении появляется символ `#`.

```
Router#
```

Для перехода из пользовательского режима в привилегированный используйте команду `enable`.

```
Router>enable
Password:
Router#
```

В целях обеспечения безопасности доступ к привилегированному режиму командного интерпретатора может быть закрыт паролем. (Установку пароля мы обсудим в главе 6 «Запуск маршрутизатора и работа с ним».)

В привилегированном режиме администратор может получить доступ ко всем функциям маршрутизатора. Этот режим дает ему доступ к средствам, позволяющим конфигурировать интерфейсы, соединяться с внешними источниками, загружать протоколы, перемещать и удалять файлы.

Вот некоторые из часто применяемых команд привилегированного режима:

- `configure`
- `erase`
- `setup`

В следующей главе, которая называется «Пользовательский интерфейс Cisco IOS», мы более подробно рассмотрим средства, доступные в обоих режимах командного интерпретатора. По мере изучения пользовательского интерфейса Cisco вы будете знакомиться с различными командами, применяемыми в обслуживании маршрутизатора. Прежде чем перейти к следующей главе, попробуйте ответить на некоторые вопросы по пройденному материалу.

Резюме

Интеллект, заложенный в маршрутизаторы Cisco, сосредоточен в их операционной системе. Благодаря IOS маршрутизатор способен выполнять все задачи по успешной маршрутизации данных между сетями.

Чтобы задействовать те функции маршрутизатора Cisco, которые включены по умолчанию, необходимо установить функциональные пакеты IOS, благодаря которым появляется возможность использовать дополнительные протоколы, межсетевые экраны и виртуальные частные сети.

В маршрутизаторах Cisco применяются две конфигурации памяти: RFR и RFF. Тип конфигурации памяти определяет, где исполняется IOS – в ОЗУ (RFR) или во флэш-памяти (RFF).

Вопросы и ответы

Вопрос Почему маршрутизаторы Cisco серии 700 используют собственную IOS?

Ответ Маршрутизаторы серии 700 – специализированные устройства, имеющие (более или менее) фиксированную конфигурацию. Операционная система, устанавливаемая на них по умолчанию, имеет соответствующие ограничения.

Тест

Вопросы

1. Какие две конфигурации памяти используются в маршрутизаторах Cisco?
2. Какой функциональный пакет устанавливается по умолчанию на большинство маршрутизаторов Cisco?
3. Каковы два основных режима командного интерпретатора Cisco?
4. Какой протокол используется для пересылки образа IOS на маршрутизаторы Cisco и обратно?

Ответы

1. RFR (исполнение в ОЗУ) и RFF (исполнение во флэш-памяти).
2. IP Basic.

3. Пользовательский и привилегированный.
4. TFTP.

Упражнения

1. Назовите две различные архитектуры памяти маршрутизаторов Cisco.
2. Какие методы существуют для обновления IOS в маршрутизаторах типа RFF?
3. В чем основное различие между способами хранения IOS в маршрутизаторах типа RFF и RFR?
4. Назовите два режима работы интерпретатора Cisco Exec.
5. Каково назначение пользовательского режима?

4

Пользовательский интерфейс Cisco IOS

В настоящей главе вы познакомитесь с пользовательским интерфейсом Cisco IOS. Из главы 3 вы уже получили некоторые сведения о IOS и режимах работы ее командного интерпретатора. Теперь поговорим подробнее о пользовательском интерфейсе и командах.

При обсуждении пользовательского интерфейса Cisco будут затронуты следующие темы:

- Взаимодействие с IOS
- Базовые элементы пользовательского интерфейса IOS
- Доступ к справочной системе Cisco IOS
- Основные команды IOS

Пользовательский интерфейс Cisco IOS – это единственное «окно» прямо в маршрутизатор. То есть пользовательский интерфейс – это средство, при помощи которого администратор получает доступ к командному интерпретатору. Если вы вообще не знакомы с маршрутизаторами и их интерфейсами, то первое, что вы заметите, будет очевидное отсутствие графического пользовательского интерфейса (GUI, Graphical User Interface) для Cisco IOS. И действительно Cisco IOS является операционной системой командной строки. Те из вас, кто ранее работал с системами командной строки, заметят, что пользовательский интерфейс Cisco во многом похож на интерфейсы многих Unix- или DOS-систем. Если же вам известны только графические интерфейсы, такие как Windows или MacOS, вам придется привыкнуть к новой модели.

Итак, давайте не будем терять время и погрузимся в глубины рассматриваемого интерфейса. В основе пользовательского интерфейса Cisco IOS лежат наборы команд, непосредственно связанных с двумя режимами работы командного интерпретатора.

Взаимодействие с IOS

Команды для маршрутизаторов Cisco разделяются на две категории: команды пользовательского режима и команды привилегированного режима работы. Команды пользовательского (базового) режима обеспечивают такие функции, как просмотр версий IOS и запуск утилит ICMP (Internet Control Message Protocol – протокол управляющих сообщений в Интернете). Когда пользователь регистрируется на маршрутизаторе Cisco, по умолчанию он находится в пользовательском режиме. В табл. 4.1 приведены некоторые из множества разнообразных команд, доступных в пользовательском и привилегированном режимах.

Таблица 4.1. Примеры команд пользовательского и привилегированного режимов

Режим	Команда	Режим	Команда
Пользовательский	`ping`	Привилегированный	`ping`
	`traceroute`		`traceroute`
	`help`		`help`
	`terminal`		`configure`
			`erase`

Обратите внимание на то, что многие команды пользовательского режима доступны также и в привилегированном. Режимы работы Cisco IOS имеют иерархическую структуру. То есть чем выше вы находитесь в командной структуре, тем больше команд вам доступны (при этом сохраняется доступ и ко многим командам более низкого уровня).

Внутри двух режимов работы Cisco IOS (пользовательского и привилегированного) имеется 17 уровней доступа, 16 из которых являются определяемыми, а один – фиксированным. Нижний уровень командной иерархии представлен пользовательским режимом, в котором доступны самые элементарные команды. Более сложные команды доступны в привилегированном режиме.

Привилегированный режим – это фактически конгломерация 16 настраиваемых командных уровней (от 0 до 15). Администратор может определить некоторые команды привилегированного режима так, чтобы они были доступны только в пределах какого-то уровня. Таким образом повышается контроль за доступом к маршрутизатору.

Примечание

Если уровни привилегированного режима не были определены, то по умолчанию IOS будет работать с уровнем 15 (доступ ко всем командам привилегированного режима).

Конфигурирование уровней привилегированного режима будет обсуждаться в главе 6 «Запуск маршрутизатора и работа с ним». Будут рассмотрены все процессы, необходимые для установки всех шестнадцати уровней доступа.

Примечание

В примерах этой книги, имеющих отношение к привилегированному режиму, считается, что уровни доступа для IOS не установлены, и маршрутизатор по умолчанию использует уровень 15.

В третьей главе говорилось о том, что привилегированный режим командного интерпретатора IOS отделен от пользовательского и защищен паролем. Чтобы получить доступ к набору команд привилегированного режима, пользователь должен ввести правильный пароль, который позволит перейти к привилегированному режиму. Командами привилегированного режима администратор может изменять конфигурацию, значения системных переменных, а также управлять портами. Любые операции с файлами проводятся с помощью команд привилегированного режима. Обычно администраторы маршрутизаторов большую часть своего времени работают в привилегированном режиме. Поэтому следует уделить особое внимание командам этого режима.

Cisco IOS также включает в себя основательную программу помощи для набора команд IOS, которая облегчает администратору процесс конфигурирования, но может немного напугать тех, кто не знаком с интерфейсом. Освоение контекстно-зависимой справочной системы Cisco IOS – это ключ к оперированию функциями и свойствами маршрутизатора.

Примечание

Большинство коммутаторов Cisco (и маршрутизаторы старших моделей с возможностями коммутации) предоставляют графический интерфейс пользователя на базе HTML для взаимодействия с Cisco IOS. Однако операционная система сама по себе остается ориентированной на командную строку. В книге будет рассмотрен только интерфейс командной строки Cisco IOS.

Причина использования командной строки в операционной системе маршрутизатора проста. Благодаря отсутствию графического интерфейса IOS имеет небольшой размер и меньше загружает процессор (оставляя большую часть времени на обработку данных для целей маршрутизации). В среднем размер Cisco IOS равен 16 Мбайт (в зависимости от предоставляемого набора функций многие версии имеют меньший размер), а рабочая частота процессора в большинстве моделей маршрутизаторов Cisco не превышает 100 МГц. Компактная операционная система дает маршрутизатору возможность делать то, что он должен – прокладывать маршруты передачи информации, не заботясь о рисовании окошек на мониторе и не отслеживая положения мыши.

Примечание

Интерфейс командной строки позволяет легко ориентироваться в последовательностях команд. Тем не менее, чтобы сделать использование IOS еще более простым, существуют стандартные функциональные клавиши.

Использование функциональных клавиш на маршрутизаторах Cisco облегчает взаимодействие пользователей с IOS. Такие клавиши обеспечивают быстрый вызов команд или средств навигации. Например, функциональная клавиша может прокрутить целую строку текста или вызвать предыдущие команды.

Примечание

Функциональные клавиши, иначе называемые клавишами усовершенствованной системы редактирования команд Cisco IOS, не настраиваются. Эти функции предопределены в IOS.

Если окажется, что на вашем маршрутизаторе не работают функциональные клавиши, это может объясняться тем, что они были выключены. Заблокировать функциональные клавиши можно, введя после приглашения на ввод команд следующее:

```
Router>terminal no editing
```

К счастью, редактирование можно без труда разрешить вновь, используя команду:

```
Router>terminal editing
```

Примечание

Такая структура команд достаточно часто встречается в Cisco IOS. То есть для того, чтобы отключить функцию, используется команда `no` перед той командой, которая эту функцию включила. Следуя этой логике, `no editing` выключает редактирование, а `editing` включает его.

Чаще всего вы будете использовать две клавиши: <Enter> и <пробел>. Помимо их явных функций завершения строки ввода и расположения пробелов между символами, данные клавиши также обеспечивают управление прокруткой. Если возвращаемых IOS данных так много, что они не помещаются на экран терминала, то появляется приведенная ниже подсказка, оповещающая администратора о том, что имеется еще информация для вывода:

```
--More--
```

Если после появления такой подсказки нажать клавишу <Enter>, то будет выведена следующая строка данных, если же нажать клавишу <пробел>, то будет выведена новая порция данных размером в полный

экран. Например, результатом просмотра текущей конфигурации маршрутизатора может быть такой вывод (не думайте пока о том, что означает такая выходная информация):

```
Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
boot system flash
no logging console
!
ip subnet-zero
!
!
!
interface Ethernet0
 ip address 10.16.4.153 255.240.0.0
 no ip directed-broadcast
!
interface Ethernet1
 no ip address
 --More--
```

Так как есть еще выходные данные, которые не поместились на экран терминала, то IOS выводит приглашение на прокрутку. Нажатие в этот момент клавиши <Enter> приведет к такому результату:

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
boot system flash
no logging console
!
ip subnet-zero
!
!
!
interface Ethernet0
 ip address 10.16.4.153 255.240.0.0
 no ip directed-broadcast
!
```

```
interface Ethernet1
 no ip address
 no ip directed-broadcast
 --More--
```

После нажатия клавиши <Enter> к изображению выходных данных была добавлена такая строка:

```
no ip directed-broadcast
```

Эта функция чрезвычайно полезна в тех случаях, когда вы ищете в перечне строк некоторую определенную строку данных. Так как маршрутизаторы работают в режиме реального времени, то для обнаружения интересующей вас информации можно просматривать строки по одной – так вы гарантированно не сможете пропустить нужные данные. Если же вы хотите увидеть всю следующую страницу результатов, используйте клавишу <пробел>.

Если в предыдущем примере нажать клавишу <пробел>, то будут выведены такие данные:

```
 --More--
 no ip directed-broadcast
 shutdown
!
ip classless
!
!
line con 0
 exec-timeout 0 0
 transport input none
line vty 0 4
 login
!
end
```

Возможность прокрутки целой страницы удобна в тех случаях, когда нужно быстро пролистать несколько страниц данных. Обе команды позволяют свободно перемещаться по спискам.

- Комбинации клавиш <Ctrl>+<B> и <Ctrl>+<F> работают точно так же, как кнопки клавиатуры со стрелками назад и вперед, соответственно. Они позволяют переместить курсор на один символ вперед или назад в строке текста и используются в случае, если требуется исправить один символ в длинной командной строке. Но если вы имеете дело с по-настоящему длинной командной строкой и посимвольное перемещение недостаточно быстро для вас, используйте функции <Esc>+<B> и <Esc>+<F>, описанные ниже.
- Комбинации клавиш <Esc>+<B> и <Esc>+<F> перемещают курсор соответственно назад и вперед на одно слово. Следовательно, если вы только что закончили длинную командную строку (а команды Cisco могут быть очень длинными) и хотите что-то в ней отредак-

тировать, то с помощью этих клавиш можно быстро просмотреть команду, перемещаясь с одного слова на другое.

- Чтобы перейти непосредственно в начало или в конец целой строки, используйте комбинации клавиш <Ctrl>+<A> и <Ctrl>+<E>. <Ctrl>+<A> переместит курсор на первый символ текущей строки, а <Ctrl>+<E> – на последний. Я часто использую эти клавиши для добавления параметров в конец длинной строки команд.
- Две последние (и, с моей точки зрения, самые полезные) комбинации функциональных клавиш IOS, <Ctrl>+<P> и <Ctrl>+<N>, называют клавишами истории команд. Используя их, администратор может воспроизвести ранее введенные команды.

Примечание

Клавиши со стрелками вверх и вниз могут заменять <Ctrl>+<P> и <Ctrl>+<N>, соответственно.

Комбинация клавиш <Ctrl>+<P> вызывает последнюю выполненную команду. Предположим, вы хотите вывести содержание каталога во флэш-памяти Cisco. Исполняется такая командная структура:

```
Router#dir
Directory of flash:/

 1 -rw-   2202092       <no date> aaa0269.bin

4194304 bytes total (1992148 bytes free)
Router#
```

Когда выполняется команда `dir`, IOS возвращает пустое приглашение на ввод. Если вы хотите запустить эту команду еще раз, нажмите комбинацию клавиш <Ctrl>+<P> (или стрелку вверх). Подобный ускоренный набор может сэкономить вам много времени при использовании команд с многочисленными параметрами.

Комбинация клавиш <Ctrl>+<P> также используется для вызова нескольких предыдущих команд. Например, пусть выполнялась такая строка команд:

```
Router#
Router#dir
Directory of flash:/

 1 -rw-   2202092       <no date> aaa0269.bin

4194304 bytes total (1992148 bytes free)
Router#ping 10.16.4.152

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.16.4.152, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Router#sho flash
```

```
PCMCIA flash directory:
File Length  Name/status
 1  2202092 aaa0269.bin
[2202156 bytes used, 1992148 available, 4194304 total]
4096K bytes of processor board PCMCIA flash (Read/Write)

Router#
```

Чтобы повторно выполнить первую команду, просто трижды нажмите комбинацию клавиш <Ctrl>+<P>. Это вернет команду `dir` в приглашение.

При использовании комбинации клавиш <Ctrl>+<P> вы можете случайно «проскочить» (слишком углубиться назад) команду, которую необходимо заново вызвать. В таком случае используйте комбинацию клавиш <Ctrl>+<N>, чтобы прокрутить восстановленные команды вперед.

Все комбинации функциональных клавиш могут быть очень полезны. Не пожалейте времени на их изучение и попробуйте использовать их при конфигурировании и поддержке маршрутизаторов. Теперь же давайте перейдем к рассмотрению основных элементов пользовательского интерфейса Cisco IOS.

Основные элементы пользовательского интерфейса IOS

Пользовательский интерфейс Cisco IOS включает в себя четыре основных компонента: сообщения о состоянии, запросы на ввод, приглашение IOS на ввод команды и курсор. Вышеназванные компоненты проиллюстрированы на рис. 4.1.

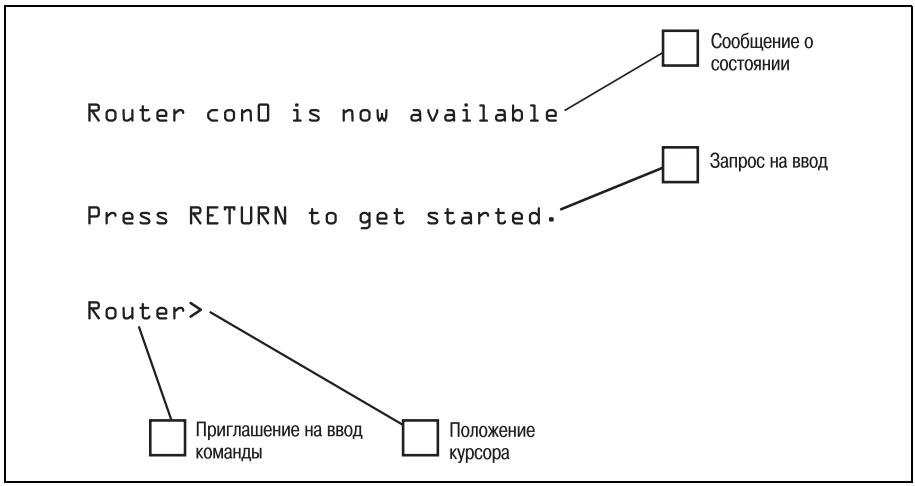

Рис. 4.1. *Пользовательский интерфейс Cisco IOS*

Из четырех основных элементов, представленных на рис. 4.1, чаще всего вы, очевидно, будете иметь дело с приглашением на ввод команды. Заметьте, что приглашение на ввод команды выглядит (рис. 4.1) как слово Router, за которым следует символ >. Обе составляющие имеют свое собственное значение в рамках IOS.

Примечание

Пятый элемент пользовательского интерфейса, заголовочное сообщение, – это конфигурируемое сообщение, которое маршрутизатор может выводить при регистрации пользователя. Так как заголовок – это скорее настраиваемый параметр, чем базовый элемент, то он будет рассмотрен в главе 6 «Запуск маршрутизатора и работа с ним».

Приглашение на ввод в командной строке Cisco IOS подразделяется на две логических части: имя маршрутизатора и индикатор режима. Имя маршрутизатора (первая часть приглашения на ввод команды) – это всегда первое слово в отображении (не считая слов в скобках). Например, в приведенном ниже приглашении на ввод имя маршрутизатора – «Frank».

```
Frank(config)#
```

Имя присваивается маршрутизатору администратором, оно произвольно и время от времени может меняться. Приглашение на ввод команды всегда выводит имя маршрутизатора. (Во всех примерах данной книги в качестве имени маршрутизатора используется Router.)

Вторая часть приглашения на ввод – это индикатор режима. Символ > показывает, что администратор вошел в систему в пользовательском режиме. Для обозначения привилегированного режима используется символ #. Следовательно, приглашение на ввод

```
Router>
```

показывает, что Exec на маршрутизаторе Router работает в пользовательском режиме и может выполнять только команды пользовательского режима, в то время как приглашение на ввод

```
Router#
```

означает, что Exec на маршрутизаторе Router работает в привилегированном режиме и может выполнять команды этого режима. Просто посмотрев на приглашение, вы легко определите, на каком маршрутизаторе вы работаете и в каком режиме обращаетесь к нему в настоящий момент. Такая возможность особенно полезна в больших сетях, где велик шанс обращения с одного устройства к нескольким маршрутизаторам.

При обсуждении имени маршрутизатора было упомянуто, что это первое слово в приглашении на ввод команды, *не считая слов в скобках*.

Если приглашение на ввод команды содержит слова в скобках, то они указывают подрежимы. Например, следующее приглашение на ввод указывает, что маршрутизатор Router работает в «режиме глобального конфигурирования» привилегированного режима:

```
Router(config)#
```

Варианты подрежимов привилегированного режима и их индикаторы перечислены в табл. 4.2.

Таблица 4.2. Индикаторы подрежима

Подрежим	Индикатор
Режим глобального конфигурирования	(config)
Режим конфигурирования маршрутизатора	(config-router)
Режим конфигурирования интерфейса	(config-if)

Реже других используется такая составляющая пользовательского интерфейса, как сообщение о начальном состоянии (только потому, что его видно лишь при запуске). Сообщения о состоянии пользовательского интерфейса Cisco IOS просто показывают текущее состояние порта (терминала), используемого IOS. В примере, изображенном на рис. 4.1, говорится, что con0 (порт консоли 0) находится в состоянии готовности. Если бы были какие-то текущие сообщения о состоянии порта, они были бы выведены.

Более распространенными являются сообщения о состоянии, появляющиеся непосредственно под приглашением на ввод. Cisco IOS выдает все сообщения о состоянии непосредственно под породившей их командой.

Примечание

Любое сообщение о состоянии, появляющееся под приглашением на ввод команды, порождается командным интерпретатором Cisco.

Примером может послужить реакция на ввод некорректной команды. Сообщение о состоянии, выдаваемое в ответ на некорректный ввод, приведено на рис. 4.2.

На рис. 4.2 показано, что после приглашения на ввод команды было введено слово run, которое не распознается Cisco IOS как команда. Поэтому IOS генерирует сообщения о состоянии, чтобы информировать пользователя о том, что маршрутизатор пытается расшифровать то, что было введено.

Первая строка, приведенная на рис. 4.2 сразу после приглашения на ввод команды, – это сообщение о состоянии, которое выглядит так:

```
Translating "run"…domain server (255.255.255.255)
```

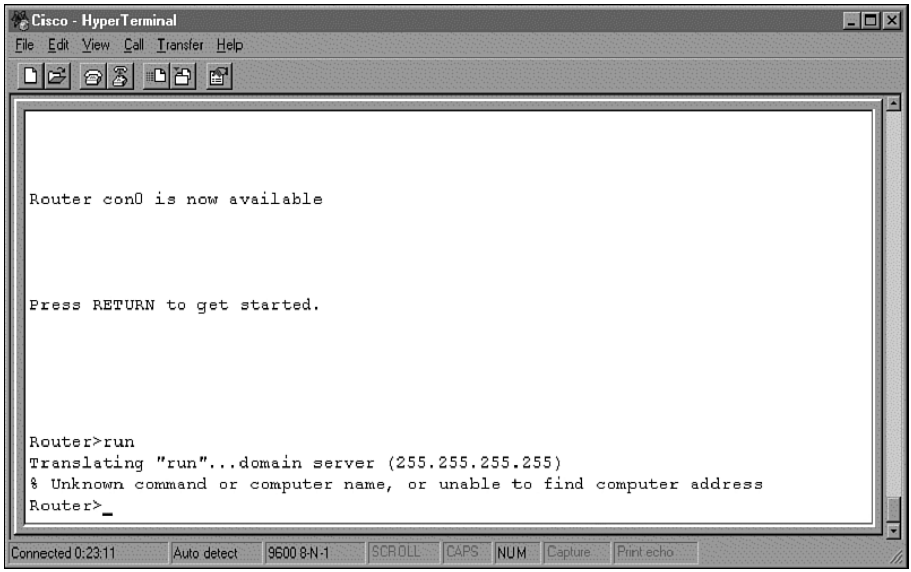

Рис. 4.2. *Сообщение о состоянии, появившееся в результате ввода некорректной команды*

Пока не углубляясь в значение сообщения (займемся этим чуть позже), можно сказать, что IOS пыталась (в тот момент) транслировать введенную команду, поэтому состояние маршрутизатора – `translating`.

Вторая строка сообщения о состоянии на рис. 4.2, "`% Unknown command or computer name, or unable to find computer address`", – это *диагностическое* сообщение, поясняющее результат состояния. Другими словами, маршрутизатор попытался транслировать команду `run`, в результате чего обнаружил, что это неизвестная команда.

Диагностические сообщения, сопровождающие сообщение о состоянии, всегда начинаются с символа `%`. Но такие сообщения встречаются не только после сообщений о состоянии – вы увидите их и при использовании справочной системы Cisco IOS.

Завершающим элементом пользовательского интерфейса Cisco IOS является запрос на ввод. Относящаяся к запросу на ввод часть пользовательского интерфейса Cisco IOS – это то средство, которое IOS использует для того, чтобы показать, что необходимы некие входные данные. В большинстве случаев ожидаемый ответ или ответ по умолчанию будет выведен непосредственно за запросом. Например:

```
Would you like to enter the initial configuration dialog? [yes/no]:
```

Первая часть выражения – это собственно сообщение, то есть вопрос, задаваемый маршрутизатором. В приведенном примере спрашивает-

ся, хочет ли администратор начать процесс начального конфигурирования. Но нас интересует вторая часть предложения.

Часть выражения, выделенная символами [ ] (квадратные скобки), – это запрос на ввод. Cisco IOS будет (обычно) предлагать список возможных ответов на заданный вопрос. Ответы всегда будут заключены в квадратные скобки и отделены друг от друга символом / (косая черта, или слэш). В предыдущем примере IOS ожидает получить ответ `yes` или `no`. Для продолжения работы пользователь должен ввести один из предложенных ответов.

Если в скобках приведен один вариант ответа, то это ответ по умолчанию. Например:

```
Enter host name [Router]:
```

Cisco IOS запрашивает новое имя для маршрутизатора. Предлагается имя по умолчанию – `Router`. Нажав клавишу Enter, вы согласитесь с выбором по умолчанию, если же вы хотите изменить имя маршрутизатора, то наберите его.

Если администратор хочет изменить уже сделанные установки (такие как IP-адрес некоторого порта), то IOS часто предлагает в качестве ответа имеющуюся конфигурацию. Пусть, например, Ethernet-порт маршрутизатора имеет адрес 10.19.25.1. Если администратор захочет изменить этот адрес, то появится такое сообщение:

```
Enter IP address for Eth0 [10.19.25.1]:
```

Оно означает, что первому порту Ethernet (Eth0) уже назначен адрес 10.19.25.1. Чтобы оставить тот же адрес, просто нажмите клавишу Enter, и значение по умолчанию будет сохранено. Если же администратор хочет изменить адрес порта, он может ввести новый адрес.

Теперь, когда вы познакомились с четырьмя основными элементами пользовательского интерфейса Cisco IOS, давайте рассмотрим сообщения IOS, которые могут появляться при запуске системы.

Сообщения при запуске IOS

При загрузке маршрутизатора Cisco можно заметить несколько различных сообщений IOS. И хотя большая их часть кажется абракадаброй или появляется и исчезает слишком быстро, для того чтобы их можно было понять, все они имеют смысл. Давайте немного поговорим о сообщениях при запуске Cisco IOS.

Первая группа появляющихся сообщений принадлежит Cisco Boot Loader (загрузчику операционной системы). Это сообщения общего характера, относящиеся к образу системы, используемому при загрузке маршрутизатора. Обычно они исчезают довольно быстро, но если вы сможете их прочитать, то найдете там следующую информацию:

1. Версия Cisco System Bootstrap

2. Физическая модель маршрутизатора
3. Конфигурация памяти маршрутизатора
4. Адрес точки входа стартового образа

На рис. 4.3 изображен маршрутизатор Cisco в момент запуска. Обратите внимание на выведенную информацию.

Все сообщения, показанные на рис. 4.3, появляются даже до загрузки стартового образа. Это делается для того, чтобы диагностировать все возможные проблемы как с маршрутизатором, так и со стартовым образом до того, как какие-либо данные будут загружены в память маршрутизатора. Как и в большинстве цифровых систем, такая проверка системы уменьшает возможность разрушения файлов конфигурации.

Процесс распаковки обозначен последовательностью символов #. Когда весь образ распакован в память, IOS возвращает статус [OK]. Это означает, что распаковка прошла успешно и маршрутизатор готов продолжить загрузку параметров конфигурации.

После того как образ распакован в память маршрутизатора, на экран будет выведена стандартная информация об авторских правах. В заключение IOS выводит данные о своей версии. Надпись об авторских правах и информация о версии IOS представлены на рис. 4.4.

Последнее сообщение о состоянии IOS, выводимое при запуске, извещает об объеме доступной памяти маршрутизатора. Затем маршрутизатор входит в пользовательский режим работы.

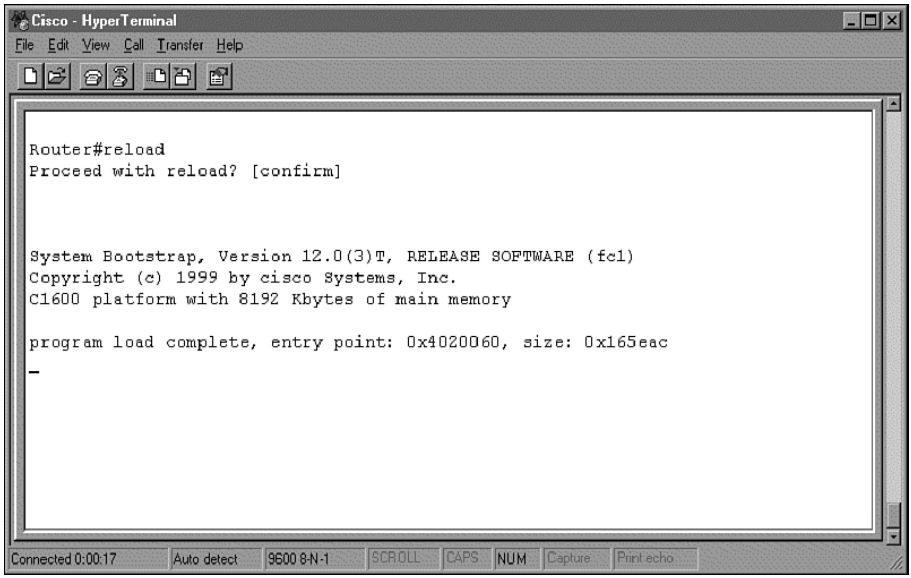

Рис. 4.3. *Сообщения, появляющиеся при запуске Cisco IOS*

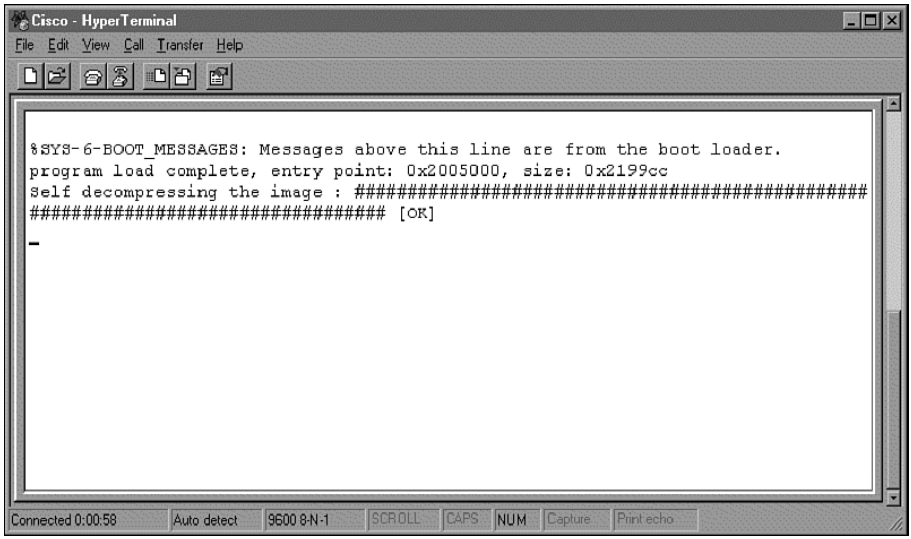

Рис. 4.4. *Информация о версии IOS после включения маршрутизатора*

Примечание

Если маршрутизатор запускается впервые, то в этот момент он не войдет в пользовательский режим. Вместо этого маршрутизатор входит в режим конфигурирования системы. В этом режиме существует ряд подсказок, помогающих администратору в конфигурировании маршрутизатора. О режиме конфигурирования поговорим в главе 6 «Запуск маршрутизатора и работа с ним».

В Cisco IOS встроена всеобъемлющая справочная система. Она может быть очень полезной администратору маршрутизатора Cisco, но даже лучшая справочная система не представляет никакой ценности, если вы не знаете, как ею пользоваться. Я до сих пор заглядываю в справочники в поиске определений малопонятных функций или трактовок ключевых слов и командных структур.

В заключительных разделах этой главы мы поговорим о том, как устроена система помощи Cisco IOS, и изучим основные команды, необходимые для перемещения по внутренней структуре маршрутизатора. После того как вы освоите такие команды и научитесь пользоваться справочной системой Cisco, вы будете готовы к восприятию последующих глав. Итак, давайте займемся справочной системой Cisco IOS.

Обращение к системе помощи Cisco IOS

Основное назначение справочной системы Cisco IOS заключается в предоставлении пользователю низкоуровневой помощи для работы с маршрутизатором. Чтобы получить доступ к основным функциям

справочной системы, введите help в ответ на приглашение на ввод команды в пользовательском режиме:

```
Router>help
```

Возвращаемое командой help сообщение показывает, что существует два уровня помощи. Первый уровень называют полной справкой. Полная справка используется, когда требуется определить, какие команды могут быть выполнены в командной строке, а также для того, чтобы узнать, какие команды можно выполнять совместно с другими командами.

Обратиться за помощью можно в любом месте команды, введя знак вопроса ?. Если соответствие не обнаружено, то перечень, выводимый справочной системой, будет пуст, и вам необходимо будет двигаться назад до тех пор, пока ввод символа ? не приведет к отображению возможных параметров.

Предусмотрены два типа справки:

- Полная справка доступна, когда вы вводите символ ? вместо аргумента команды (например, show ?) и хотите получить описание всех возможных аргументов.
- Частичная справка предоставляется, когда вы вводите неполное название аргумента и хотите узнать, какие аргументы совпадают с вводом (например, show pr?).

Полная справка

Чтобы узнать, какие команды можно выполнить в конкретный момент времени, следует просто ввести в командной строке символ ? и воспользоваться полной справкой Cisco. Будучи самостоятельно введенной после приглашения на ввод, команда ? выведет перечень всех команд, которые могут быть запущены именно в этом приглашении на ввод. Например, ввод символа ? после приглашения на ввод в пользовательском режиме приведет к выводу такого списка команд:

```
Router>?
Exec commands:
 access-enable    Create a temporary Access-List entry
 access-profile   Apply user-profile to interface
 clear            Reset functions
 connect          Open a terminal connection
 disable          Turn off privileged commands
 disconnect       Disconnect an existing network connection
 enable           Turn on privileged commands
 exit             Exit from the EXEC
 help             Description of the interactive help system
 lock             Lock the terminal
 login            Log in as a particular user
 logout           Exit from the EXEC
```

```
name-connection  Name an existing network connection
pad              Open a X.29 PAD connection
ping             Send echo messages
ppp              Start IETF Point-to-Point Protocol (PPP)
resume           Resume an active network connection
rlogin           Open an rlogin connection
set              Set system parameter (not config)
show             Show running system information
slip             Start Serial-line IP (SLIP)
systat           Display information about terminal lines
--More--
```

Вывод функции полной справки состоит из списка команд, которые в текущий момент могут быть запущены в рамках пользовательского режима. Если вы ищете команду для выполнения какой-то конкретной операции, то такая команда (если она существует) появится в списке.

Возможности полной справки Cisco IOS также позволяют узнать, какие команды могут использоваться вместе с данной. Например, если ввести `terminal` после приглашения на ввод команды в привилегированном режиме, будет выдан такой ответ:

```
Router#terminal
% Incomplete command.
```

Если же использовать символ `?` после команды `terminal`, то будет выведен перечень команд, которые могут следовать за ключевым словом `terminal`:

```
Router#terminal ?
 autohangup             Automatically hangup when last connection closes
 data-character-bits    Size of characters being handled
 databits               Set number of data bits per character
 default                Set a command to its defaults
 dispatch-character     Define the dispatch character
 dispatch-timeout       Set the dispatch timer
 domain-lookup          Enable domain lookups in show commands
 download               Put line into 'download' mode
 editing                Enable command line editing
 escape-character       Change the current line's escape character
 exec-character-bits    Size of characters to the command exec
 flowcontrol            Set the flow control
 full-help              Provide help to unprivileged user
 help                   Description of the interactive help system
 history                Enable and control the command history function
 hold-character         Define the hold character
 international          Enable international 8-bit character support
 ip                     IP options
 length                 Set number of lines on a screen
 monitor                Copy debug output to the current terminal line
 no                     Negate a command or set its defaults
```

```
notify                  Inform users of output from concurrent sessions
padding                 Set padding for a specified output character
parity                  Set terminal parity
rxspeed                 Set the receive speed
special-character-bits  Size of the escape (and other special) characters
speed                   Set the transmit and receive speeds
start-character         Define the start character
stop-character          Define the stop character
stopbits                Set async line stop bits
telnet                  Telnet protocol-specific configuration
terminal-type           Set the terminal type
transport               Define transport protocols for line
txspeed                 Set the transmit speeds
width                   Set width of the display terminal
```

Примечание

Обратите внимание на интервал между командой `terminal` и символом справки `?`. Этот интервал имеет важное значение. Например,

```
Router>terminal ?
```

выдаст результаты, сильно отличающиеся от результатов выполнения

```
Router>terminal?.
```

В то время как первый пример перечислит все ключевые слова и параметры, связанные с командой `terminal`, второй пример выведет все команды, которые начинаются со слова `terminal`.

Эти интервалы между символами и отличают полную справку от частичной (также называемой контекстно-зависимой), речь о которой пойдет в следующем разделе.

Еще одно назначение полнофункциональной справочной системы заключается в том, чтобы указывать вам на то, что введенная команда не существует. При попытке ввести в приглашение на ввод несуществующую команду IOS выведет следующее сообщение:

```
Router>run
Translating "run"...domain server (255.255.255.255)
```

Контекстно-зависимая помощь

Частичная или контекстно-зависимая справка Cisco дает пользователям возможность получить результат для не полностью введенной команды. Например, если вы знаете, что команда для присвоения имени физическому порту начинается с `name-`, а что идет дальше – не помните, используйте контекстную справку для того, чтобы получить ответ на свой вопрос.

Справочная система IOS воспринимает символ `?` как групповой, используя его для того, чтобы получить все результаты, совпадающие с набором символов, стоящих перед ним. Другими словами, ввод в ко-

мандной строке name-? приведет к выводу таких результатов (отметьте, что интервал между командой и знаком вопроса отсутствует):

```
Router>name-?
name-connection
```

Единственной командой пользовательского режима, начинающейся на name-, является name-connection. Это и есть команда для назначения имени физическому порту маршрутизатора.

Контекстно-зависимая справка также может применяться для осуществления поиска, при котором необходимо знать всего одну букву из интересующей вас команды. Например, посмотрим на приведенный ниже поиск, использующий контекстную справку:

```
Router>t?
telnet terminal traceroute tunnel

Router>t
```

Обратите внимание на состояние командной строки после выполнения контекстно-зависимого поиска. Cisco IOS всегда возвращает символы, по которым запрашивалась информация, в приглашение на ввод команды. В предыдущем примере после выполнения поиска для t? в командную строку был автоматически введен символ t. Теперь все, что вы будете вводить в этой строке, будет следовать за буквой t.

Основные команды IOS

Вы уже хорошо представляете себе, как работает Cisco IOS, и можете свободно взаимодействовать с пользовательским интерфейсом и справочной системой. Пришло время исследовать базовые команды IOS. Остаток главы познакомит вас с некоторыми такими командами. Их изучение позволит вам более уверенно чувствовать себя с маршрутизатором и подготовит к восприятию более сложных функций, которые встретятся далее в этой книге.

Примечание

Замечательной особенностью Cisco IOS является способность распознавать не полностью введенные команды. То есть если ввести в командной строке tr, то IOS распознает эти символы как команду traceroute, так как в пользовательском режиме IOS нет других команд, начинающихся с символов tr.

А вот команд, начинающихся с буквы t, не считая traceroute, есть еще четыре. Поэтому, если ввести в командной строке один символ t и нажать клавишу Enter, то в результате будет выведено сообщение о вводе неоднозначной (ambiguous) команды:

```
% Ambiguous command: "t"
```

В большей части примеров данного издания используются неполные названия команд (когда это возможно). Поэтому постарайтесь освоить сокращения, т. к. их применение упрощает администрирование маршрутизатора.

Первая команда, которую необходимо знать, это команда enable – команда пользовательского режима, которая переключает командный интерпретатор Cisco из пользовательского режима работы в привилегированный:

```
Router>en
Router#
```

Будучи администратором маршрутизатора, вы обнаружите, что большинство команд (если не все), с которыми вы имеете дело в ходе вашей повседневной работы, принадлежат привилегированному режиму. Следовательно, вам необходимо знать, как получить доступ к этому режиму работы командного интерпретатора. Но вход в привилегированный режим – это только полдела. Необходимо знать, как вернуться обратно в пользовательский режим.

```
Router#disa
Router>
```

Команда disable переключает командный интерпретатор из привилегированного режима в пользовательский.

Команда exit осуществляет выход из пользовательского интерфейса. Используя команды enable и disable, вы сможете управлять переключением режимов работы командного интерпретатора.

ping

Может быть, вы уже знакомы с командой ping. «Ping» – это общепринятый межплатформенный сетевой термин для проверки существования адресата путем передачи ему специального сигнала – запроса отклика ICMP и ожидания ответа. Cisco IOS предоставляет возможность выполнения команды ping в пользовательском режиме.

```
Router>ping 10.16.4.152
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.16.4.152, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Анализируя выходные данные, можно понять, что команда ping пыталась отправить пять 100-байтных пакетов по указанному адресу. Последовательность точек под сообщением – это пиктограммы состояния, показывающие, что пакеты не достигли предполагаемого адресата. Они подкрепляются сообщением Success rate is 0 percent (0/5). Различные пиктограммы состояния для команды ping представлены в табл. 4.3.

Таблица 4.3. Пиктограммы состояния команды `ping`

Пиктограмма	Значение
!	Получение ответа
.	Отсутствие ответа
U	Адресат недостижим
C	Перегрузка
I	Тест прерван пользователем
?	Неизвестный тип пакета
&	Исчерпано время жизни

Давайте рассмотрим еще несколько основных команд.

show

Команда `show` относится к привилегированному режиму работы и используется для конфигурирования маршрутизатора. Полная справка по команде `show` выводит разнообразные команды, которые предоставляют подробную информацию о маршрутизаторе. Возможные аргументы команды `show` представлены в табл. 4.4.

Таблица 4.4. Возможные аргументы команды `show`

Команда	Значение
`access-expression`	Перечисляет формулы доступа
`access-lists`	Перечисляет списки доступа
`accounting`	Показывает данные об учетных записях для активных сеансов связи
`aliases`	Выводит псевдонимы команд
`arp`	Выводит таблицу ARP (Address Resolution Protocol – протокол разрешения адресов)
`async`	Выводит информацию о терминальных линиях, используемых как интерфейсы маршрутизатора
`bridge`	Выводит базу данных Bridge Forwarding/Filtering (может быть или не быть подробной)
`buffers`	Выводит статистику буферного пула
`cdp`	Выводит информацию о протоколе Cisco Discovery Protocol (CDP)
`clock`	Выводит системную дату и время
`compress`	Показывает статистику сжатия
`configuration`	Перечисляет содержимое энергонезависимой (флэш) памяти
`controllers`	Выводит состояние контроллера интерфейса

Команда	Значение
debugging	Выводит состояние каждого параметра отладки
dhcp	Выводит информацию о протоколе DHCP (Dynamic Host Configuration Protocol – протокол динамической конфигурации хоста)
diag	Выводит диагностическую информацию для платы расширения WAN
dialer	Выводит параметры наборного устройства и статистику по нему
dnsix	Выводит информацию о Department of Defense Intelligence Information System Network
dxi	Показывает информацию об интерфейсе Asynchronous Transfer Mode (ATM)
entry	Показывает состояние очереди
file	Показывает информацию о файловой системе
flash:	Показывает информацию о файловой системе flash:
flh-log	Показывает буфер протокола Flash Load Helper
frame-relay	Показывает информацию о Frame Relay
history	Выводит историю команд сессии
hosts	Выводит доменное имя IP, режим поиска, серверы имен и таблицу хостов
interfaces	Состояние и установки интерфейса
ip	Выводит информацию об IP
key	Выводит информацию о ключе
line	Выводит информацию о линии TTY
llc2	Выводит информацию о линии IBM LLC2
location	Выводит расположение системы
logging	Показывает содержимое буферов протоколирования
memory	Показывает статистику для памяти
modemcap	Показывает базу данных Modem Capabilities (характеристики модема)
ppp	Выводит параметры и статистику протокола PPP (Point-to-Point, точка-точка)
privilege	Показывает текущий уровень привилегированного режима
processes	Показывает статистику по активным процессам
protocols	Перечисляет активные протоколы сетевой маршрутизации
queue	Показывает список очередности
queueing	Показывает установки организации очередей

Таблица 4.4 (продолжение)

Команда	Значение
registry	Выводит информацию о регистрах
reload	Выводит информацию о расписании перезагрузок
rhosts	Выводит соответствие удаленных хостов пользователям
rmon	Показывает статистику rmon
route-map	Выводит информацию о карте маршрутов
rtr	Показывает данные о Response Time Reporter (RTR)
running-config	Показывает текущую настройку маршрутизатора
sessions	Показывает информацию о Telnet-соединениях
smf	Показывает информацию о программном фильтре MAC
snapshot	Показывает статистику и параметры моментальных снимков
snmp	Показывает статистику протокола SNMP (Simple Network Management Protocol, простой протокол сетевого управления)
sntp	Показывает информацию о SNTP (Simple Network Time Protocol, простой протокол сетевого времени)
spanning-tree	Показывает топологию связующего дерева
stacks	Выводит данные об использовании стека процессом
standby	Выводит информацию о протоколе HSRP (Hot Standby Router Protocol, протокол маршрутизатора горячего резерва)
startup-config	Показывает начальную конфигурацию маршрутизатора
subscriber-policy	Выводит данные о политике подписки
subsys	Показывает информацию о подсистеме
tacacs	Показывает статистику сервера Terminal Access Controller Access Control System (TACACS+)
tcp	Состояние соединений по протоколу TCP (Transmission Control Protocol, протокол управления передачей)
tech-support	Показывает системную информацию для технической поддержки
terminal	Выводит установочные параметры терминала
traffic-shape	Показывает параметры формирования трафика
users	Выводит информацию о терминальных линиях
version	Выводит состояние аппаратного и программного обеспечения системы
whoami	Выводит информацию о текущей линии TTY
x25	Выводит информацию по X.25
x29	Выводит информацию по X.29

Чаще других с командой show будут использоваться три элемента, приведенные в таблице: interfaces, running-config и startup-config.

Команда show interfaces выводит информацию о состоянии физических портов вашего маршрутизатора. Обычно просмотр данных о состоянии интерфейса является первым этапом в определении того, существуют ли какие-то проблемы с одним из портов маршрутизатора (такие как проблема с физическим портом или сетевая). Выполнение команды show interfaces даст результат, подобный приведенному в следующем примере. (Пока что не пытайтесь понять значение выведенной информации, займемся этим позже. Сейчас просто познакомьтесь со структурой команды и ее вывода.)

```
Router#sh int e0
Ethernet0 is up, line protocol is down
 Hardware is QUICC Ethernet, address is 00d0.58a8.e150 (bia 00d0.58a8.e150)
 Internet address is 10.16.4.153/12
 MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 128/255, load 1/255
 Encapsulation ARPA, loopback not set, keepalive set (10 sec)
 ARP type: ARPA, ARP Timeout 04:00:00
 Last input never, output 00:00:00, output hang never
 Last clearing of "show interface" counters never
 Queueing strategy: fifo
 Output queue 0/40, 0 drops; input queue 0/75, 0 drops
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes, 0 no buffer
   Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   0 input packets with dribble condition detected
   6496 packets output, 389772 bytes, 0 underruns
   6496 output errors, 0 collisions, 1 interface resets
   0 babbles, 0 late collision, 0 deferred
   6496 lost carrier, 0 no carrier
   0 output buffer failures, 0 output buffers swapped out
```

Обратите внимание, что фактически выполнялась команда sh int e0. Аргумент e0 указывает команде show, что вас интересует состояние только для порта Ethernet 0. Если бы команда была введена без параметра e0, то IOS вывела бы статусы всех портов маршрутизатора.

Команды show running-config и show startup-config используются для просмотра конфигурационных файлов, которые управляют маршрутизатором. Установку конфигурационных параметров маршрутизатора мы обсудим в следующей главе, просмотреть же файлы можно с помощью команды show.

Очевидно, что далее в этой книге вы встретитесь еще со многими командами IOS. Если вы хотите пока попрактиковаться с основными командами, выполните упражнения в конце главы.

Освоение пользовательского интерфейса Cisco IOS – это огромный шаг вперед в изучении маршрутизации Cisco. Преодолев этот барьер, перейдем к разговору о «Перемещении данных маршрутизаторами» (глава 5).

Резюме

Пользовательский интерфейс Cisco IOS похож на интерфейсы других операционных систем командной строки (таких как Linux и MS DOS).

Cisco предусматривает множество комбинаций функциональных клавиш (также называемых клавишами усовершенствованной системы редактирования команд), упрощающих выполнение часто встречающихся задач.

В Cisco IOS существует два основных режима выполнения команд: пользовательский и привилегированный. Привилегированный режим работы интерпретатора IOS подразделяется на 16 уровней доступа.

Справочная система Cisco IOS разделена на две функционально различные части. Система полной справки оказывает содействие в описании параметров команд, в то время как контекстно-зависимая справка осуществляет синтаксическую помощь.

Вопросы-ответы

Вопрос Почему для маршрутизаторов Cisco не предусмотрен графический интерфейс пользователя?

Ответ Графический интерфейс занимает очень много системных ресурсов. Основная функция маршрутизаторов заключается в том, чтобы перемещать данные с места на место, а не в том, чтобы выводить их на экран. Пренебрегая графическим интерфейсом пользователя, Cisco сохраняет функциональность маршрутизатора максимально рациональной.

Тест

Вопросы

1. В каких двух режимах работает интерпретатор Cisco IOS?
2. Об использовании какой разновидности справки свидетельствует положение знака вопроса в приведенном примере: `Router>sho?`
3. Какая команда используется для входа в привилегированный режим командного интерпретатора Cisco?
4. Когда на экран выводится информация об авторских правах?

Ответы

1. В пользовательском и привилегированном.
2. Контекстно-зависимой.
3. Команда `enable`.
4. Во время запуска.

Упражнения

1. Установка системных часов:

   Попробуйте установить системные часы, используя команду `clock`. Ответ приведен ниже.

```
Router#clo ?
  set Set the time and date

Router#clo set ?
  hh:mm:ss Current Time

Router#clo set 09:32:00 ?
  <1-31> Day of the month
  MONTH  Month of the year

Router#clo set 09:32:00 07 July ?
  <1993-2035> Year

Router#clo set 09:32:00 07 July 2001
Router#
```

2. Просмотр системной памяти:

   Теперь попытайтесь просмотреть системную память.

```
Router#sh flash

PCMCIA flash directory:
File Length  Name/status
 1  2202092 aaa0269.bin
[2202156 bytes used, 1992148 available, 4194304 total]
4096K bytes of processor board PCMCIA flash
(Read/Write)
```

5

Перемещение данных маршрутизаторами

Цель этой главы – дать представление о процессах, происходящих при перемещении данных от системы к системе. Мы обсудим роль протоколов в процессе маршрутизации и то, как они используются при передаче данных по сети.

В главе рассматриваются следующие темы:

- Маршрутизаторы Cisco и сетевые уровни
- Протоколы маршрутизации
- Механизмы маршрутизации
- Маршрутизируемые протоколы и протоколы маршрутизации

Эти темы представляют широкий диапазон концепций, лежащих в основе процессов передачи данных в сетях. Понимание этих процессов поможет в усвоении материала последующих глав, непосредственно касающихся маршрутизации и маршрутизируемых протоколов.

После прочтения данной главы вы должны представлять себе полную картину процессов маршрутизации на многих уровнях. Мы будем рассматривать маршрутизацию с точки зрения модели OSI (Open Systems Interconnect – взаимодействие открытых систем) на уровне протоколов, пакетов и аппаратуры. Материал этой главы поможет создать фундамент, который ляжет в основу опыта обслуживания маршрутизаторов Cisco.

Эта глава открывает путь к пониманию материала последующих глав. Поняв, как маршрутизаторы перемещают данные, вы легко справитесь с настройкой Cisco IOS для работы с различными протоколами и обеспечения маршрутизации между разными средами.

Давайте взглянем поближе на то, как работают маршрутизаторы Cisco (на двоичном уровне) и какую роль в их работе играют уровни модели OSI.

Маршрутизаторы Cisco и сетевые уровни

Все применяемые ныне сетевые протоколы передачи данных соответствуют одной общей спецификации. Эти протоколы были разработаны в соответствии с набором требований модели OSI. Модель OSI – это универсальный каркас, на основе которого различные разработчики создают протоколы, способные взаимодействовать друг с другом. Благодаря такому общему набору правил компьютеры и другие устройства способны общаться между собой независимо от их производителей, разработчиков и платформ.

Примечание

В обсуждении технологий маршрутизации основополагающей концепцией является способность к взаимодействию. При том количестве различных систем, протоколов и типов данных, которые встречаются в Интернете, производитель не в состоянии предугадать, где и как будут использоваться его маршрутизаторы.

Есть две фундаментальные причины, заставляющие строить все протоколы по одной общей схеме. Во-первых, разработчик, следуя заранее определенным указаниям по реализации протокола, не упустит из вида важных функций. Если, например, команда разработчиков начинает работу над новым транспортным протоколом, она должна следовать правилам, определенным для транспортного уровня модели OSI. Согласно правилам, этот протокол должен быть ориентированным на соединение и уметь (среди прочего) управлять потоком. Выполнив все эти требования, разработчики могут быть уверены в том, что их новый транспортный протокол сможет работать с любым устройством, совместимым с транспортным уровнем OSI. В скоординированной подобным образом среде облегчается создание сетей.

Маршрутизаторы тоже используют преимущества общего подхода. Процесс перемещения данных из одного места в другое полностью основан на принципах модели OSI и инкапсуляции протоколов. Маршрутизаторы Cisco работают на сетевом уровне модели OSI. Это позволяет им взаимодействовать с любым протоколом, соответствующим этой спецификации.

Примечание

Способность маршрутизировать протоколы, относящиеся к определенному сетевому уровню, и фактическая *маршрутизация каждого* протокола – далеко не одно и то же. Другими словами, даже если маршрутизаторы Cisco теоретически и могут работать с любыми протоколами сетевого уровня OSI, это не означает, что IOS каждого конкретного маршрутизатора может быть настроена на выполнение таких опера-

ций. Маршрутизатор Cisco может оказаться неспособным обрабатывать некоторый протокол просто потому, что в IOS не предусмотрены необходимые настройки.

То, что все маршрутизаторы работают на сетевом уровне, имеет одно простое объяснение. Именно сетевой уровень отвечает за адресацию протокола. Следовательно, каждый протокол сетевого уровня должен иметь возможность обратиться (адресоваться) к любой доступной системе. Такие адреса и лежат в основе маршрутизации.

Чтобы лучше разобраться в концепциях маршрутизации, адресации и протоколов, приведем пример из повседневной жизни. Например, адрес в протоколе можно сравнить с адресом дома. Этот адрес определяет номер дома, улицу, город и штат:

```
123 Maple Street
Anytown, Massachusetts
```

Адрес дома однозначно определяет его местонахождение. Точно так же и компьютеры адресуются протоколами.

Адрес компьютера в некотором протоколе очень похож на адрес дома. Адрес определяет сеть, в которой находится компьютер и его номер. Для доставки любой информации, предназначенной этому компьютеру, достаточно знать его адрес.

Давайте рассмотрим сценарий, определяющий передачу информации от одного компьютера к другому. Если некто хочет послать письмо другу, живущему в соседнем штате, он должен выполнить определенные действия. Отправитель письма должен положить его в конверт. На лицевой стороне конверта следует написать адрес. Затем письмо необходимо отнести в ближайшее почтовое отделение.

В почтовом отделении, получив письмо, прочитают адрес, чтобы определить, куда его следует отправить. Местное почтовое отделение отправляет письмо в почтовое отделение получателя. Затем почтальон приносит письмо адресату на дом.

Примечание

Не путайте протокольные адреса с физическими адресами. За физическую адресацию отвечает канальный уровень модели OSI. Физические адреса, в частности MAC-адрес, служат уникальными идентификаторами устройств и обычно назначаются аппаратно. Протокольные адреса, в свою очередь, назначаются протоколами и могут совпадать у различных устройств (желательно, чтобы эти устройства находились в разных сетях).

В то время как физический адрес устройства остается неизменным, протокольные адреса зависят от протоколов и могут меняться. Некоторые протоколы, например IPX, используют физический адрес как часть протокольного адреса. И все же не путайте их.

В нашем сценарии почтовое отделение соответствует маршрутизатору. Маршрутизатор читает адрес пункта назначения, присутствующий в каждом пакете, чтобы определить, куда должны отправиться данные. Но, как и почтовое отделение, маршрутизатор сможет прочитать адрес пункта назначения, только если он записан в определенном формате. Формат адреса определяется протоколом.

То есть в то время как маршрутизатор занимается доставкой информации из одного места в другое, протокол отвечает за то, чтобы эта информация была представлена в правильном формате.

Это поверхностное описание процесса маршрутизации выглядит не очень сложным. В действительности процесс не отличается от описанного. Но если бы все было так просто, не возникло бы необходимости в этой книге.

Сложность процессов, связанных с маршрутизацией, начинает проявляться, когда мы решаем добавить к базовому алгоритму метрики и другие правила. Дополнительные правила – это то, за что маршрутизацию считают трудной для понимания. Рассмотрим базовые правила, оставив пока в стороне все сложности; вернемся к ним позже.

Следующий раздел этой главы поясняет назначение протоколов с точки зрения маршрутизации. Одна из функций, выполняемых протоколами, заключается в инкапсуляции передаваемых данных. Это значительно облегчает процедуру маршрутизации.

Маршрутизация протокола

Прежде чем с головой окунуться в глубины маршрутизации (которая будет описана со стороны маршрутизатора), необходимо определить роль протоколов в этом процессе. Или точнее, что имеют в виду, когда говорят об инкапсуляции данных протоколом, и как инкапсуляция может помочь процессу маршрутизации? Чтобы ответить на эти вопросы, мы должны выяснить, что происходит с данными до того, как они попадают в маршрутизатор.

Когда такое устройство, как ПК, хочет посылать данные другому устройству, оно передает их драйверу протокола, находящемуся на ПК. Протокол выполняет инкапсуляцию данных и подготавливает их к отправке.

Инкапсуляция – один из ключевых факторов, делающих маршрутизацию возможной. Маршрутизаторы не могут перемещать данные произвольного формата. Информация должна быть структурирована таким образом, чтобы маршрутизатор мог легко определить, какого типа данные он получил и куда их следует отправить. Смысл инкапсуляции как раз и заключается в том, чтобы предоставить эти данные.

Инкапсулируя данные, протокол добавляет к ним адрес пункта назначения, чтобы маршрутизатор мог их туда послать.

Примечание

До того как данные инкапсулированы (и к ним добавлен адрес получателя), маршрутизатор не может узнать, куда их следует отправить.

Хотя протокол и предусматривает указание адреса доставки, маршрутизатору необходимо знать, что представляют собой данные. Когда человек получает письмо, он знает, что конверт, в котором оно пришло, не является частью посланной ему информации. Он служит лишь контейнером для самого письма. Инкапсуляция играет роль конверта, в котором маршрутизаторы пересылают данные.

Протокол форматирует данные таким образом, чтобы они помещались в определенный размер (как и в случае с конвертом). Благодаря этому маршрутизатор знает, что информация, отправляемая по указанному адресу, помещается в x бит. Все эти сведения хранятся в блоках данных, называемых заголовками протокола.

Заголовки протокола

Одна из обязанностей протокола, когда он получает поток данных для передачи, состоит в инкапсуляции. Инкапсулируя поток данных, протокол преобразует их в формат, который может быть легко понят любой системой, получившей эти данные. Но протокол не предусматривает того, что каждое устройство, встретившееся на пути данных, должно читать весь поток. Поэтому данные отформатированы так, что промежуточным устройствам достаточно прочитать небольшой фрагмент, чтобы понять, кому предназначен конкретный пакет.

Именно с этой целью протокол добавляет в поток данных заголовки, содержащие информацию, на основе которой любое устройство, поддерживающее этот протокол, может сделать заключение об инкапсулированных данных.

Примечание

В процессе инкапсуляции данных, полученных от устройства, протокол не изменяет их содержания. Заголовки вставляются в начале блоков данных, не изменяя их.

Не путайте инкапсуляцию с шифрованием. В результате шифрования данные преобразуются к такому виду, что ни одно устройство (за исключением адресата) не может прочитать содержимое пакета.

Проиллюстрируем назначение заголовков на примере полей IP-протокола. Одна из функций IP, когда он получает поток данных для передачи по сети, состоит в добавлении к данным заголовков. Из этих заголовков любое устройство, пропускающее поток (например, маршрути-

затор), может получить некоторые важные сведения об инкапсулированных данных. На рис. 5.1 показаны поля IP-заголовка.

Первое поле, «Версия», сообщает маршрутизатору, какая версия IP была использована для формирования пакета. На данный момент основной является версия IP-4, хотя IP-6 быстро становится реальностью (а некоторые компании уже приступили к тестированию IP-8). Следовательно, маршрутизаторы нуждаются в стандартном способе определения версии протокола, с которым они имеют дело.

Следующее поле называется «Длина заголовка». Согласно названию, оно содержит значение длины самого IP-заголовка. Получив длину заголовка, маршрутизатор может определить, где этот заголовок заканчивается и начинаются собственно данные.

Поле «Тип сервиса» (TOS, type of service) определяет приоритет инкапсулированного пакета. Хотя это и относится к дополнительным возможностям, многие маршрутизаторы Cisco способны выполнять так называемую «приоритетную маршрутизацию» (TOS routing) с учетом значения данного поля. Таким образом (хотя сейчас это поле нас не интересует) оно может оказывать влияние на процесс перемещения данных от устройства к устройству.

Поле «Общая длина» содержит значение длины всего инкапсулированного пакета. Пользуясь этим значением, маршрутизатор может определить длину блока данных, следующего за заголовком (то есть длина блока данных равна общей длине за вычетом длины заголовка).

Поле «Идентификатор» содержит уникальный номер, присваиваемый каждому пакету. Эти номера помогают принимающему устройству собрать вместе полученные пакеты.

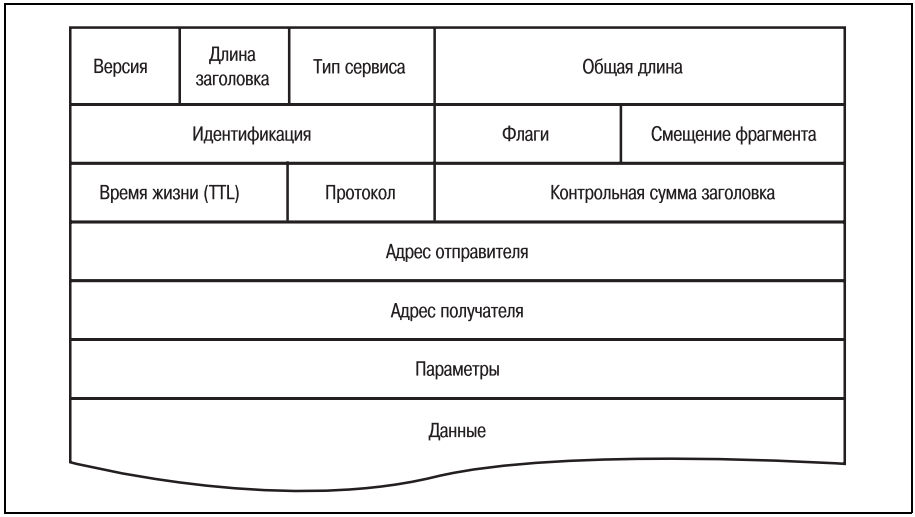

Рис. 5.1. *Поля IP-заголовка*

Следующее поле, «Флаг», используется при определении того, может ли пакет быть фрагментирован. Если может, то флаг определяет также, является ли данный пакет последним в последовательности фрагментов. Маршрутизаторы могут фрагментировать пакеты перед дальнейшей отправкой.

Маршрутизаторы Cisco могут быть настроены так, чтобы не маршрутизировать пакеты, имеющие размер, превышающий заданный. Если, например, какой-либо из каналов имеет низкую пропускную способность, маршрутизатор может отправлять в него только те пакеты, размер которых находится внутри определенного диапазона. Кроме того, маршрутизатор может быть сконфигурирован так, что будет приводить слишком длинные пакеты к требуемому размеру, деля их на части. В этом процессе используется поле флага. Однако при этом возникает необходимость в еще одном поле – поле смещения фрагмента.

Поле «Смещение фрагмента» тесно связано с полем «Флаг». Оно определяет, на каком байте заканчивается фрагментированный пакет. Это позволяет устройству-получателю правильно собрать фрагментированные пакеты (для последующей сборки потока данных).

Обычно маршрутизаторы передают пакет в течение ограниченного времени. То есть если маршрутизатор не может найти получателя пакета за указанный промежуток времени, он прекращает поиски. Поле «Время жизни» (TTL, Time to Live) используется для определения срока существования пакета. В нем указывается максимальная продолжительность попыток доставки пакета, по достижении которой пакет считается недоставленным. Когда маршрутизатор пытается отправить пакет, а адресат недоступен, он отмечает время, хранящееся в поле TTL, и начинает обратный отсчет. Если пакет не будет доставлен раньше, чем истечет его время жизни, он будет отброшен.

Примечание

Малое значение TTL может вызвать проблемы в больших сетях. Например, если пакет должен пройти через несколько маршрутизаторов (объединяющих несколько сетей), а значение TTL недостаточно велико, оно может исчерпаться прежде, чем пакет достигнет точки назначения. Поэтому даже нормально функционирующие системы могут не получить данные из-за низкого значения TTL.

Поле «Протокол» используется приемным устройством, а не маршрутизатором. Это поле сообщает получателю, какому протоколу передать данные по окончании их доставки и сборки.

Поле «Контрольная сумма заголовка» вычисляется по содержанию заголовка и служит для проверки его целостности. Проверяя контрольную сумму, маршрутизатор может гарантировать, что испорченные пакеты не будут отправлены далее.

Следующее поле «Адрес отправителя» – одно из наиболее важных для процесса маршрутизации. Оно сообщает маршрутизатору, кто отпра-

вил пакет. Этот адрес может быть использован для фильтрации и определения непрерывности маршрута. Например, маршрутизатор может быть настроен так, чтобы отправлять пакеты, пришедшие от одного источника в определенную сеть. Соответствие пакета этому критерию будет определяться по значению данного поля. Поле «Адрес получателя» расположено непосредственно после адреса отправителя и, безусловно, является самым важным для маршрутизации. Это поле определяет адрес принимающего устройства. Очевидно, что без адреса получателя маршрутизация была бы невозможна.

Поле «Параметры» содержит специфичные для IP параметры, и, наконец, в поле «Данные» помещается та информация, которая должна быть передана из одной системы в другую.

Большая часть информации, используемой для маршрутизации, хранится в заголовках пакетов. В заголовках передаются все данные, необходимые для доставки пакетов в пункт назначения. Маршрутизаторы получают из этих заголовков сведения, необходимые им для успешной пересылки данных.

Другое, не менее важное назначение инкапсуляции состоит в том, чтобы разделить поток данных на небольшие, легко управляемые порции. Такие пакеты облегчают чтение и обработку данных маршрутизаторами.

Пакетирование данных

Одной из функций протоколов является разделение потока данных в процессе инкапсуляции на пакеты такого размера, который облегчает манипулирование ими. Благодаря такому разбиению упрощается работа маршрутизаторов по пересылке больших объемов данных. Даже при сегодняшней кажущейся бесконечной пропускной способности отсутствие сегментирования больших объемов данных привело бы к многочисленным заторам.

Однако проблема трафика – не единственная проблема, решаемая с помощью инкапсуляции. Отправка данных небольшими (и, как правило, унифицированными) пакетами, увеличивает вероятность того, что приемное устройство останется доступным до окончания передачи и данные будут благополучно доставлены.

В процессе инкапсуляции данные, предназначенные для пересылки, «нарезаются» протоколом на фрагменты переменной длины. Затем каждый фрагмент, или пакет, маркируется и получает заголовок. Благодаря тому что каждый пакет снабжается заголовком, не обязательно отправлять все пакеты одновременно.

Примечание

Каждый протокол имеет свои, определенные спецификацией, размеры пакетов. При возникновении проблем в сети, просматривая пакеты, можно по их размерам определить тип протокола.

Когда пакеты достигают пункта назначения, система может прочитать все заголовки и определить последовательность пакетов (это поможет ей собрать их в правильном порядке). Затем из пакетов удаляются заголовки, и восстанавливается исходный поток данных.

Этот раздел должен был дать вам информацию, достаточную для понимания того, как протоколы способствуют процессу маршрутизации. Однако функции, предоставляемые маршрутизируемыми протоколами, составляют лишь одну треть от полной картины. Аппаратура маршрутизатора должна выполнить еще много работы, прежде чем данные смогут попасть к адресату.

Механизм маршрутизации

Вопреки распространенному мнению, маршрутизация сама по себе достаточно проста для понимания. Любая маршрутизация основывается на логических правилах и стратегиях. Сложность возникает тогда, когда используются уровни безопасности, вторичные и третичные протоколы, сложная топология сети.

В оставшейся части этой главы рассказывается о том, как происходит маршрутизация и как маршрутизаторы Cisco перемещают данные от системы к системе.

Маршрутизация в простой сети заключается в перемещении данных из одной сети в другую, в то время как в сложных сетях может потребоваться передача данных между несколькими маршрутизаторами, охватывающими несколько сетей. Существует значительная вероятность столкнуться с такими трудностями, как, например, множество метрик в сложной маршрутизируемой среде.

Поэтому давайте рассмотрим отдельно физический процесс маршрутизации, поскольку он применим ко всем случаям – и простым, и сложным.

Маршрутизация в простой сети

Простейший случай маршрутизации имеет место, когда данные должны перемещаться только между двумя сетями (рис. 5.2).

Обратите внимание на то, что рассматриваемая среда состоит из двух сетей (обозначенных A и B), каждая со своими собственными IP-адресами. В силу того что сети разделены физически и имеют разные схемы адресации, они не могут свободно обмениваться данными. Решение заключается в установке простого маршрутизирующего оборудования.

Примечание

Приведенный сценарий и решение для него являются очень упрощенными. Однако все маршрутизируемые сети (как простые, так и сложные) построены на тех же принципах.

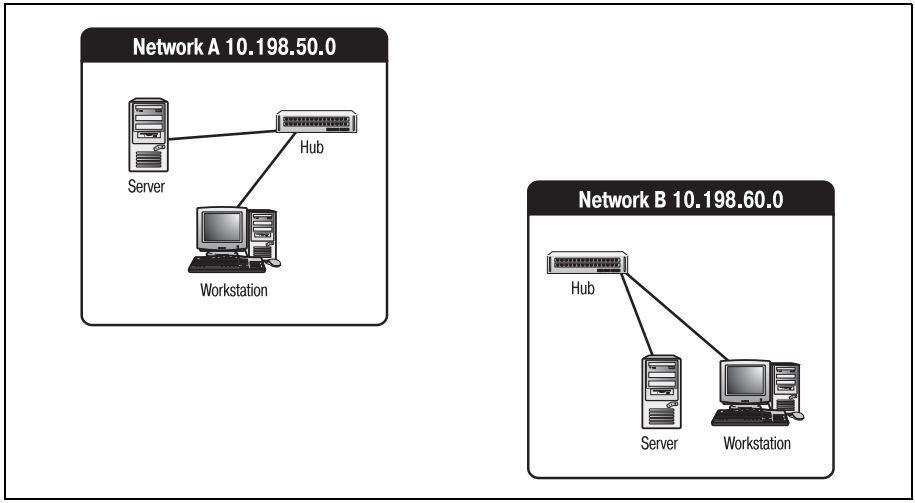

Рис. 5.2. *Простая сетевая среда*

Давайте рассмотрим работу маршрутизатора, помещенного между этими двумя сетями.

Персональным компьютерам и другим адресуемым устройствам (серверам и прочим) необходимо указать адрес нового маршрутизатора. Этот параметр обычно называется «шлюзом по умолчанию» (default gateway), но в некоторых системах название может быть другим. Однако независимо от названия этот параметр в любой системе должен содержать адрес маршрутизатора. На рис. 5.3 показаны те же сети с установленным между ними маршрутизатором.

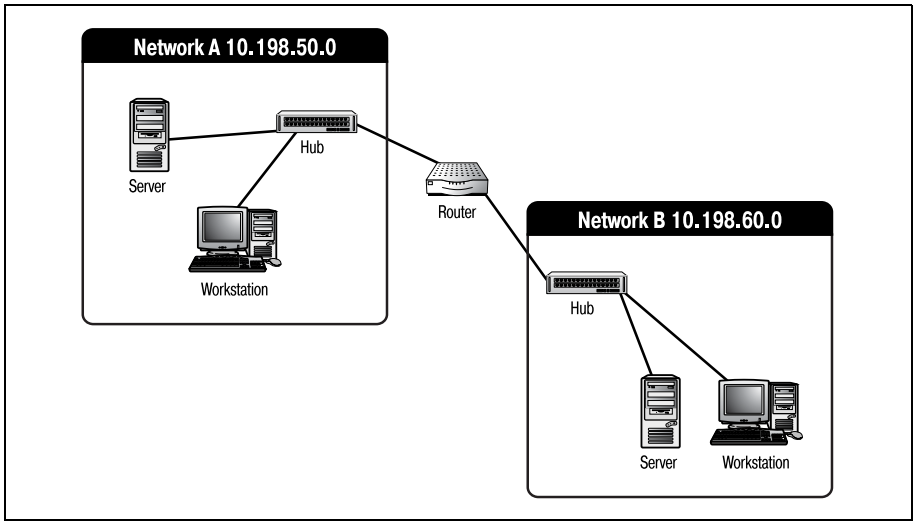

Рис. 5.3. *Две сети, объединенные маршрутизатором*

Обратите внимание: маршрутизатор соединен с концентратором в каждой из сетей. Маршрутизация имеет место не между отдельными системами, а между сетями. Следовательно, возникает необходимость в еще одном устройстве, соединяющем сеть с маршрутизатором.

Примечание

Хотя в этом примере и использованы концентраторы (hubs), предпочтение следует отдавать коммутаторам (switchs) , особенно в сильно загруженных сетях. Коммутаторы имеют возможность пересылки данных между заданными устройствами в отличие от концентраторов, выполняющих широковещательную рассылку пакетов всем устройствам сети.

Правильная установка маршрутизатора между двумя сетями может оказаться самой сложной частью работы. Дело в том, что каждому из интерфейсов маршрутизатора, соединяемого с сетью, необходимо назначить правильный адрес. Это значит, что интерфейс, соединяемый с сетью 10.198.50.0, должен получить адрес из диапазона 10.198.50.x, а интерфейс, соединяемый с сетью 10.198.60.0, – из диапазона 10.198.60.x.

Примечание

В большинстве сетей первые пять или десять адресов зарезервированы для маршрутизаторов. Хотя этого не требует ни один стандарт, большинство администраторов придерживаются такого соглашения.

Теперь, когда маршрутизатор установлен и на всех ПК указан его адрес, посмотрим, как данные будут перемещаться из сети A через маршрутизатор в сеть B.

Пусть компьютеры сети A (10.198.50.0) посылают данные друг другу через концентратор. Допустим, ПК с адресом 10.198.50.5 отправляет пакет по адресу 10.198.50.8. У него есть только один путь – к концентратору. Концентратор, будучи достаточно несложным устройством, просто ретранслирует полученный им пакет всем присоединенным к нему устройствам. В заголовке пакета указаны адреса отправителя и получателя. Каждое устройство, соединенное с концентратором, читает заголовок и проверяет, не ему ли послан данный пакет. Одним из этих устройств и будет 10.198.50.8. Когда оно получит сообщение, то обработает его должным образом и отправит ответ.

Остальные устройства прочитали заголовок и определили, что пакет предназначен не им, поэтому они просто проигнорировали его и продолжили ожидание сообщений.

При наличии маршрутизатора устройства сети A могут адресовать пакеты компьютерам сети B. Когда такой пакет, покинув отправившее его устройство, достигнет концентратора, то будет проигнорирован (по причине несовпадения адреса) всеми устройствами сети A. Но пакет попадет и в маршрутизатор, который (в отличие от остальных ком-

пьютеров сети A) не отбросит его сразу, а проанализирует, чтобы проверить, не известно ли ему местонахождение адресата.

Первым делом маршрутизатор откроет и прочитает заголовок пакета, откуда узнает адрес его получателя. С целью определения сети назначения (10.198.60.0) он сравнит полученный адрес (10.198.60.17) с маской подсети интерфейса, соединенного с сетью A.

Затем маршрутизатор просмотрит свои конфигурационные записи, чтобы проверить, не соответствует ли одна из них указанному сетевому адресу. Маршрутизатор обнаружит, что интерфейс 10.198.60.1 принадлежит сети 10.198.60.0. Теперь интерфейс 10.198.50.1 может передать пакет интерфейсу 10.198.60.1, который, в свою очередь, отправит его концентратору сети B.

Это весьма упрощенное описание того, как маршрутизатор перемещает данные между сетями и как в этом участвует протокол, в действительности довольно точно. Однако вряд ли вы встретите где-нибудь подобную сеть. Большинство имеющихся сегодня сетей используют сложные схемы маршрутизации.

Маршрутизация в сложной сети

Совокупность большого числа сетей и маршрутизаторов образует сложную систему, в которой между каждой парой сетей имеется более одного маршрута. Пример сложной маршрутизируемой сети приведен на рис. 5.4.

Предположим, что устройство, находящееся в сети A, отправляет данные устройству сети D. Маршрутизирующее оборудование выполняет следующую последовательность действий:

1. Инкапсулированный протоколом пакет достигает маршрутизатора A.
2. Маршрутизатор A просматривает таблицу маршрутов в поисках записи, определяющей местоположение сети D, и находит в ней два пути, ведущих к этой сети. Первый путь проходит через маршрутизаторы F и G, а второй – через маршрутизаторы H и G.
3. Маршрутизатор A, применяя алгоритм, реализованный в протоколе маршрутизации – в данном случае это протокол OSPF (Open Shortest Path First – протокол первоочередного открытия кратчайших маршрутов), – сравнивает метрики обоих путей и определяет, что путь, проходящий через маршрутизаторы F и G, является наилучшим (в смысле наикратчайшим).
4. Маршрутизатор A инкапсулирует пакет (уже инкапсулированный протоколом IP) в соответствии с протоколом OSPF. Затем он записывает в поле адреса назначения OSPF-заголовка адрес сети D.
5. Маршрутизатор A посылает OSPF-инкапсулированный пакет маршрутизатору F.

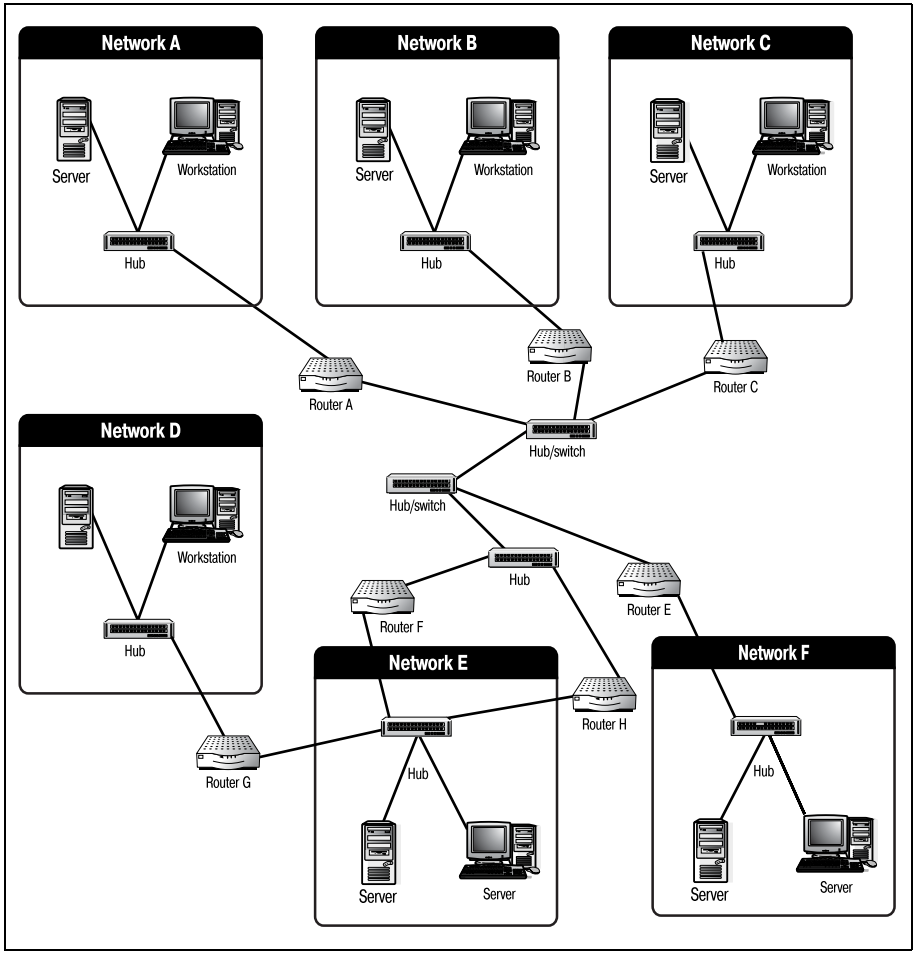

Рис. 5.4. *Сеть со сложной маршрутизацией*

6. Маршрутизатор F читает заголовки OSPF и видит, что пакет адресован маршрутизатору G.
7. Маршрутизатор F просматривает свою таблицу маршрутов, выясняет, что у него имеется физическое соединение с маршрутизатором G, и пересылает пакет ему.
8. Маршрутизатор G получает пакет, удаляет OSPF-заголовок и пересылает IP-пакет в сеть D.

Этот сценарий мог бы быть и более сложным. Во многих сетях применяются списки доступа и множественные метрики (оба понятия будут рассмотрены в этой книге позже).

Как бы то ни было, этот пример является хорошей иллюстрацией того, как происходит маршрутизация. Понимание этого процесса поможет

вам правильно выбрать параметры для настройки маршрутизатора Cisco в различных ситуациях.

В обоих сценариях, как в простом, так и в сложном, упоминалась таблица маршрутов. Эта таблица – своего рода база данных, размещенная в памяти маршрутизатора и содержащая всю информацию, необходимую для поиска местонахождения адресуемой сети.

Таблицы маршрутов

Таблицу маршрутов можно рассматривать как маленькую базу данных. Но описать ее непросто, так как формат записей в ней изменяется в зависимости от используемого протокола.

Поэтому здесь мы обсудим только общие элементы, содержащиеся в таблицах маршрутов. Частные случаи, связанные с использованием конкретных протоколов (например, RIP), будут рассмотрены в соответствующих главах.

Основная информация, хранящаяся в таблице маршрутов, заключается в сведениях о соответствии сетей и маршрутизаторов, записанных в форме «один к одному». Например, фрагмент таблицы маршрутов маршрутизатора A (из сценария, изображенного на рис. 5.4) может выглядеть так:

```
Network A - ME
Network B - Router B
Network C - Router C
Network D - Router F
Network D - Router H
Network E - Router F
Network E - Router H
Network F - Router E
```

При внимательном рассмотрении вы можете обнаружить одно несоответствие между таблицей и схемой, приведенной на рис. 5.4. Маршрутизатор G служит шлюзом сети D, но с точки зрения маршрутизатора A сеть D обслуживают маршрутизаторы F и H. Причина кроется в том, что большинство протоколов маршрутизации не позволяют маршрутизаторам ссылаться на устройства, непосредственно с ними не связанные. Другими словами, отсутствие физического соединения между маршрутизаторами G и A заставляет последний считать, что путь к сети D лежит только через маршрутизаторы F и H.

Маршрутизатор A знает о существовании сети D, но его не интересует, как в нее попасть. Все, что ему надо знать – это то, кому надо отправить пакеты, адресованные в сеть D. В данном случае A может передать пакеты любому из маршрутизаторов F или H, в таблицах маршрутов которых есть записи вида:

```
Network D - Router G
```

При такой организации таблиц отдельный маршрутизатор не несет ответственности за всю сеть. Каждый из них отвечает только за свой небольшой участок (в пределах своих физических соединений). Использование полной таблицы маршрутов для всей сети на каждом маршрутизаторе повлекло бы за собой ряд проблем. Во-первых, это размер самой таблицы. По мере разрастания таблицы маршрутов поиск в ней будет требовать все больше и больше времени, что замедлит обработку данных. Кроме того, хранение такой таблицы потребует дополнительной памяти, необходимой программам маршрутизации и конфигурационным файлам.

Вторая проблема, связанная с хранением полной таблицы, носит название *конвергенции* (*convergence*). Конвергенцией называют такое состояние сети, в котором таблицы маршрутов всех устройств не противоречат друг другу. Целью протокола маршрутизации является наискорейшее достижение конвергенции при изменении данных в таблице. Если каждый маршрутизатор будет хранить полную таблицу, то изменения, сделанные на одном конце сети, могут не скоро попасть в таблицы маршрутов на другом ее конце. Замедление конвергенции приведет к возникновению петель и разрывов в маршрутах, в результате чего данные могут быть утеряны.

Таблицы маршрутов выполняют еще одну важную функцию: хранение и обновление *метрик маршрутов* (*routing metrics*). На основании метрик делается выбор в пользу того или иного маршрута. То есть метрики, сопоставленные каждому из маршрутов в таблице, используются для расчетов, на основании которых принимается решение, по какому из маршрутов, ведущих к одному и тому же адресу, следует отправить пакет.

В сценарии, показанном на рис. 5.4, маршрутизатор A имеет выбор из двух маршрутов, по которым он может отправить пакеты в сеть D. Можно послать данные маршрутизатору F, а можно и маршрутизатору H – оба пути, в конце концов, приведут к сети D. Для принятия решения маршрутизатор A использует метрики, присвоенные каждому из путей.

Значения метрик зависят от используемого протокола. Но для всех протоколов одна характеристика метрик остается неизменной: 99% из них – это произвольные значения, присвоенные администратором сети или специалистом по маршрутизаторам.

Метрики типа «стоимости», назначаемые администраторами, определяют относительную стоимость использования данного маршрута по отношению к другим. Существует множество факторов, способных повлиять на решение администратора присвоить меньшее значение одному пути и большее – другому. Выбор может зависеть от объема трафика, надежности оборудования, стоимости (в денежном выражении) канала. Однако конкретные значения определяются только администратором.

В нашем примере путь через маршрутизатор H может иметь метрику 200, а через маршрутизатор F – метрику 100. Когда маршрутизатор A проверит все метрики (а в некоторых протоколах их может быть десяток), он рассчитает окончательную стоимость маршрута. Путь с наименьшим значением будет выбран в качестве наилучшего.

В этом разделе уже упоминалась конвергенция. Однако эта тема заслуживает более подробного рассмотрения, поэтому давайте поближе познакомимся с этим важным понятием.

Достижение конвергенции

Конвергенция маршрутов в сети – такая же неотъемлемая часть процесса маршрутизации, как и все рассмотренные выше факторы. Проще говоря, если каждый маршрутизатор получит собственную, отличную от остальных таблицу маршрутов, то и пересылку данных он будет выполнять своим уникальным способом. Это вызовет множество проблем, и сеть, скорее всего, перестанет работать.

Следовательно, система должна быть организована так, чтобы все маршрутизаторы некоторой сети работали с одной и той же таблицей. В результате достигается конвергенция маршрутизаторов.

Ответственность за достижение конвергенции маршрутизаторов возложена на протокол маршрутизации. Поэтому способов достижения конвергенции столько же, сколько протоколов. Однако базовая концепция этого процесса остается неизменной.

Чтобы добиться конвергенции, каждый маршрутизатор сети посылает свою часть таблицы маршрутов всем окружающим маршрутизаторам. Получив порцию изменений, каждый маршрутизатор обновляет свою таблицу и посылает изменения дальше. Этот процесс продолжается до тех пор, пока все маршрутизаторы не получат одни и те же данные о сети.

Примечание

По мере рассмотрения различных протоколов на страницах этой книги мы будем обсуждать соответствующие особенности достижения конвергенции.

Маршрутизируемые протоколы и протоколы маршрутизации

При изложении материала этой главы мы использовали термины «маршрутизируемый протокол» и «протокол маршрутизации». Когда в сети находится несколько маршрутизаторов, оба протокола необходимы и оба одинаково важны.

Скорее всего, вы уже хорошо знакомы с маршрутизируемыми протоколами. К ним относятся такие распространенные в локальных и глобаль-

ных сетях протоколы, как TCP/IP и Frame Relay. Пакеты этих протоколов перемещаются между маршрутизаторами в пакетах, сформированных протоколами маршрутизации. Помимо этого пакеты маршрутизирующих протоколов используются для передачи изменений в маршрутах и значений метрик. Без использования протоколов маршрутизации информация, передаваемая маршрутизируемыми протоколами, не смогла бы перемещаться от маршрутизатора к маршрутизатору.

Примечание

При наличии в сети единственного маршрутизатора не возникает необходимости в применении протокола маршрутизации. То есть эти протоколы переносят данные только между маршрутизаторами. Следовательно, если в вашей сети только один маршрутизатор, то протокол маршрутизации вам не потребуется.

Теперь, познакомившись с концепциями маршрутизации, как с точки зрения протоколов, так и с точки зрения оборудования, мы сможем в следующей главе «испачкать руки». Это значит, что в ней мы будем заниматься конфигурированием маршрутизатора Cisco «с нуля».

Резюме

Маршрутизаторы Cisco (как и все прочие маршрутизаторы) нуждаются в использовании протоколов для перемещения данных в процессе маршрутизации. Для перемещения данных протоколы используют метод инкапсуляции.

В процессе инкапсуляции протоколы форматируют данные по стандартному шаблону. Протоколы добавляют к данным информацию, необходимую для маршрутизации, в форме заголовков пакетов.

Вся информация о маршрутах, доступная маршрутизатору, хранится в простой базе данных, называемой таблицей маршрутов. Таблица маршрутов содержит такую информацию, как адрес назначения и местоположение сетей, в которые маршрутизатор может доставлять данные.

Маршрутизаторы обмениваются данными о маршрутах, рассылая обновления. Обновления маршрутов обеспечивают единую систему маршрутов в сети. Когда все маршрутизаторы имеют одинаковую картину сетевого окружения, достигается состояние конвергенции.

Вопросы и ответы

Вопрос Почему маршрутизаторы могут передавать данные только на сетевом уровне модели OSI?

Ответ Сетевой уровень модели OSI отвечает за адресацию сетевых устройств. Без использования таких адресов доставка любой информации в заданное место практически невозможна.

Вопрос Необходимы ли метрики маршрутов и почему?

Ответ Метрики маршрутов помогают маршрутизаторам определить наилучший путь между двумя сетями. В отличие от статичного описания маршрутов, использование метрик позволяет маршрутизаторам реагировать на изменения в сетевом окружении.

Тест

Вопросы

1. Каковы две основные функции маршрутизируемых протоколов?
2. Какое поле IP-заголовка содержит значение времени жизни пакета?
3. Какой информацией должны обмениваться маршрутизаторы, чтобы достичь конвергенции?

Ответы

1. Адресация и инкапсуляция пакетов.
2. TTL (время жизни).
3. Таблица маршрутов.

6

Запуск маршрутизатора и работа с ним

В этой главе мы рассмотрим одну из самых важных процедур в настройке и сопровождении маршрутизатора – конфигурирование. Конфигурация является основой (сердцем или умом) функциональности маршрутизатора. Конфигурация вашего маршрутизатора определяет практически все его параметры: от имен и адресов его портов до протоколов, которые он может распознавать.

Чтобы в полной мере изучить процесс конфигурирования, исследуем такие темы:

- Предварительное конфигурирование
- Конфигурационные файлы
- Реконфигурирование маршрутизатора
- Изменение отдельных элементов конфигурации
- Просмотр и конфигурирование интерфейсов
- Конфигурирование баннеров

Процесс конфигурирования может быть двух видов: *предварительное конфигурирование* (с нуля) и *реконфигурирование*. Предварительное конфигурирование проводится, если ранее на маршрутизаторе не было установлено никакой конфигурации. То есть маршрутизатор новый и никогда не использовался или предыдущая конфигурация была полностью уничтожена. Реконфигурирование же изменяет уже существующую конфигурацию. Несмотря на то что во время выполнения этих двух процессов маршрутизатору могут поставляться одни и те же элементы информации, доступ к ним организован по-разному и выполняются они также по-разному.

В этой главе (и в большей части оставшихся глав) в качестве примера для конфигурирования будет использоваться Cisco 1605R с IOS версии 12.0(3). Этот маршрутизатор имеет два встроенных порта Ethernet (Eth0 и Eth1), порт расширения WAN и консольный порт. Такой маршрутизатор экономически эффективен, поэтому, если вы подумываете о том, чтобы купить маршрутизатор для тренировок, я предложил бы 1605R. Однако большинство примеров, приведенных в данной книге, будут работать на любом маршрутизаторе Cisco (особенно если на нем работает IOS версии 12.0(3)).

Примечание

Если у вас нет возможности попрактиковаться с маршрутизатором, попробуйте выполнить упражнения, которые находятся в конце каждой главы, на бумаге. Эти упражнения спланированы так, чтобы помочь вам запомнить основные моменты, представленные в главе. Ответы на упражнения приводятся после собственно постановки задачи и выделены так же, как тот текст, который вы читаете сейчас.

Давайте поговорим о первом процессе конфигурирования – предварительном конфигурировании.

Предварительное конфигурирование

Купив новый маршрутизатор Cisco, вы получаете его без установленных правил маршрутизации. Дело в том, что Cisco ничего не знает об архитектуре и схеме адресов вашей конкретной сети, поэтому конфигурировать маршрутизатор на заводе не имеет смысла. Следовательно, первое, что необходимо сделать после того как маршрутизатор включен, – это установить начальную конфигурацию.

Предварительное конфигурирование состоит в ответе на ряд вопросов. Эти ответы и определят конфигурацию маршрутизатора. Но IOS не может предугадать и предложить вам все возможные варианты конфигураций в своих подсказках. Поэтому установка конфигурации (называемая базовой настройкой управляющих параметров) предназначена для предоставления только той информации, которой достаточно, чтобы устройство заработало. Когда же маршрутизатор уже работает, вы можете задать детали требуемой вам конфигурации.

Прежде чем говорить о конфигурировании маршрутизатора, необходимо знать, как подсоединить терминал или ПК таким образом, чтобы они могли взаимодействовать с маршрутизатором. Если вы никогда ранее не работали с маршрутизатором Cisco, уделите пристальное внимание этому разделу.

Подключение к маршрутизатору

Если вы первый раз имеете дело с маршрутизаторами Cisco (или с маршрутизаторами вообще), то наверняка обратили внимание на нечто

странное на задней стенке маршрутизатора: отсутствие порта для монитора или клавиатуры. И это не недоразумение. Большинство маршрутизаторов (в том числе все маршрутизаторы Cisco) не имеют возможности управлять собственными мониторами и клавиатурами, поэтому доступ к ним осуществляется с внешнего устройства.

Продолжая исследовать заднюю панель маршрутизатора, вы заметите консольный порт – чаще всего гнездо разъема RJ-45, но на некоторых старших моделях это может быть и последовательный порт DB9. Как уже говорилось во второй главе, консольный порт является портом прямого доступа, который необходимо обнаружить, чтобы получить доступ к маршрутизатору.

Если вы работаете с новым маршрутизатором (таким, который еще ни разу не конфигурировали), то консольный порт будет единственным активным портом, при помощи которого вы сможете подключиться к маршрутизатору.

Примечание

Для получения доступа к маршрутизатору Cisco с установленной конфигурацией можно использовать Telnet для соединения с одним из Ethernet-портов. Но точно такие же возможности должны быть доступны через консольный порт. Так что если сомневаетесь, используйте порт консоли.

Консольный порт, вне зависимости от типа физического порта (RJ-45 или DB9), всегда действует как последовательный. Поэтому если вы используете ПК для подключения к маршрутизатору, необходимо подключить консольный порт маршрутизатора к последовательному порту ПК. Это можно сделать с помощью входящего в комплект поставки консольного «rollover»-кабеля Cisco и адаптера RJ-45/DB9.

Примечание

Различные виды кабелей, которые можно подключать к разъему RJ-45 – перекрестный, «rollover» и стандартный кабель Ethernet, – имеют ряд важных отличий.

Стандартный кабель Ethernet, используемый для подключения Ethernet-порта компьютера к концентратору, состоит из восьми разноцветных проводов, ведущих к восьми контактам. Если сложить два конца стандартного кабеля Ethernet вместе, то можно заметить взаимно однозначное соответствие между восемью цветными проводами с одного и другого конца кабеля. Соотношения между контактами кабеля Ethernet представлены в табл. 6.1.

Таблица 6.1. Контакты стандартного кабеля Ethernet

Контакты Ethernet	Функция
1 ‡ 1	прием
2 ‡ 2	прием
3 ‡ 3	передача

Контакты Ethernet	Функция
4 ‡ 4	не используется
5 ‡ 5	не используется
6 ‡ 6	передача
7 ‡ 7	не используется
8 ‡ 8	не используется

Перекрестный (*crossover*) кабель Ethernet – это нечто другое. Обычно он используется для соединения Ethernet-портов друг с другом напрямую, а не через сеть. Перекрестный кабель меняет местами провода 3 и 1 и провода 6 и 2. Контакты перекрестного кабеля Ethernet представлены в табл. 6.2.

Таблица 6.2. Контакты перекрестного кабеля Ethernet

Контакты Ethernet	Функция
1 ‡ 3	прием ‡ передача
2 ‡ 6	прием ‡ передача
3 ‡ 1	передача ‡ прием
4 ‡ 4	не используется
5 ‡ 5	не используется
6 ‡ 2	передача ‡ прием
7 ‡ 7	не используется
8 ‡ 8	не используется

Поставляемый Cisco *rollover*-кабель является уникальным. Все восемь проводов кабеля переворачиваются для создания зеркального отображения. Затем добавляется адаптер RJ-45/DB9. Контакты консольного (*rollover*) кабеля Ethernet и их соединение с разъемом DB9 приведены в табл. 6.3.

Таблица 6.3. Контакты консольного кабеля и DB9

Контакты Ethernet	Адаптер Ethernet – DB9 (контакт 9 не используется)	Функция
1 ‡ 8	8 ‡ 8	прием ‡ передача
2 ‡ 7	7 ‡ 6	не используется
3 ‡ 6	6 ‡ 2	передача ‡ прием
4 ‡ 5	5 ‡ 5	земля
5 ‡ 4	4 ‡ 5	земля
6 ‡ 3	3 ‡ 3	прием ‡ передача
7 ‡ 2	2 ‡ 4	не используется
8 ‡ 1	1 ‡ 7	передача ‡ прием

После того как ПК подключен к консольному порту маршрутизатора, вы получаете возможность общаться с маршрутизатором, используя стандартную программу эмуляции терминала.

Примечание

Если при попытке взаимодействия возникли проблемы с эмулятором терминала, убедитесь, что вы используете для консольного порта следующие значения параметров по умолчанию:

```
Baud: 9600
DataBits: 8
Parity:  None
StopBits: 1
Flow Control: Hardware
```

Получив работающее соединение с консольным портом маршрутизатора, перейдем к обсуждению предварительного конфигурирования. Соединение с консолью будет нашим «окном» в меню конфигурации, поэтому следовало установить его до того, как приступать к установке каких бы то ни было параметров.

Диалог конфигурирования системы

После того как новый маршрутизатор распакован (и зарегистрирован), следует выбрать для него такое место, на котором он будет впоследствии установлен и начнет работать в сети. Тогда вам не придется двигать уже сконфигурированный маршрутизатор (а это могло бы привести к нарушениям в работе сети). Еще одной утилитарной причиной размещения маршрутизатора в его окончательном «рабочем» местоположении является то, что вы получаете доступ к портам и кабелям, которые в конечном счете будут подключены к маршрутизатору.

С помощью эмулятора терминала вы можете наблюдать за процессом загрузки маршрутизатора. По завершении загрузки маршрутизатор обнаруживает, что не установлено никакой конфигурации, и входит в диалог конфигурирования. В ходе запуска (от загрузки до входа в режим конфигурирования) на терминал выводятся следующие сообщения:

```
System Bootstrap, Version 12.0(3)T, RELEASE SOFTWARE (fc1)
Copyright (c) 1999 by cisco Systems, Inc.
C1600 platform with 8192 Kbytes of main memory

program load complete, entry point: 0x4020060, size: 0x165eac

%SYS-6-BOOT_MESSAGES: Messages above this line are from the boot loader.
program load complete, entry point: 0x2005000, size: 0x2199cc
Self decompressing the image : ########################################
################################### [OK]
              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
```

```
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706


Cisco Internetwork Operating System Software
IOS (tm) 1600 Software (C1600-Y-M), Version 12.0(3), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Mon 08-Feb-99 20:15 by phanguye
Image text-base: 0x02005000, data-base: 0x02455958

cisco 1605 (68360) processor (revision C) with 7680K/512K bytes of memory.
Processor board ID 17531816, with hardware revision 00000002
Bridging software.
X.25 software, Version 3.0.0.
2 Ethernet/IEEE 802.3 interface(s)
System/IO memory with parity disabled
8192K bytes of DRAM onboard
System running from RAM
8K bytes of non-volatile configuration memory.
4096K bytes of processor board PCMCIA flash (Read/Write)
         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]:
```

Обратите внимание на последнюю выведенную строку. Здесь новый маршрутизатор остановится на неограниченное время, ожидая вашего ответа на вопрос: «Would you like to enter the initial configuration dialog?» (Хотите ли вы войти в режим начального конфигурирования?). Если маршрутизатор не приглашает вас войти в диалог начального конфигурирования и не ждет вашего ответа, то это означает, что на нем уже установлена конфигурация. Конфигурирование маршрутизатора с уже имеющейся конфигурацией будет описано в следующем разделе «Реконфигурирование маршрутизатора».

Примечание

Откуда Cisco IOS знает, первый ли раз включается маршрутизатор? Она этого и не знает. Во время загрузки IOS проверяет наличие файла под названием `startup config`. Этот файл хранит всю конфигурационную информацию для маршрутизатора.

Если файл существует, маршрутизатор читает его и загружает конфигурацию. Если же файла нет, то маршрутизатор понимает, что текущей конфигурации не задано, и входит в режим конфигурирования.

Чтобы продолжить конфигурирование, ответьте `yes` на заданный вопрос.

Примечание

Если вы ответите no, то выйдете из диалога конфигурирования системы и войдете в пользовательский режим. Маршрутизатор не будет сконфигурирован.

Получив от пользователя ответ yes, маршрутизатор начинает диалог, состоящий из последовательности вопросов, устроенной так, чтобы собрать достаточно информации для того, чтобы устройство заработало. Конфигурирование маршрутизатора посредством участия в таком диалоге называется базовой настройкой управляющих параметров.

Давайте приступим к рассмотрению базовой настройки и установим конфигурацию, необходимую для работы маршрутизатора Cisco.

Базовая настройка параметров

После того как вы ответите yes и войдете в диалог начального конфигурирования, Cisco IOS выведет следующее сообщение:

```
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system
Would you like to enter basic management setup? [yes/no]:
```

Посмотрите на последнее сообщение перед вопросом. Программное обеспечение маршрутизатора напоминает вам, что при базовом конфигурировании собирается только минимальная информация, также упоминается *extended setup* (расширенная настройка), являющаяся более подробной. Можно предположить, что, введя no в ответ на вопрос: «Would you like to enter basic management setup?» (Хотите ли вы войти в базовую настройку?), вы войдете в диалог расширенного конфигурирования. Так и есть, и мы поговорим об этом чуть позже в этом же разделе. Пока же сосредоточимся на базовой настройке системы.

Вводим yes, чтобы приступить к базовой настройке. В ответ маршрутизатор запрашивает первую порцию информации, которая должна быть установлена на маршрутизаторе:

```
Would you like to enter basic management setup? [yes/no]: y
Configuring global parameters:

  Enter host name [Router]:
```

Первое, о чем спрашивает маршрутизатор, – это имя, которое ему будет присвоено. Другие устройства, в том числе другие маршрутизаторы Cisco, будут использовать это имя для ссылок на конфигурируемое вами устройство.

Теоретически можно обращаться к маршрутизатору по его протокольному адресу (например, по его IP-адресу, если маршрутизатор использует протокол IP). Однако есть три причины, по которым можно не захотеть пользоваться такой возможностью:

1. Маршрутизатор еще не запрашивал у вас адрес, следовательно, он не знает своего адреса.
2. Маршрутизаторы часто имеют несколько портов с адресами, так что в качестве имени придется выбрать один из них.
3. Имена запоминать и распознавать гораздо легче, чем длинные строки цифр, образующих адреса.

Выбирая имя для маршрутизатора, постарайтесь использовать что-то описательное, чтобы было легко искать устройство в списках. Например, `MyCorp1` или `Floor4A` подходят больше, чем `Router1`. Однако в сегодняшнем озабоченном безопасностью мире имя устройства, подобное `MainSecurityFirewallForTheOfficesFinancialCenter`, будет все же слишком описательным (кроме того, длина имени ограничена 30 символами).

Когда вы определитесь с именем, введите его и нажмите клавишу <Enter>. Диалог конфигурирования перейдет к следующему вопросу:

```
Enter host name [Router]: STYSCisco

The enable secret is a password used to protect access to
privileged EXEC and configuration modes. This password,
after entered, becomes encrypted in the configuration.
Enter enable secret:
```

Теперь маршрутизатор предлагает ввести пароль «enable secret». Как говорится в сообщении, это секретный пароль, обеспечивающий безопасность доступа в привилегированный режим работы командного интерпретатора.

Примечание

Как только установлен секретный пароль, пользователь, пытающийся войти в привилегированный режим, увидит на своем экране следующее:

```
Router>enable
Password:
```

Команда `enable` по-прежнему переводит маршрутизатор в привилегированный режим, но пользователь должен ввести пароль, чтобы получить возможность работать в этом режиме.

Еще одной интересной особенностью пароля `enable secret` является то, что он хранится в Cisco IOS зашифрованным, в то время как все остальные пароли хранятся открытыми. (Всего в Cisco IOS существует пять паролей, но в процессе базовой настройки будут запрошены еще только два.) Каждый, кто обращается к вашему маршрутизатору с нечестными намерениями, теоретически может найти все остальные ва-

ши пароли в расшифрованном виде в IOS. Но им придется потрудиться, пытаясь расшифровать секретный пароль.

Почему секретный пароль защищен, а остальные – нет? Дело в том, что любое конфигурирование маршрутизатора производится в привилегированном режиме, поэтому если вы можете заблокировать доступ в этот режим, то знание других паролей не поможет потенциальным злоумышленникам.

Примечание

Хотя пароль в IOS и шифруется, но во время настройки он присутствует на экране в открытом виде. Поэтому у вас может возникнуть желание очистить экран после конфигурирования.

Выбор простого для запоминания секретного пароля убережет вас от дополнительной головной боли – восстановления пароля. После того как секретный пароль задан, будет запрошен пароль `enable`:

```
The enable password is used when you do not specify an
enable secret password, with some older software versions, and
some boot images.
Enter enable password: Enable
```

IOS сообщает, что пароль `enable` используется тогда, когда нет секретного пароля. Помните, что нельзя ввести пустой пароль в диалоге конфигурирования. Можно позже войти в режим конфигурирования и создать нулевой пароль.

Последний пароль, который требуется задать в диалоге конфигурирования, – пароль виртуального терминала.

```
The virtual terminal password is used to protect
access to the router over a network interface.
Enter virtual terminal password: vterm
```

Пароль виртуального терминала применяется для защищенного доступа по Telnet к портам маршрутизатора Cisco. Пользователи могут получить доступ к сетевым маршрутизаторам Cisco, обращаясь по Telnet к специальному порту маршрутизатора. Например, если маршрутизатор Cisco был сконфигурирован так, чтобы соединять две сети, то пользователи одной сети могут подключиться к маршрутизатору по Telnet через порт, соответствующий этой конкретной сети.

Пароль виртуального терминала обеспечивает безопасность доступа к маршрутизатору через такой порт. После того как пароль установлен, пользователи, обращающиеся к виртуальному терминалу, должны будут ввести подходящий пароль виртуального терминала, чтобы получить доступ.

Примечание

Как видите, маршрутизаторы Cisco имеют множество различных паролей (всего их пять, но два не задаются в рамках диалога конфигурирования). Очень важно поддерживать их в порядке и понимать различия между ними.

После того как определен пароль виртуального терминала, маршрутизатор спрашивает, хотите ли вы использовать протокол SNMP (Simple Network Management Protocol – простой протокол сетевого управления).

```
Enter virtual terminal password: vterm
  Configure SNMP Network Management? [yes]:
```

Заметьте, что ответом по умолчанию является `yes`. Я склоняюсь к ответу `no`. Если у вас нет непреодолимого желания ответить `yes`, просто пропустите этот вопрос. Обычно SNMP используется только при наличии нескольких маршрутизаторов, когда администратору необходимо контролировать состояние разнообразных статистических данных маршрутизатора. (Даже если окажется, что вам необходимо установить SNMP для порта, вы можете сделать это после завершения диалога конфигурирования. SNMP относится к более сложным элементам маршрутизации и не будет подробно рассматриваться в настоящем издании.) Ответ `no` продолжит ведение диалога:

```
  Configure SNMP Network Management? [yes]: n

Current interface summary
Any interface listed with OK? value "NO" does not have a valid configuration
Interface   IP-Address      OK? Method Status      Protocol

Ethernet0   unassigned      NO  unset  up            down
Ethernet1   unassigned      NO  unset  up            down

Enter interface name used to connect to the
management network from the above interface summary:
```

Теперь маршрутизатор выводит текущий список физических портов вашего маршрутизатора и их статусы. Так как вы используете маршрутизатор в первый раз, для всех портов под «OK?» должно стоять «NO», а под «Protocol» – «down». Это означает, что указанные порты не прошли конфигурирование и для них не указан никакой активный протокол. То есть в их текущем состоянии интерфейсы не являются рабочими.

Для того чтобы интерфейс маршрутизатора был полностью функциональным, необходимо выполнение двух условий. Во-первых, интерфейс должен быть включен. По умолчанию все интерфейсы маршрутизатора Cisco выключены. (Продолжив работу с диалогом конфигурирования, вы включите нужные интерфейсы.)

Примечание

Включение интерфейса маршрутизатора – это программное, а не аппаратное действие. С помощью Cisco IOS вы можете указать маршрутизатору, должен ли определенный интерфейс быть включен или выключен.

Второе условие полноценного функционирования интерфейса маршрутизатора заключается в установке протокола, который будет использовать данный интерфейс. Так как маршрутизация полностью зависит от использования протоколов, интерфейс Cisco не будет функциональным («up»), если для него не сконфигурирован протокол. Отвечая на дальнейшие вопросы диалога конфигурирования, вы предоставите маршрутизатору информацию, необходимую для приведения интерфейса в рабочее состояние.

Будет задан вопрос об имени интерфейса (физического порта), который вы хотите сконфигурировать. Рассмотрим пример конфигурирования интерфейса «Ethernet0».

```
Enter interface name used to connect to the
management network from the above interface summary: Ethernet0

Configuring interface Ethernet0:
  Configure IP on this interface? [yes]:
```

Теперь предлагается указать, будет ли на выбранном интерфейсе применяться протокол IP. Если ответить `no`, то установка будет завершена и интерфейсы маршрутизатора останутся несконфигурированными. Так как вы, вероятно, хотите иметь хотя бы один функционирующий интерфейс, вам следует ответить `yes`.

```
Configuring interface Ethernet0:
  Configure IP on this interface? [yes]: y
    IP address for this interface:
```

Затем необходимо сообщить диалогу IP-адрес и маску подсети для интерфейса (об этом говорилось в главе 5):

```
IP address for this interface: 10.162.24.153
    Subnet mask for this interface [255.0.0.0] : 255.0.0.0
```

После того как IP-информация для конфигурируемого интерфейса предоставлена, диалог завершается и выводится сводка конфигурации.

```
Class A network is 10.0.0.0, 8 subnet bits; mask is /8

The following configuration command script was created:

hostname STYSCisco
enable secret 5 $1$ohv2$phQNY94qZoJAZ2h7XBigG0
enable password enable
line vty 0 4
```

```
password vterm
no snmp-server
!
no ip routing
!
interface Ethernet0
no shutdown
ip address 10.162.24.153 255.0.0.0
!
interface Ethernet1
shutdown
no ip address
!
end

[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

Enter your selection [2]:
```

Cisco IOS просит подтвердить выбор конфигурации, изменить конфигурацию или просто прерваться и выйти (выбор варианта 2 означает сохранение вашей конфигурации и перезагрузку маршрутизатора). Если вам кажется, что в процессе конфигурирования была допущена ошибка, вы можете заново выполнить конфигурирование, выбрав ответ 0. Все значения, выбранные вами в первый раз, будут предложены во второй раз в качестве ответов по умолчанию.

Обратите внимание на то, что в ходе базовой установки вам было разрешено сконфигурировать только один интерфейс, хотя у маршрутизатора имеется несколько портов. В подобном выходе из диалога после конфигурирования одного интерфейса состоит отличие между диалогами базового и расширенного конфигурирования. При расширенном конфигурировании вы имеете возможность установить столько интерфейсов, сколько потребуется.

Диалог расширенного конфигурирования

Обратимся к диалогу расширенного конфигурирования. Чтобы войти в него, ответьте no на приглашение IOS войти в диалог базовой установки.

```
Would you like to enter basic management setup? [yes/no]: no

First, would you like to see the current interface summary? [yes]:
```

IOS сразу же спрашивает, хотите ли вы просмотреть текущие интерфейсы. Выбрав ответ yes, вы увидите тот же список интерфейсов, который был выведен при базовом конфигурировании, когда вы ответили no на предложение об установке SNMP.

После того как выведен текущий список интерфейсов, IOS запрашивает ту же информацию, что и при начальной настройке: имя устройства, секретный пароль, пароль enable и пароль виртуального терминала.

В конце диалога вам будет предложено установить SNMP. После того как вы ответите no, диалог расширенного конфигурирования начнет отличаться от базового:

```
Configure SNMP Network Management? [yes]: n
  Configure IP? [yes]:
```

Вместо того чтобы запрашивать имя отдельного интерфейса, который вы хотели бы сконфигурировать, IOS спрашивает о необходимости глобального конфигурирования протоколов. Маршрутизатор сначала определит, какие протоколы маршрутизации и маршрутизируемые протоколы вы хотите использовать. Затем IOS запросит информацию, необходимую для установки каждого протокола на каждом интерфейсе.

Примечание

В главе 5 говорилось о том, что маршрутизируемые протоколы – это протоколы, используемые устройствами для взаимодействия друг с другом, такие как IP или IPX. Эти протоколы называются «маршрутизируемыми», потому что маршрутизаторы их маршрутизируют. Протоколы маршрутизации – это протоколы, которые маршрутизаторы используют для маршрутизации маршрутизируемых протоколов. (Попробуйте произнести в пять раз быстрее, потом медленно прочитайте – в этом есть смысл. Можете обратиться к разделам главы 5 «Маршрутизация в сложных сетях» и «Маршрутизируемые протоколы и протоколы маршрутизации», чтобы освежить в памяти информацию по протоколам.)

Маршрутизируемые протоколы, которые вам разрешено конфигурировать, определяются установленным на маршрутизаторе функциональным пакетом. Например, если у вас установлен функциональный пакет IP (так обычно и бывает), вы сможете маршрутизировать только IP, если же установлен функциональный пакет IP/IPX, то и IP, и IPX.

```
Configure IP? [yes]: y
    Configure IGRP routing? [yes]: n
    Configure RIP routing? [no]: n

Configuring interface parameters:
```

(Обратите внимание на то, что я ответил no на предложение установить два протокола маршрутизации: IGRP и RIP. Так как мы еще не обсуждали функции и назначение этих протоколов, просто пропустим их и установим IP. Поговорим об IP, IGRP и RIP чуть позже.)

Теперь IOS предлагает сконфигурировать все выбранные протоколы для каждого из интерфейсов:

```
Configuring interface parameters:

Do you want to configure Ethernet0  interface? [yes]:
```

Приглашения на ввод информации для конфигурирования интерфейсов повторяют те, что вы видели в процессе базовой установки.

```
Do you want to configure Ethernet0  interface? [yes]: y
  Configure IP on this interface? [yes]: y
    IP address for this interface: 10.162.24.153
    Subnet mask for this interface [255.0.0.0] : 255.0.0.0
    Class A network is 10.0.0.0, 8 subnet bits; mask is /8

Do you want to configure Ethernet1  interface? [yes]:
```

Однако вместо того чтобы завершить диалог после конфигурирования первого интерфейса, IOS в случае расширенной установки предлагает установить любые другие имеющиеся на маршрутизаторе интерфейсы. После того как все интерфейсы сконфигурированы, расширенная установка заканчивается и вы получаете возможность сохранить конфигурацию, аннулировать ее или же приступить к реконфигурированию.

Примечание

Если ваш маршрутизатор подключен к другому устройству через интерфейс, который вы только что установили, после перезагрузки вы можете столкнуться со следующими ошибками:

```
00:38:14: %QUICC_ETHER-1-LOSTCARR: Unit 0, lost carrier. Transceiver
problem?
00:39:14: %QUICC_ETHER-1-LOSTCARR: Unit 0, lost carrier. Transceiver
problem?
00:40:13: %LINK-5-CHANGED: Interface Ethernet1, changed state to
administratively down
00:40:14: %QUICC_ETHER-1-LOSTCARR: Unit 0, lost carrier. Transceiver
problem?
```

Эти ошибки указывают на то, что хотя вы только что установили данный интерфейс, маршрутизатор не может использовать его для обнаружения других устройств. Подключение к интерфейсу концентратора Ethernet должно решить проблему.

Также существует возможность заблокировать интерфейс при помощи команды `shutdown`:

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int e0
Router(config-if)#shutdown
```

(Замените `e0` на тип и номер интерфейса, который вы хотите отключить, введя, например, `e1`, `bri0` или `token1`.)

Вас наверняка интересует, где IOS хранит конфигурации. В следующем разделе будет рассказано о файлах, используемых IOS для хранения конфигурационной информации.

Конфигурационные файлы

Итак, куда же попадает только что созданная вами конфигурация? Cisco IOS хранит конфигурацию в двух разных файлах. Первый из них называется `startup-config`, а второй – `running-config`. Оба файла хранятся на маршрутизаторе, но служат для разных целей.

Файл startup-config

Когда маршрутизатор пошагово проходит процесс загрузки, ему необходим конфигурационный файл, для того чтобы знать параметры устройства. Это файл `startup-config`.

Примечание

Те из вас, кто знаком с ранними операционными системами Microsoft, могут рассматривать файл `startup-config` как аналог `autoexec.bat` для маршрутизатора Cisco.

Если маршрутизатор обнаруживает во флэш-памяти файл с именем `startup-config`, то он приступает к его открытию и чтению. Файл содержит все данные, собранные во время диалога конфигурирования. Когда маршрутизатор пройдет весь `startup-config`, он будет полностью рабочим (по крайней мере настолько, насколько вы определили это при его конфигурировании).

Примечание

Чрезвычайно важным является следующий этап процесса запуска. Если вы поймете, что происходит с файлом `startup-config` после загрузки маршрутизатора, это поможет преодолеть возникающие на вашем пути трудности и избежать путаницы.

После того как маршрутизатор завершил загрузку, он копирует информацию из `startup-config` в файл `running-config`. Далее в процессе своей повседневной работы маршрутизатор использует для получения данных о конфигурации файл `running-config`.

Файл running-config

Файл `running-config` хранится в энергозависимой части памяти маршрутизатора (называемой системной). То есть всякий раз, когда маршрутизатор выключается, файл `running-config` оказывается потерян. После успешной начальной загрузки маршрутизатор копирует текущий `startup-config` в системную память и переименовывает его в `running-config`.

Файл `running-config` хранит все изменения, внесенные в конфигурацию маршрутизатора во время его работы. Например, если вы изменили IP-адрес порта Eth0, такое изменение отразится только в `running-config`. После перезагрузки маршрутизатора изменения не будут запомнены.

Существует способ сохранить изменения, сделанные в конфигурации маршрутизатора. Однако прежде чем говорить о том, как сделать изменения постоянными, необходимо понять, как можно просматривать и редактировать информацию в различных конфигурационных файлах.

Просмотр и редактирование конфигурационных файлов

Содержимое файлов `startup-config` и `running-config` можно просматривать в привилегированном режиме. В памяти маршрутизатора эти файлы хранятся в текстовом виде (что позволяет маршрутизатору читать их как инструкции).

Местоположение файлов `startup-config` и `running-config` можно определить в привилегированном режиме. Чтобы найти файл `running-config`, перейдите к каталогу `system` и выполните команду `dir`.

```
STYSCisco#cd system:
STYSCisco#dir
Directory of system:/

  2  dr-x           0            <no date>  memory
  1  -rw-         541            <no date>  running-config
```

Файл `startup-config` находится во флэш-памяти, к которой обращаются, используя в качестве имени устройства `nvram` (nonvolatile RAM – энергонезависимая память):

```
STYSCisco#cd nvram:
STYSCisco#dir
Directory of nvram:/

  1  -rw-         604            <no date>  startup-config
  2  ----           5            <no date>  private-config
```

(Файл `private-config` содержит информацию о шифровании и является недоступным.)

Чтобы просмотреть `startup-config`, введите команду `show`. (Помните, в главе 4 мы говорили о том, что Cisco IOS распознает частично введенные команды.)

```
STYSCisco#show startup-config
Using 604 out of 7506 bytes
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname STYSCisco
```

```
!
enable secret 5 $1$8HHJ$6dzol4.tGP2gUB6uFfwHSO
enable password enable
!
ip subnet-zero
!
!
!
interface Ethernet0
 ip address 10.16.4.153 255.240.0.0
 no ip directed-broadcast
 no cdp enable
!
interface Ethernet1
 no ip address
 no ip directed-broadcast
 shutdown
 no cdp enable
!
ip classless
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
line con 0
 exec-timeout 0 0
 transport input none
line vty 0 4
 password vterm
 login
!
end
```

Вы наверняка сможете обнаружить соответствие ответов, введенных вами во время процесса установки, с данными, содержащимися в файле `startup-config`. Вся информация диалога конфигурирования сохранена в этом файле.

Просмотр содержимого `running-config` обеспечивается аналогичной командой:

```
Router#show running-config
```

В этот момент содержимое двух файлов является одинаковым: вы только что создали ваш первый `startup-config`, который после загрузки стал файлом `running-config`, так что файлы должны быть идентичны. Но это не значит, что так будет всегда.

Редактировать `startup-config` можно при помощи команды `setup`. Команда `setup` (она будет описана в следующем разделе «Реконфигурирование маршрутизатора») указывает маршрутизатору на то, что он должен начать диалог конфигурирования, даже если файл `startup-config`

уже существует. Диалог конфигурирования, инициированный командой setup, повторяет диалоги базового и расширенного конфигурирования. Это единственный способ прямого редактирования startup-config. После того как startup-config отредактирован, маршрутизатор обязательно попросит вас о перезапуске, чтобы он мог прочитать только что введенные инструкции. (Помните, что маршрутизатор загружает файл startup-config только в процессе своей начальной загрузки.)

Примечание

Можно полностью заменить содержимое startup-config, скопировав поверх него файл running-config. Но такая операция рассматривается не как редактирование, а как замещение.

Большая часть других команд управления функционированием Cisco IOS влияет только на running-config. Пользователи получают возможность изменить параметры конфигурации маршрутизатора, не рискуя при этом потерять сделанные ранее установки. Команды, которые были использованы при установке IP-адреса для интерфейса, пишут только в файл running-config.

Пусть, например, RouterA сконфигурирован так, что для двух Ethernet-портов установлен протокол IP и администратор изменяет адреса этих портов, чтобы протестировать новую схему IP-протокола. После того как проверка завершена, администратор хочет вернуть маршрутизатор обратно в его предыдущее рабочее состояние. Ему нужно или просто перезагрузить маршрутизатор, или же записать startup-config поверх running-config. Тогда конфигурация, хранящаяся в файле running-config, будет перезагружена.

Чтобы скопировать startup-config поверх running-config, тем самым вернув маршрутизатор обратно в его предыдущее состояние без необходимости перезагрузки, выполните следующие шаги:

1. Сначала убедитесь (при помощи команды show) в том, что файл startup-config содержит нужную вам конфигурацию.
2. Затем, переключившись в привилегированный режим, выполните такую команду:

```
STYSCisco#copy starting-config running-config
Destination filename [running-config]?
604 bytes copied in 2.136 secs (302 bytes/sec)
```

3. Текущий startup-config будет скопирован в running-config. Согласившись на имя файла по умолчанию, running-config, вы обеспечите корректное распознавание конфигурации маршрутизатором.

Если же пользователь хочет сохранить новую конфигурацию и не хочет, чтобы она была потеряна при перезагрузке, ему следует выполнить те же операции, только с точностью до наоборот. То есть скопировать running-config поверх startup-config. Тогда текущая конфигура-

ция будет запомнена, и именно ее использует маршрутизатор при следующей загрузке.

```
STYSCisco#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
```

После того как вы научились конфигурировать маршрутизатор с нуля, давайте поговорим о реконфигурировании маршрутизатора, на котором уже заданы установочные параметры.

Реконфигурирование маршрутизатора

Загружая маршрутизатор Cisco, вы можете заметить, что он сразу входит в пользовательский режим, минуя диалог конфигурирования. Это означает, что маршрутизатор обнаружил наличие конфигурации в своей флэш-памяти. Если вы хотите изменить существующую конфигурацию, у вас есть две возможности.

Во-первых, можно удалить текущий `startup-config` и перезагрузить маршрутизатор. Тогда имеющаяся конфигурация будет стерта, и маршрутизатор будет вынужден войти в диалог конфигурирования.

```
STYSCisco#erase startup-config
Erasing the nvram filesystem will remove all files! Continue? [confirm]

Erase of nvram: complete
STYSCisco#
```

Перезапустить маршрутизатор можно, выключив питание или же с помощью команды `reload`.

```
STYSCisco#reload
Proceed with reload? [confirm]

02:33:39: %SYS-5-RELOAD: Reload requested
```

Теперь на маршрутизаторе не установлено никакой конфигурации, и он войдет в диалог предварительного конфигурирования, как если бы он был абсолютно новым.

Есть и второй способ реконфигурирования маршрутизатора – использование команды `setup` в привилегированном режиме. Эта команда получает доступ к диалогу конфигурирования, не разрушая (немедленно) существующий файл `startup-config`.

```
STYSCisco#setup


         --- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: y
At any point you may enter a question mark '?' for help.
```

```
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Basic management setup configures only enough connectivity
for management of the system, extended setup will ask you
to configure each interface on the system

Would you like to enter basic management setup? [yes/no]:
```

С этого момента процесс конфигурирования идентичен другим диалогам настройки. Просто следуйте от вопроса к вопросу, и вскоре ваш маршрутизатор заработает.

Изменение отдельных элементов конфигурации

В первой половине этой главы были представлены различные способы конфигурирования маршрутизатора. Все описанные процессы имеют нечто общее: они собирают минимум информации для большой совокупности объектов.

Но чаще всего возникает необходимость сконфигурировать один-два атрибута для одного-двух интерфейсов, не меняя пароль виртуального терминала. И наоборот, вы можете захотеть изменить секретный пароль, не думая о том, какие протоколы установлены для порта Eth0.

В оставшейся части главы будет показано, как установить все элементы диалога конфигурирования по отдельности. Так вы будете иметь полный контроль над тем, что именно вы конфигурируете.

Установка имени маршрутизатора

Установить (или изменить) имя вашего маршрутизатора можно в привилегированном режиме. Для выполнения большей части команд задания конфигурации требуется войти в режим конфигурирования, который доступен из привилегированного режима.

Как уже говорилось, имя маршрутизатора необходимо для того, чтобы отличать его от других устройств сети. Команда `hostname` позволяет редактировать и изменять имя маршрутизатора. (Помните, что любые изменения конфигурации, сделанные вне диалога конфигурирования, отразятся только в файле `running-config`. И если `running-config` не будет затем скопирован в `startup-config`, то эти изменения не будут сохранены.)

```
Router>enable
Password:
Router#configure
Configuring from terminal, memory, or network [terminal]? terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

Команда config (configure) открывает файл running-config и готовит его к редактированию. Как видно из представленного выше сообщения IOS, с помощью команды config можно редактировать три различных элемента: терминал, память и сеть. Мы сейчас будем работать с терминалами. Когда речь идет о маршрутизаторах Cisco, «терминал» понимается как любой порт или интерфейс, физический или логический.

```
Router(config)#hostname STYSCisco
STYSCisco(config)#
```

Команда hostname, за которой следует новое имя маршрутизатора, установит введенное значение. Обратите внимание на то, что маршрутизатор сразу же обновляет приглашение на ввод команды: теперь в нем присутствует новое имя.

Нажав <Ctrl>+<Z>, вы сохраните running-config и вернетесь обратно в привилегированный режим работы (без конфигурирования).

```
STYSCisco(config)#^Z
STYSCisco#
00:12:39: %SYS-5-CONFIG_I: Configured from console by console
```

Перейдем к установке паролей маршрутизатора.

Пароли маршрутизатора

В диалогах базового и расширенного конфигурирования вы можете задать всего три пароля: секретный пароль enable secret, пароль enable и пароль виртуального терминала. Но вообще-то в Cisco IOS существует пять устанавливаемых паролей. Два оставшихся (дополнительный и консольный) устанавливаются только в командной строке. Давайте поговорим о задании всех пяти паролей IOS.

Секретный пароль

Секретный пароль, управляющий доступом к привилегированному режиму работы командного интерпретатора, можно изменить при помощи команды configure.

```
STYSCisco#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
STYSCisco(config)#enable secret 0 secretpassword
STYSCisco(config)#^Z
STYSCisco#
STYSCisco#configure terminal
```

Выбрав конфигурирование терминала, используйте команду enable, которая будет служить признаком того, что вы хотите изменить один из паролей enable. Командная строка enable secret 0 secretpassword вполне понятна: вы указываете, что хотите заменить секретный пароль enable secret на secretpassword. Предшествующий новому паролю ноль показывает, что вводимый пароль не зашифрован.

Примечание

Используйте 5 (то есть введите строку `enable secret 5 secretpassword`), чтобы показать, что вы хотите задать зашифрованный пароль. Если затем ввести команду `show` для чтения конфигурационного файла, то пароль будет выведен в зашифрованном виде – `$1$SuLj$3frAAOQrjkr3GyBT737lk1`.

Пароль enable

Пароль `enable`, используемый в случае, когда не установлен секретный пароль, задается аналогично:

```
STYSCisco#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

STYSCisco(config)#enable password newpassword
STYSCisco(config)#
```

Отличие между паролями заключается в том, что пароль `enable` не может быть зашифрован. В остальном процессы изменения секретного пароля и пароля `enable` идентичны.

Пароль виртуального терминала

За установку пароля виртуального терминала, защищающего доступ к Telnet-портам, отвечает параметр `line` команды `configure`.

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line vty 0 4
Router(config-line)#login
Router(config-line)#password 0 vterm
Router(config-line)#^Z
Router#
```

Чтобы установить пароль виртуального терминала, необходимо выйти за пределы общего конфигурирования терминала и заняться конфигурированием свойств линии передачи данных.

Примечание

К свойствам линии относятся любые элементы, непосредственно влияющие на работу физических и логических портов маршрутизатора.

Используя команду `line`, мы указываем маршрутизатору, что хотим изменить характеристики виртуального терминала для линий 0-4:

```
Router(config)#line vty 0 4
```

Выполнив эту команду, маршрутизатор входит в режим конфигурирования линий. Теперь можно сообщить, что мы хотим изменить свойства входа в систему и установить пароль для линии 0 – `vterm`:

```
Router(config-line)#login
Router(config-line)#password 0 vterm
```

Как всегда, завершаем команду посредством <Ctrl>+<Z>, чтобы выйти из режима конфигурирования.

Дополнительный пароль

Дополнительный пароль регулирует доступ к дополнительному (auxiliary) порту, если он есть на маршрутизаторе. Дополнительный порт используется для подключения внешнего модема (обеспечивающего наборный доступ к маршрутизатору, но не к сети, к которой подсоединен маршрутизатор). Дополнительный порт есть не у каждого маршрутизатора Cisco, так что и дополнительный пароль можно установить не для каждого.

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line aux 0
Router(config-line)#login
Router(config-line)#password 0 auxpassword
Router(config-line)#^Z
Router#
```

Последний пароль, который можно установить на маршрутизаторе Cisco, – консольный.

Консольный пароль

Консольный пароль контролирует доступ к порту консоли. Так как пароли легко теряются или забываются, лично я редко устанавливаю этот пароль, но выбор, конечно, за вами. Устанавливается консольный пароль следующим образом:

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line con 0
Router(config-line)#login
Router(config-line)#password 0 conpassword
Router(config-line)#^Z
Router#
```

Вы только что установили все пять паролей, предоставленных Cisco IOS. Может быть, вы обратили внимание на то, что нет возможности ввести пустой пароль. Как же тогда удалить пароль, который больше не нужен? И что делать, если вы придумали настолько надежный пароль, что не можете его вспомнить?

Восстановление и удаление паролей

Удалить ненужный пароль достаточно легко. Фактически для удаления почти всех нежелательных свойств маршрутизатора выполняется

одна и та же процедура. В большинстве случаев для удаления свойства или атрибута требуется ввести ту же команду, которая это свойство установила, только предварив ее выражением no.

Например, чтобы удалить установленный ранее консольный пароль, следует выполнить такую команду:

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line con 0
Router(config-line)#login
Router(config-line)# no password 0 conpassword
Router(config-line)#^Z
Router#
```

Посмотрите на no, стоящее перед словом password. Оно указывает IOS, что вы больше не хотите использовать пароль для порта консоли. Но IOS не знает, какой пароль вы хотите исключить, поэтому необходимо включить пароль в выражение no.

Примечание

Если не включить пароль в выражение no, как это сделано в следующем примере, то IOS воспримет его как указание на аннулирование пустого пароля.

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line con 0
Router(config-line)#login
Router(config-line)# no password 0
Router(config-line)#^Z
Router#
```

А так как пустого пароля не существует, команда не будет выполнена.

Примечание

Такой же процесс действует для многих других атрибутов, которые можно установить в рамках Cisco IOS. Например, чтобы удалить IP-адрес порта Ethernet, выполните такую команду:

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface ethernet 0
Router(config-if)#no ip address 10.52.189.122 255.255.0.0
Router(config-if)#^Z
Router#
```

Опять-таки, no помещается перед атрибутом, который требуется удалить.

Иметь возможность удалить ненужные пароли – это замечательно, но что делать, если вы не можете вспомнить пароли, которые хотели бы удалить? Следует выполнить процедуру восстановления паролей.

Когда дело доходит до восстановления паролей, большинство пользователей не замечают очевидного. Все системные пароли (за исключением `enable secret`) хранятся в виде открытого текста как в `running-config`, так и в `startup-config`. Поэтому прежде чем пытаться изменить регистры памяти, просто загляните в `running-config` и `startup-config`.

Если же утерян секретный пароль `enable secret`, необходимо выполнить восстановление пароля.

Примечание

Приведенные ниже шаги могут быть реализованы на большей части малых и средних маршрутизаторов (а также на некоторых маршрутизаторах серий масштаба предприятия), но все же проверьте на веб-сайте Cisco, такова ли процедура восстановления пароля для вашего конкретного маршрутизатора.

Первым шагом является выполнение команды `show version` в пользовательском режиме. В результате будет выведен регистр памяти для конфигурации:

```
Router>show version
Cisco Internetwork Operating System Software
IOS (tm) 1600 Software (C1600-Y-M), Version 12.0(3), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Mon 08-Feb-99 20:15 by phanguye
Image text-base: 0x02005000, data-base: 0x02455958

ROM: System Bootstrap, Version 12.0(3)T, RELEASE SOFTWARE (fc1)
ROM: 1600 Software (C1600-RBOOT-R), Version 12.0(3)T,  RELEASE SOFTWARE (fc1)

Router uptime is 1 hour, 8 minutes
System restarted by power-on
System image file is "flash:aaa0269.bin"

cisco 1605 (68360) processor (revision C) with 7680K/512K bytes of memory.
Processor board ID 17531816, with hardware revision 00000002
Bridging software.
X.25 software, Version 3.0.0.
2 Ethernet/IEEE 802.3 interface(s)
System/IO memory with parity disabled
8192K bytes of DRAM onboard
System running from RAM
8K bytes of non-volatile configuration memory.
4096K bytes of processor board PCMCIA flash (Read/Write)

Configuration register is 0x2102
```

Заметьте, что последняя строка, выведенная командой, содержит регистр конфигурации. Запомните это значение, прежде чем переходить к следующему шагу.

Примечание

Для успешного выполнения следующего шага необходимо наличие программы эмуляции терминала, в которой существует возможность передачи символа прерывания.

Теперь перезагрузим маршрутизатор. Когда он будет загружаться (в течение первых 60 секунд), отправьте ему прерывание. Это остановит процесс загрузки и переведет устройство в режим ROMMON (ROM monitor – ПЗУ-монитор):

```
System Bootstrap, Version 12.0(3)T, RELEASE SOFTWARE (fc1)
Copyright (c) 1999 by cisco Systems, Inc.
C1600 platform with 8192 Kbytes of main memory

monitor: command "boot" aborted due to user interrupt
rommon 1 >
```

Внутри ПЗУ-монитора запустите команду `confreg`. Она проведет вас по диалогу конфигурирования:

```
rommon 4 > confreg

    Configuration Summary
enabled are:
load rom after netboot fails
console baud: 9600
boot: image specified by the boot system commands
      or default to: cisco2-C1600

do you wish to change the configuration? y/n  [n]:
```

Ответьте `yes` на первый вопрос: «Do you wish to change the configuration?» (Хотите ли вы изменить конфигурацию?). Потом отвечайте `no` на все последующие вопросы до тех пор, пока не появится вопрос: «Ignore system config info?» (Игнорировать данные о конфигурации системы?):

```
do you wish to change the configuration? y/n  [n]:  y
enable  "diagnostic mode"? y/n  [n]:  n
enable  "use net in IP bcast address"? y/n  [n]:  n
disable "load rom after netboot fails"? y/n  [n]:  n
enable  "use all zero broadcast"? y/n  [n]:  n
enable  "break/abort has effect"? y/n  [n]:  n
enable  "ignore system config info"? y/n  [n]
```

На этот вопрос ответьте `yes`. Затем отвечайте `no` на следующие вопросы, пока не увидите: «Change boot characteristics?» (Изменить свойства загрузки?). Ответьте `yes` и нажмите <Enter> (по умолчанию используется значение 2).

Маршрутизатор снова спросит вас о том, хотите ли вы изменить конфигурацию. Ответьте `no` и перезагрузите его. Когда маршрутизатор снова включится, он будет работать в режиме конфигурирования. Вы можете или полностью пройти процедуру конфигурирования, следуя подсказкам маршрутизатора, или же ответить `no` и установить секретный пароль `enable secret` вручную.

Освоение конфигурирования маршрутизатора Cisco и управления доступом к нему является важным этапом в изучении продукта. Но на самом деле «сердце» маршрутизатора – это функциональность его интерфейсов. В следующем разделе дается общее представление о возможностях просмотра и конфигурирования интерфейсов.

Новый раздел будет служить введением в главы, составляющие оставшуюся часть книги.

Просмотр и конфигурирование интерфейсов

Интерфейсы маршрутизатора Cisco – это физические порты, через которые могут передаваться данные. Это могут быть Ethernet, Token Ring, ISDN, AUX или любой другой порт. Физические порты являются непосредственными входами в маршрутизатор и выходами из него. Любые данные, перемещаемые устройством, проходят через эти интерфейсы.

Чтобы отслеживать интерфейсы, маршрутизатор нумерует их начиная с 0. Так, Ethernet-порты маршрутизатора Cisco 1605R – это `Ethernet0` и `Ethernet1`. Нумерация важна в тех случаях, когда вы хотите просмотреть свойства конкретного интерфейса. П142

Чтобы просмотреть текущий статус интерфейса `Ethernet0`, используйте такую команду:

```
Router#show interface ethernet 0
Ethernet0 is administratively down, line protocol is down
  Hardware is QUICC Ethernet, address is 00d0.58a8.e150 (bia 00d0.58a8.e150)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 170/255, load 1/255

Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 01:39:31, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     48 packets output, 2880 bytes, 0 underruns
     48 output errors, 0 collisions, 0 interface resets
```

```
     0 babbles, 0 late collision, 0 deferred
     48 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Router#
```

Примечание

Команда `show interface` без параметров выводит статус всех интерфейсов. В зависимости от модели на одном маршрутизаторе может быть 100 и более интерфейсов.

Первая выведенная строка: `Ethernet0 is administratively down, line protocol is down`, наиболее интересна для нас сейчас. Все остальные строки являются продуктом протоколов, связанных с интерфейсом.

Первая половина рассматриваемой строки сообщает, что `Ethernet0 is administratively down` (порт Ethernet0 административно заблокирован). Это означает, что интерфейс не был активирован (в IOS) для использования маршрутизатором. Интерфейс маршрутизатора Cisco становится полностью функциональным только тогда, когда он административно включен, а также включен протокол линии передачи данных. Давайте сделаем этот интерфейс рабочим.

Чтобы административно разблокировать интерфейс, требуется изменить его свойство `shutdown`. Используем команду `configure`:

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface ethernet 0
Router(config-if)#no shutdown
Router(config-if)#
01:58:38: %LINK-3-UPDOWN: Interface Ethernet0, changed state to up
```

Команда `configure` вводит нас в режим глобального конфигурирования, где мы сообщаем IOS, что хотим отредактировать интерфейс `Ethernet0`. По приглашению на ввод команды `Router(configure-if)#` видно, что редактируются интерфейсы маршрутизатора.

Для изменения административного состояния интерфейса (блокирования или включения) применяется команда `shutdown`. Для включения интерфейса перед командой помещается ключевое слово `no`. Это указывает IOS, что блокировать интерфейс `Ethernet0` не требуется.

Теперь интерфейс административно включен, но протокол линии передачи данных все еще выключен, что видно из приведенного ниже фрагмента вывода команды `sh int e 0`:

```
Router#show interface ethernet 0
Ethernet0 is administratively up, line protocol is down
  Hardware is QUICC Ethernet, address is 00d0.58a8.e150 (bia 00d0.58a8.e150)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 170/255, load 1/255
```

Пока протокол линии передачи данных не работает, интерфейс нельзя считать полностью функциональным. (На самом деле он не может сде-

лать почти ничего.) Поэтому нашей следующей задачей будет включение этого протокола. Для начала необходимо связать протокол с интерфейсом. В данном примере мы будем устанавливать протокол IP.

Первым шагом к подключению интерфейса IP является установка адреса.

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface ethernet 0
Router(config-if)#ip address 10.52.189.122 255.255.0.0
Router(config-if)#^Z
Router#
02:16:09: %SYS-5-CONFIG_I: Configured from console by console
Router#
```

Параметры конфигурирования IP-адреса – это адрес и маска подсети. Команда `ip address` сообщает IOS, что вы хотите установить указанный адрес с указанной маской подсети для текущего интерфейса.

```
ip address 10.52.189.122 255.255.0.0
```

Но просто установив адрес, вы не измените статус интерфейса на включенный. Необходимо выполнить второй шаг: предоставить что-то, с чем протокол сможет поддерживать связь. Маршрутизатор будет считать протокол выключенным до тех пор, пока он не определит, что интерфейс можно использовать. Достаточно подключить к порту интерфейса концентратор, и протокол передачи данных будет включен.

После подключения маршрутизатора к концентратору через конфигурируемый интерфейс команда `sh int` выводит новый статус для `Ethernet0`.

```
Router#show interface ethernet 0
Ethernet0 is administratively up, line protocol is up
  Hardware is QUICC Ethernet, address is 00d0.58a8.e150 (bia 00d0.58a8.e150)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 170/255, load 1/255
```

Чтобы сконфигурировать любой другой интерфейс, повторите те же операции, заменив `int e0` на название, соответствующее выбранному интерфейсу.

Вы успешно установили свой первый интерфейс для маршрутизатора Cisco, давайте теперь познакомимся еще с одним вопросом.

Создание баннера

Одна из самых забавных вещей, которые можно делать с маршрутизатором Cisco, – это конфигурировать различные баннеры, которые фигурируют в заголовках командных строк. Для каждой линии передачи данных, используемой для доступа к маршрутизатору, можно задать баннерное сообщение.

Например, чтобы сконфигурировать баннер для входа в систему через консоль, выполните следующие действия:

```
Router#configure terminal
Router(config)#banner motd ! This is the message of the day !
Router(config)#
```

Можно заменить восклицательные знаки любым символом, тогда этот символ станет символом-ограничителем. То есть маршрутизатор будет знать, что ваше сообщение закончено, когда он встретит символ-ограничитель. В табл. 6.4 в общих чертах представлены различные баннеры, доступные в Cisco IOS, и объяснено, где они появляются.

Таблица 6.4. Различные баннерные сообщения Cisco

Баннер	Местоположение
MOTD (message of the day)	Такой баннер выводится перед всеми остальными баннерами для каждой линии. То есть перед приглашением на вход в систему пользователь всегда будет видеть баннер MOTD.
Exec	Баннер Exec выводится (после баннера MOTD) для каждого, кто входит на маршрутизатор через виртуальный терминал.
Incoming	Баннер Incoming выводится только для пользователей, которые пользуются обратной сессией Telnet для обращения к маршрутизатору.
Login	Баннер Login выводится после баннера MOTD, чтобы подать сигнал о приглашении на вход в систему.

Резюме

- Загрузка маршрутизатора, на который еще никогда не устанавливалась конфигурация, инициирует диалог конфигурирования Cisco IOS.
- Cisco IOS в диалоге конфигурирования запрашивает у пользователя всю основную информацию, необходимую для создания функциональной конфигурации маршрутизатора.
- Диалог конфигурирования предназначен только для сбора того минимального объема информации, которого достаточно для того, чтобы маршрутизатор заработал. После завершения диалога пользователю еще предстоит «вылизывать» конфигурацию.
- Если на маршрутизаторе уже установлена какая-то конфигурация, можно войти в диалог конфигурирования, используя команду `setup` в привилегированном режиме.

Вопросы и ответы

Вопрос Почему пользователь маршрутизатора обязательно должен уточнить конфигурацию после завершения диалога конфигурирования?

Ответ Во время диалога собирается только базовая информация, относящаяся к маршрутизатору и его окружению. То есть предлагается ввести только те данные, которые необходимы для того, чтобы маршрутизатор смог начать работать в сети (то есть чтобы у него был хотя бы один рабочий интерфейс с адресом). Такие элементы, как протоколы маршрутизации и WIC, в диалоге не затрагиваются.

Тест

Вопросы

1. Как называется уникальный кабель Ethernet, используемый для консольных соединений?
2. Какие команды используются для входа в режим конфигурирования линии передачи данных?
3. Баннер какого типа выводится при входе в систему для любого типа линий?
4. Какая команда используется для инициирования диалога конфигурирования?

Ответы

1. Консольный rollover-кабель.
2. `line` (в режиме глобального конфигурирования).
3. MOTD (Message of The Day).
4. `setup` (в привилегированном режиме).

Упражнения

1. Пользователь устанавливает баннер MOTD следующим образом:

```
Router#configure terminal
Router(config)#banner motd B This is Sophia's router! So Back Off B
Router(config)#
```

Но когда сообщение выводится, оно выглядит по-другому:

```
This is Sophia's router! So
```

Почему?

Ответ

Буква B была выбрана в качестве символа-ограничителя. Поэтому маршрутизатор прекратил вывод сообщения, встретив первый эк-

экземпляр этого символа, которым в данном случае стала буква B в слове «Back».

2. Каким будет IP-адрес интерфейса Ethernet1 после перезагрузки маршрутизатора, если два конфигурационных файла выглядят следующим образом:

```
running-config:
Current configuration:
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
enable secret 5 $1$Zk7H$Svxp5px.vyLcRg7hDXo8Z1
enable password enable
!
ip subnet-zero
no ip routing
!
!
!
interface Ethernet0
 ip address 10.52.189.122 255.255.0.0
 no ip directed-broadcast
 no ip route-cache
 no shutdown
 no cdp enable
!
interface Ethernet1
 no ip address 198.56.34.1 255.255.0.0
 no ip directed-broadcast
 no ip route-cache
 no shutdown
 no cdp enable
!
ip classless
!
!
line con 0
 exec-timeout 0 0
 transport input none
line vty 0 4
 password vterm
 login
!
end
startup-config:
```

```
Current configuration:
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
enable secret 5 $1$Zk7H$Svxp5px.vyLcRg7hDXo8Z1
enable password enable
!
ip subnet-zero
no ip routing
!
!
!
interface Ethernet0
 ip address 10.52.189.122 255.255.0.0
 no ip directed-broadcast
 no ip route-cache
 no shutdown
 no cdp enable
!
interface Ethernet1
 no ip address 10.75.24.125 255.255.0.0
 no ip directed-broadcast
 no ip route-cache
 no shutdown
 no cdp enable
!
ip classless
!
!
line con 0
 exec-timeout 0 0
 transport input none
line vty 0 4
 password vterm
 login
!
end
```

Ответ

10.75.24.125

3. Как установить пароль для Telnet-порта в `goaway`?

Решение

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#line vty 0 4
Router(config-line)#login
Router(config-line)#password 0 goaway
```

или

```
Router#setup  (и установить пароль виртуального терминала в «goaway»)
```

7

Резервное копирование маршрутизаторов Cisco

Эта глава посвящена вопросам защиты маршрутизатора и его конфигураций. Умение настраивать маршрутизаторы мало чего стоит, если вы потеряете все настройки. Другими словами, созданная вами конфигурация хороша лишь настолько, насколько давно вы ее сохраняли (или, в некоторых случаях, насколько хороша ее резервная копия). Для пользователей Cisco важно понимать, где и когда необходимо резервирование, и уметь делать это эффективно.

Изучая вопросы защиты маршрутизаторов Cisco, рассмотрим следующие темы:

- Физическое резервирование маршрутизаторов
- Резервные копии конфигурации Cisco IOS

Первая тема касается решений по защите оборудования, принимаемых во время проектирования сети. Такие методы, как использование нескольких источников электропитания и маршрутизаторов с возможностью горячей замены, составляют основу физического резервирования.

Поскольку в конфигурации Cisco немного параметров, связанных с аппаратным резервированием, несколько первых разделов этой главы посвящены в основном теоретическим вопросам. Однако, чтобы стать хорошим специалистом, вы должны не только разбираться в теории, но и на практике уметь создавать эффективные схемы резервирования.

Во второй теме рассматривается резервное копирование файлов данных маршрутизатора. Ранее мы уже говорили о том, что набор параметров, определяющих функционирование устройства, сосредоточен в двух файлах: `running-config` и `startup-config`. Теперь мы обсудим спосо-

бы их копирования и безопасного хранения на случай аварии маршрутизатора.

Эта глава не только научит вас защищать свои инвестиции (как материальные, так и интеллектуальные), но и познакомит вас с функциями, лежащими в основе работы маршрутизаторов Cisco. Понимание этих функций необходимо при изучении технологии маршрутизации.

Физическое резервирование

Физическое резервирование применяется для защиты маршрутизаторов на физическом уровне. То есть этот вид резервирования используется на случай таких событий, как потеря питания или отказ интерфейса. Эти неисправности, способные вывести сеть из строя, случаются достаточно часто.

Такой способ защиты сети зачастую либо оказывается слишком дорогим, либо назначение сети не оправдывает связанных с ним дополнительных затрат на резервирование оборудования.

Если сеть, которую вы проектируете (или в которой работаете), предназначена для решения ответственных задач, вам следует обратить особое внимание на методы физического резервирования маршрутизаторов. Многие предприятия требуют, чтобы их маршрутизирующее оборудование было зарезервировано и обеспечивало минимально возможное время простоя. Такие сети, состоящие из нескольких подсетей с несколькими маршрутизаторами, обычно связывают несколько географически удаленных пунктов. Эти сети полностью полагаются на работоспособность маршрутизаторов, и их простой может быть приравнен к потере прибыли.

Примечание

Решение о том, что является критичным для организации, зависит от вида бизнеса и методов управления. Отключение некоторых небольших удаленных подразделений может и не повлиять на доходы основного бизнеса. Такие сети не считаются критически важными.

В этой главе будут рассмотрены несколько видов резервирования оборудования маршрутизации. Основным (и наиболее просто реализуемым) является резервирование питания, для которого чаще всего применяются два способа:

- Источники бесперебойного питания (ИБП)
- Дополнительные блоки питания (оборудование Cisco)

Установка ИБП защищает маршрутизатор от перебоев в питании. Источник бесперебойного питания обычно помещается между маршрутизатором и розеткой электропитания. Такое подключение защищает маршрутизатор двумя путями.

Питающее напряжение подается на маршрутизатор через ИБП, что позволяет последнему подавлять скачки напряжения, которые могут повредить маршрутизирующее оборудование. Скачки напряжения (особенно во время грозы) могут представлять серьезную угрозу для большинства устройств.

Примечание

Не все ИБП обеспечивают защиту от скачков напряжения. Ознакомьтесь с документацией производителя, прежде чем полагаться на такую защиту.

Кроме того, ИБП, как правило, способны обнаружить отсутствие напряжения в силовой сети здания. В этом случае ИБП перейдет на питание от внутренней батареи и продолжит поставлять питающее напряжение маршрутизатору. Важно помнить, что сеть не сможет функционировать в полном объеме, если одни только маршрутизаторы защищены по питанию. Эффективный план защиты на случай аварии должен предусматривать защиту всех компьютеров и других устройств.

Примечание

Батарея ИБП не предназначена для питания устройств в процессе повседневной работы. Емкости большинства ИБП хватает на поддержание 20-минутной работы маршрутизатора. Большинство производителей считают это время достаточным для того, чтобы корректно завершить работу любого оборудования.

Если использование ИБП не обеспечивает достаточной защиты сети, есть еще одна возможность. Некоторые устройства Cisco могут получать питание от дополнительного источника. То есть сам маршрутизатор может содержать два блока питания, и если один из них выйдет из строя, оставшийся продолжит работу, а неисправный может быть заменен.

Еще одним способом защиты является автономное (offline) резервирование. В этом случае необходим дополнительный, аналогичный работающему в сети маршрутизатор, на котором хранится полная копия файлов конфигурации. При возникновении серьезных проблем с основным маршрутизатором вместо него устанавливается запасной.

Автономный резервный маршрутизатор – хорошее решение для достаточно больших компаний со сложной средой маршрутизации, но стандартизированным оборудованием для маршрутизации (другими словами, в которых преобладают маршрутизаторы Cisco одинаковых серий, а лучше и одинаковых моделей).

Последний способ физического резервирования, который мы рассмотрим, – это оперативное резервирование, называемое также системой с избыточностью. Компании могут создать двойные каналы передачи данных между сетями, так чтобы каждый канал обслуживался отдельным маршрутизатором. Тогда если один маршрутизатор выходит из строя, второй продолжает обслуживать маршрут.

Резервные источники питания

Одной из наиболее часто встречающихся проблем, влияющих на работоспособность маршрутизатора, является отключение электропитания. К сожалению, такие отключения очень сложно, если вообще возможно, предугадать. Они часто случаются без предупреждений. Хотя ИБП и не производятся специально для Cisco, они представляют собой хорошее средство защиты от такого рода катастрофы.

Примечание

ИБП применяются достаточно широко и напрямую не связаны с темой нашей книги. Но они обеспечивают плавный переход к разговору о вторичных источниках питания.

Источник бесперебойного питания расположен между маршрутизатором и энергосистемой общего пользования. ИБП (также называемый резервным аккумулятором) имеет некоторую фиксированную емкость. В случае отключения электричества аккумулятор продолжает питать маршрутизатор электроэнергией.

Однако необходимо помнить, что обычно ИБП не предназначен для того, чтобы обеспечивать работу маршрутизатора в течение долгого времени. Назначение ИБП в том, чтобы предоставить вам достаточно времени для корректного завершения работы устройств, снижая тем самым риск утери и порчи данных.

ИБП просто включается в стандартную сетевую розетку вашей компании. Затем в ИБП можно включить маршрутизатор, гарантируя таким образом поддержание некоторой работоспособности устройства на случай отключения питания. Если же для удовлетворения ваших потребностей в резервировании недостаточно просто возможности корректного выключения оборудования, то вам может понадобиться генератор.

Предприятия с большими потребностями в маршрутизации обычно используют генераторы для снабжения своих маршрутизаторов неограниченным (потенциально) количеством электроэнергии. Многие генераторы выдают постоянный ток. В таких системах может возникнуть необходимость внести некоторые изменения в маршрутизаторы Cisco.

Почти все модели маршрутизаторов Cisco масштаба предприятия допускают использование блока питания постоянного тока (глава 2 «Введение в аппаратную часть Cisco»). Такие источники, получая питание от генератора (или большого резервного аккумулятора), соединены непосредственно с маршрутизатором. Процесс подключения источника постоянного тока к маршрутизатору Cisco не очень сложен, однако сам блок питания обычно рассматривается как дополнительное оборудование. Источники постоянного тока Cisco DC пригодны не для всех маршрутизаторов; большинство маршрутизаторов Cisco не способны работать с таким оборудованием.

Примечание

Источники постоянного тока Cisco доступны только для маршрутизаторов уровня предприятия (в том числе для класса ISP), но не многие из них поставляются с такими источниками в составе стандартного оборудования. Такие маршрутизаторы предназначены для гораздо более крупных сред маршрутизации, чем рассматриваемые в этой книге.

Достаточно часто используется такая форма резервирования питания, как использование избыточных источников питания Cisco. Многие маршрутизаторы Cisco имеют до четырех отсеков (расположенных на задней стенке корпуса), в которые может быть установлено несколько блоков питания. Большая часть компаний пользуется возможностью установить в этих открытых отсеках второй и третий блоки питания.

Использование нескольких блоков питания Cisco

Наличие нескольких блоков питания может защитить маршрутизаторы от двух основных проблем. Первая – это неисправность блока питания, которая может вывести маршрутизатор из строя на несколько часов, до тех пор, пока не будет найдена подходящая замена.

Вторая проблема заключается в скачках напряжения и грозовых помехах. Хотя несколько блоков питания и не могут защитить вас так, как специальные сетевые фильтры, но все же появляется больше шансов пережить такой перепад напряжения, если у вас есть резервный источник питания.

Так как блоки питания постоянно преобразуют энергию, они выделяют значительное количество тепла, что может привести к ускоренному старению внутренних компонентов блока питания, делая его более подверженным внезапным отказам, чем другие части оборудования.

Несколько блоков питания (два или более) могут защитить вас от такой неисправности. Маршрутизаторы Cisco используют преимущество наличия нескольких блоков питания, автоматически переходя на использование другого, если один из блоков перестает работать. Для этого оба блока должны быть подключены к сети (и включены).

Однако если ваша сеть очень важна, и ваши маршрутизаторы вполне оправдывают некоторые дополнительные меры безопасности, проанализируйте возможность использования автономных и оперативно доступных резервных маршрутизаторов.

Резервные маршрутизаторы

Резервные маршрутизаторы делятся на две категории: оперативно доступные и автономные. Автономные маршрутизаторы, как правило, используются в сетях, где инженеры пришли к соглашению об использовании стандартизованной платформы. Они обычно остаются вне сети до тех пор, пока не понадобятся, чтобы заместить работающие маршрутизаторы. А оперативно доступные маршрутизаторы всегда на месте и имеют свои функции в сети. Когда один маршрутизатор выходит из строя, оперативно доступный резервный маршрутизатор берет на себя его маршруты.

Методы автономного и оперативного резервирования требуют дополнительного конфигурирования Cisco IOS (поговорим об этом в этой же

главе, но чуть позже). Конфигурирование включает в себя процессы переноса конфигурационных файлов с одного маршрутизатора Cisco на другой.

При выборе стратегии резервирования маршрутизаторов наиболее полезными оказываются два инструмента: карандаш и бумага. Составьте схему как можно большей части сети и опробуйте на бумаге различные варианты, прежде чем принимать окончательное решение.

Автономные резервные маршрутизаторы

Если в вашей сети используется несколько маршрутизаторов Cisco одинаковых серий, вы можете подумать об использовании в качестве средства резервирования автономного запасного маршрутизатора. Автономные резервные маршрутизаторы используются в тех средах, где необходимо минимизировать время простоя маршрутизатора, а оперативно доступные запасные маршрутизаторы слишком дороги.

Причина использования автономных резервных маршрутизаторов проста: вы получаете маршрутизатор, который без труда может заменить вышедшее из строя устройство особой важности и взять на себя его функции. Для применения автономного запасного маршрутизатора необходимо провести дополнительное планирование на этапе проектирования системы, но эти затраченные усилия оправдывают себя в случае повреждения маршрутизатора.

Давайте посмотрим, как именно автономный резервный маршрутизатор может использоваться в среде маршрутизации. На рис. 7.1 изображена маршрутизируемая сеть, в которой основные действующие маршрутизаторы стандартизованы.

Пусть (к примеру) все маршрутизаторы на рис. 7.1 относятся к серии Cisco 2600 и имеют по два Ethernet-порта. Маршрутизаторы связаны друг с другом и некоторые из них – с центральным концентратором/коммутатором.

Примечание

Для упрощения примера можно убрать с рисунка некоторые элементы оборудования. Для обсуждаемого вопроса это не имеет значения.

Если возникает проблема с маршрутизатором D, то уничтожается маршрут между двумя основными сегментами среды. Сети A, B и C будут изолированы от сетей D, E и F. Замена этого маршрутизатора является критически важной.

Если бы в этой сети присутствовал автономный резервный маршрутизатор, его без труда можно было бы поместить вместо маршрутизатора D. Но теперь вы, наверное, уже догадываетесь, что размещение маршрутизатора в нужном месте еще не означает, что он заработает.

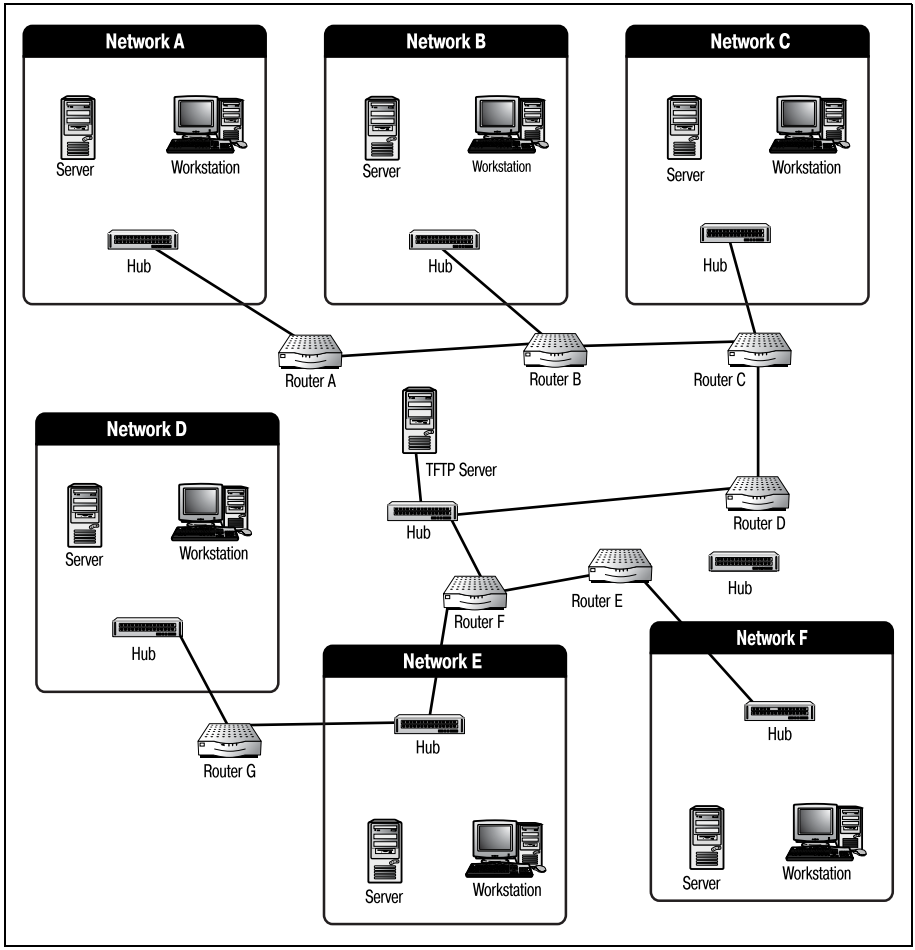

Рис. 7.1. *Стандартизованная среда маршрутизации*

Если таблицы маршрутов автономные резервные маршрутизаторы могут получить от других устройств сети, то конфигурирование интерфейсов должно быть проведено вручную. Лучший способ обеспечить соответствие конфигураций резервного и вышедшего из строя маршрутизаторов – хранение резервных копий конфигурационных файлов.

- Убедитесь, что на маршрутизатор подано питание (30 секунд).
- При первых признаках проблем со связью проверьте физические кабели интерфейсов маршрутизатора (1 минута).
- Используйте команду `show interface` для просмотра статусов интерфейсов отказавшего маршрутизатора. (Так мы пытаемся подтвердить, что неисправные интерфейсы находятся в состоянии «protocol down», а не «administratively down».) Вывод `show interface` должен выглядеть так:

```
Ethernet0 is administratively up, line protocol is down
  Hardware is QUICC Ethernet, address is 00d0.58a8.e150 (bia
00d0.58a8.e150)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 252/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 1w0d, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     3 packets output, 180 bytes, 0 underruns
     3 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     3 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Ethernet1 is administratively up, line protocol is down
  Hardware is QUICC Ethernet, address is 00d0.58a8.e151 (bia
00d0.58a8.e151)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 252/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 1w0d, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     3 packets output, 180 bytes, 0 underruns
     3 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     3 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Оба интерфейса имеют статус «protocol down», но административно не заблокированы, что указывает на дополнительные проблемы. К этому моменту маршрутизатор проработал приблизительно две минуты.

Предположим, что вы обнаружили неисправность в течение первых пяти минут после ее возникновения, затем потратили три минуты на то, чтобы поставить на место поврежденного маршрутизатора

новый, значит, всего простой маршрутизатора длится уже около 10 минут.

- Принято решение заменить маршрутизатор автономным резервным маршрутизатором.
- Неисправный маршрутизатор удаляется из сети (2 минуты).
- Запасной автономный маршрутизатор помещается на место неисправного маршрутизатора (2 минуты).
- Запускается диалог базового конфигурирования для разрешения работы одного интерфейса Ethernet (3 минуты).
- С TFTP-сервера копируются конфигурационные файлы неисправного маршрутизатора (2 минуты).
- Маршрутизатор перезагружается, и начинается процесс обновления сетевых данных, чтобы продублировать таблицы маршрутов соседних маршрутизаторов.

Весь процесс подключения и конфигурирования резервного автономного маршрутизатора должен занять около 20 минут, в течение которых сеть будет простаивать. Для некоторых компаний это недопустимо. В таких сетях решением проблемы могут стать оперативно доступные маршрутизаторы.

Хотя, учитывая тему книги, мы концентрируем ваше внимание именно на применении автономных резервных маршрутизаторов, но все же вы должны знать, что существуют и другие способы защиты оборудования и среды маршрутизации. Поэтому ниже мы предлагаем вам краткое описание оперативно доступных резервных маршрутизаторов.

Оперативно доступные резервные маршрутизаторы

Использование оперативно доступных резервных маршрутизаторов требует значительных дополнительных усилий при проектировании и конфигурировании, но это того стоит. Во многих случаях при выходе маршрутизатора из строя создание избыточных путей между сегментами может сэкономить время простоя и время, необходимое для конфигурирования запасного маршрутизатора, что обещает душевное спокойствие людям, работающим в компаниях, где потерянное время означает потерянную прибыль.

В большинстве компаний избыточные маршруты создаются не для каждой линии сети. Выбираются самые важные линии, и обслуживающие их маршрутизаторы клонируются.

Заняв свое место, запасной маршрутизатор может делать два дела. Во-первых, он может просто «сидеть и ждать», пока что-то случится. Как только основной маршрутизатор выходит из строя, запасной (оперативно доступный резервный) принимает на себя его функции и начинает управлять путями неисправного маршрутизатора. Когда маршрутизатор используется в таком качестве, он не выполняет никаких сетевых функций до тех пор, пока основной маршрутизатор не выйдет из

строя. Второе, чем может заниматься запасной маршрутизатор в ожидании неисправности основного, – это перераспределение нагрузки.

Оперативно доступный резервный маршрутизатор может работать как устройство перераспределения нагрузки. То есть он может поделить обязанности по маршрутизации с основным маршрутизатором, при этом каждый из них будет отвечать за определенную часть передающегося по маршруту трафика. Если один маршрутизатор выходит из строя, то другой просто берет на себя оставшуюся часть трафика.

Давайте сначала посмотрим, как проектируются избыточные маршруты. На рис. 7.2 представлена сеть из предыдущего примера, в которой реализован избыточный путь.

В этом сценарии маршрутизаторы 1 и 2 служат запасными оперативно доступными маршрутизаторами для D и E. Как видите, концентрато-

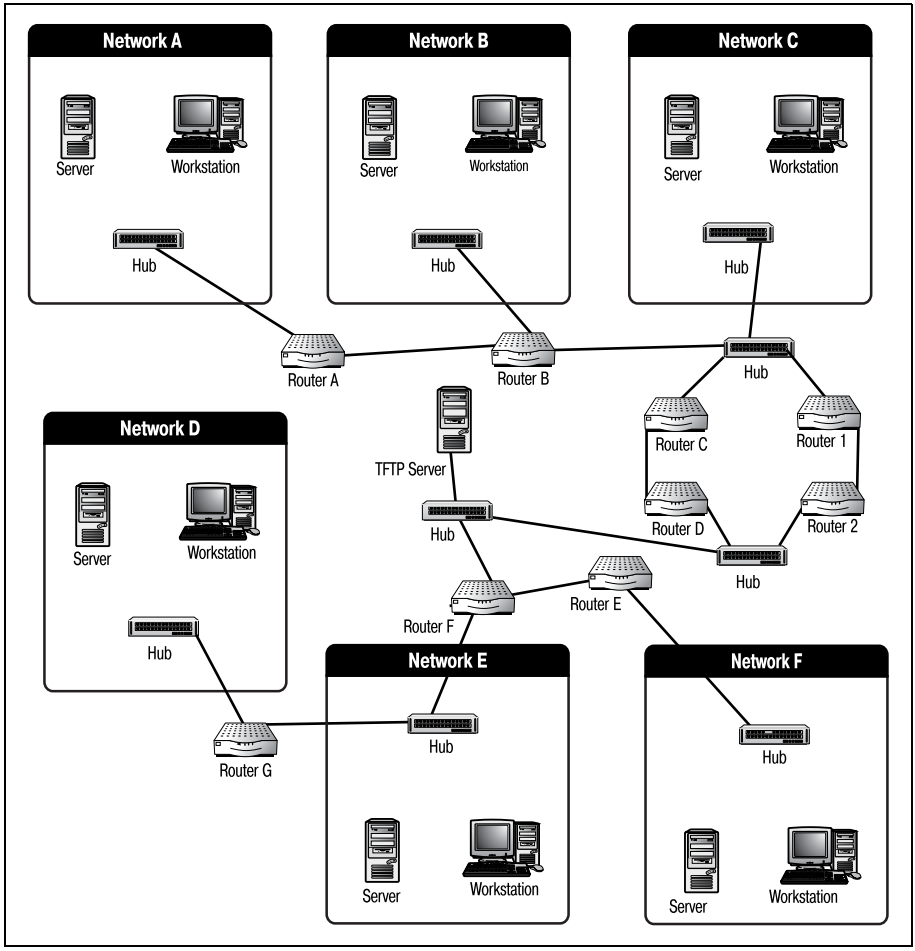

Рис. 7.2. *Избыточный путь*

ры/коммутаторы питают оба маршрутизатора. При такой схеме действий, как на рис. 7.2, для снабжения двух маршрутизаторов данными настойчиво рекомендуется использовать коммутаторы (switches) вместо концентраторов (habs).

Примечание

Концентратор (hab), как правило, пересылает информацию всем портам одновременно. Поэтому нельзя гарантировать, что одному из маршрутизаторов будет оказано предпочтение.

Используя коммутаторы (особенно коммутаторы Cisco), вы можете установить маршрут к основному маршрутизатору. То есть на коммутаторе Cisco вы можете создать статический путь, который будет поставлять информацию только маршрутизатору D. Если маршрутизатор D оказывается неисправным, вы просто дезактивируете этот статический путь и активируете тот, который передает данные маршрутизатору 1.

Хотя такое решение может быть эффективным, но имеет ли оно смысл? Если резервные маршрутизаторы уже присутствуют в сети, почему бы не использовать их?

Самой популярной формой избыточности маршрутизаторов является перераспределение нагрузки. Для реализации перераспределения нагрузки требуется достаточно сложное конфигурирование, которое может быть выполнено только с помощью специального оборудования Cisco. На рис. 7.3 изображена та же сеть, с которой мы уже работали, но теперь в ней реализовано перераспределение нагрузки.

Обратите внимание на то, что на схеме появились две новые линии: одна между маршрутизаторами D и 1, а вторая между маршрутизаторами E и 2. Эти каналы помогают маршрутизаторам общаться друг с другом и договариваться о перераспределении нагрузки.

В коммутаторах, обслуживающих эти маршрутизаторы, также необходимо сделать ряд конфигурационных изменений. Вместо одного статического пути, поставляющего данные одному маршрутизатору, следует установить два пути с одинаковыми метриками. Одинаковые метрики обеспечивают отсутствие предпочтений в пользу одного или другого пути со стороны коммутатора.

Примечание

Хотя это и выходит за пределы темы обсуждения нашей книги, все же отметим, что для защиты от петель маршрутизации может возникнуть необходимость реализации на коммутаторе STP (Spanning Tree Protocol – протокол остовного дерева).

При использовании перераспределения нагрузки на двух избыточных путях каждому маршрутизатору приписывается определенный процент трафика. Например, можно настроить маршрутизатор D так, чтобы он обрабатывал 40% трафика канала, в то время как маршрутиза-

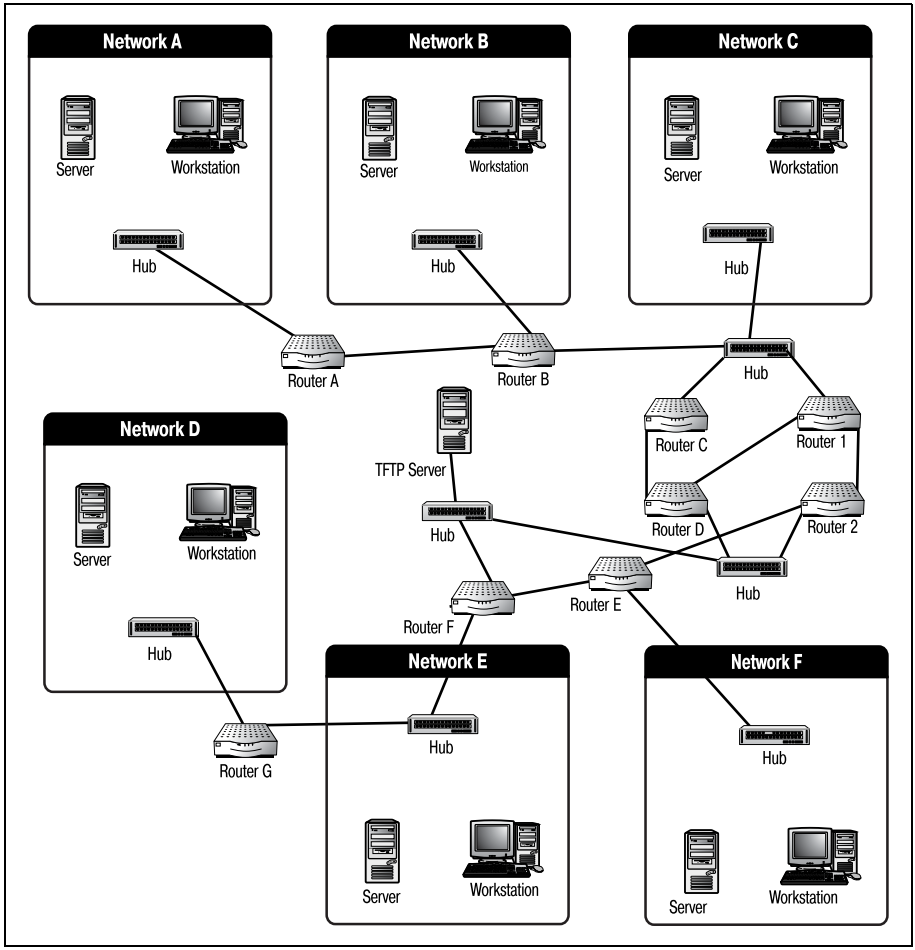

Рис. 7.3. *Перераспределение нагрузки*

тор 1 передает оставшиеся 60%. Если какой-то из маршрутизаторов оказывается поврежденным, второй автоматически понимает это и принимает на себя весь трафик.

Такие сложные (как представленная выше) конфигурации могут быть выполнены только для маршрутизаторов Cisco старших моделей, например серии 7000. Эти маршрутизаторы созданы для сложных сред маршрутизации, в которых между двумя любыми точками сети может существовать несколько избыточных путей.

Эта часть главы должна была, если и не описать подробно, то обрисовать вам различные возможности резервирования оборудования, доступные пользователям Cisco. Но наиболее важным в резервировании все же является процесс сохранения конфигураций маршрутизатора.

Резервирование конфигурационных файлов Cisco IOS

Оставшаяся часть главы посвящена процессам резервирования конфигурационных файлов вашего маршрутизатора Cisco. Хранение текущих резервных копий полной конфигурации маршрутизатора очень важно вне зависимости от того, большая ваша сеть или маленькая, критически важная или нет.

В технических кругах известен такой афоризм: «Вы хороши лишь настолько, насколько хороша ваша последняя резервная копия». Это чистая правда по отношению к тому, что касается маршрутизаторов. Так как большая часть памяти маршрутизатора энергозависима, одно неожиданное отключение может оказаться опустошительным. Поэтому хорошим правилом является сохранение актуальных, постоянно обновляемых резервных копий ваших конфигураций.

Что и зачем следует резервировать

Выбрать, что резервировать на маршрутизаторе, достаточно просто: все, что можно. Помните, что многие установки маршрутизатора хранятся в энергозависимой памяти, поэтому необходимо иметь достоверные резервные копии этих файлов на случай аварии. Первый файл, который следует резервировать, – это `running-config`. Данный файл является самым важным, так как он содержит параметры настройки для текущей работы маршрутизатора. Если маршрутизатор вышел из строя, а текущей копии `running-config` не существует, то потребуется восстанавливать установки с нуля. Файл `running-config` хранится только в ОЗУ маршрутизатора, поэтому он теряется при каждом выключении маршрутизатора или при потере питания.

Второй файл, который следует резервировать, – это `startup-config`. В главе 6 говорилось, что `startup-config` и `running-config` должны быть идентичны. К сожалению, так бывает не всегда. Даже лучшие специалисты проводят испытания с настройками `running-config` и забывают скопировать их в `startup-config`. Поэтому лучше всего хранить текущие копии обоих файлов.

Примечание

Почти во всех случаях ваши файлы `running-config` и `startup-config` идентичны. Но бывает так, что изменение конфигурации, сделанное в `running-config`, не скопировано в `startup-config` (или же в `startup-config` скопировано неправильное изменение). В любом случае сохранение двух файлов – это просто дополнительная мера предосторожности.

Последний файл вашего маршрутизатора Cisco, который следует включить в план резервирования, – образ Cisco IOS. Хотя порча IOS – это наименее вероятная причина сбоя в работе маршрутизатора, но все

может быть. Наличие копии образа IOS также полезно при покупке запасного автономного маршрутизатора (на котором может стоять более новая или более старая версия). Приобретя запасной маршрутизатор, вы можете установить на него имеющуюся версию IOS, чтобы гарантировать совместимость.

Приоритетом для инженера Cisco, занимающегося резервным копированием файлов, должно стать создание копий файлов `startup-config` и `running-config`.

running-config и startup-config

Лучшим способом сохранения конфигурации маршрутизатора является копирование конфигурационных файлов на TFTP-сервер. TFTP-сервер очень удобен для хранения данных внешнего маршрутизатора: он может хранить копии всех ваших образов IOS и копии файлов `startup-config` и `running-config`. Хотя такое решение не слишком надежно, но все же это неплохой способ гарантировать наличие резервной копии конфигурации маршрутизатора на случай выхода его из строя.

Примечание

Помните, что, как и любая другая, резервная копия, хранящаяся на TFTP-сервере, хороша лишь настолько, насколько недавно она сделана. Поэтому если вы сохранили свои настройки на сервере, затем что-то изменили, а потом восстановили конфигурацию с сервера, то все ваши изменения потерялись.

Резервное копирование без использования TFTP-сервера

Если у вас нет доступа к сервисам TFTP или их использование невозможно в вашей среде маршрутизации, вам следует принять некоторые базовые меры предосторожности, чтобы обеспечить максимально возможную безопасность ваших настроек. Хотя лучшая защита достигается при хранении конфигурационных файлов вне маршрутизатора, но уже просто поддерживая `startup-config` в текущем состоянии, вы добьетесь многого.

Использование команды `copy` должно войти у вас в привычку вне зависимости от конкретного плана резервного копирования. Эта команда особенно важна, если вы не используете TFTP-сервер.

Примечание

Копирование `running-config` в `startup-config` будет поддерживать ваш `startup-config` в наиболее актуальном состоянии. Помните, что маршрутизатор загружает конфигурацию из файла `startup-config` только при начальной загрузке. Любые изменения, которые не перенесены из `running-config` в `startup-config`, не будут запомнены.

Чтобы скопировать ваш файл running-config в startup-config, используйте команду copy в привилегированном режиме:

```
Router#copy run start
```

Одна эта команда скопирует все настройки, с которыми маршрутизатор работает в настоящий момент, в файл startup-config. Если теперь произойдет сбой в работе маршрутизатора, то конфигурация, сохраненная в startup-config, будет автоматически скопирована IOS обратно в running-config.

При загрузке маршрутизатора копирование startup-config в running-config происходит без вмешательства пользователя. А вот копировать running-config в startup-config, чтобы актуальная конфигурация была доступна при загрузке устройства, должны именно вы как пользователь маршрутизатора.

Резервное копирование текущего образа IOS без использования TFTP-сервера несколько проще. Простейший способ получения «хорошей» копии образа IOS – это загрузка его с веб-сайта Cisco CCO (Cisco Connection Online; см. главу 3 «Введение в Cisco IOS»). К сожалению, большая часть маршрутизаторов Cisco поставляется без копии образа IOS на отдельном носителе (маршрутизаторы поставляются с предустановленной IOS). Поэтому если у вас нет TFTP-сервера, на который можно было бы скопировать образ, вам потребуется получить образ от Cisco.

Примечание

Есть две возможности получения образа IOS от Cisco. Первая заключается в скачивании образа с веб-сайта Cisco CCO (www.cisco.com). Для входа на этот сайт вы должны быть зарегистрированным пользователем и обладать паролем, который также можно получить от Cisco.

Есть и другая возможность – купить у дилера IOS на носителе.

Если у вас есть версия Cisco IOS, хранящаяся вне маршрутизатора, то перенести образ на маршрутизатор можно посредством специальных программ. Наибольшего доверия заслуживает Cisco Remote Software Loader. Cisco RSL – это Windows-программа, которая автоматически обновляет IOS на маршрутизаторе без участия TFTP-сервера. RSL имеет в своем составе агент TFTP, но действовать как автономный TFTP-сервер она не может.

Хотя такой план действий может быть и не лучшим для резервирования конфигурации вашего маршрутизатора Cisco, но если в вашем распоряжении нет других средств, то можно воспользоваться и этими.

Использование TFTP-сервера

Чтобы создать резервную копию любого конфигурационного файла маршрутизатора, используйте команду copy. Эта команда позволяет перемещать файлы маршрутизатора в другие его области или же на внешние носители. Если маршрутизатор (или его настройки) повреж-

ден, то снова используйте команду copy, чтобы скопировать файлы с TFTP-сервера обратно на маршрутизатор.

Примечание

Команда copy доступна только в привилегированном режиме работы командного интерпретатора Cisco.

Примечание

Сервер TFTP (Trivial File Transfer Protocol – простой протокол передачи файлов) отправляет файлы устройствам, используя упрощенную версию FTP, которая работает по протоколу UDP (User Datagram Protocol – протокол передачи дейтаграмм пользователя). TFTP несовместим с полнофункциональным FTP.

В главе 3 «Введение в Cisco IOS» рассказывалось о копировании образа Cisco IOS с TFTP-сервера на маршрутизатор. Процессы, изучаемые в данной главе, не сильно отличаются от рассмотренных в главе 3. Команда copy используется и здесь, и там, но сейчас мы поговорим о дополнительных функциях, которые предусматривают перемещение отдельных конфигурационных файлов с маршрутизатора и на него.

Сначала необходимо определить, резервные копии каких файлов вы хотели бы хранить. Как правило, предполагается хранение только одного из файлов startup-config и running-config, так как если вы поддерживаете маршрутизатор должным образом, то они должны быть идентичны.

Чтобы скопировать файлы на TFTP-сервер, выполните следующие шаги:

```
Router#copy running-config tftp
Address or name of remote host []? 10.4.16.152
Destination filename [running-config]?routera-running-config
```

Обратите внимание на то, что структура команды аналогична использованной при копировании файла running-config в startup-config. После того как в качестве места назначения копии указан TFTP-сервер, IOS запрашивает у вас IP-адрес сервера. Введите адрес и имя файла назначения, тогда IOS создаст копию вашей конфигурации на TFTP-сервере.

Если маршрутизатор имеет доступ к TFTP-серверу, то одна такая команда сохранит для вас текущую копию конфигурации маршрутизатора.

Примечание

Если вы работаете с несколькими маршрутизаторами, позаботьтесь о том, чтобы изменить имя копии (файла назначения) так, чтобы оно отражало название маршрутизатора, которому принадлежит конфигурация. TFTP-серверы копируют только в каталог root (и только из него). Поэтому, если вы скопируете настройки файла running-config для каждого маршрутизатора на один TFTP-сервер, не изменяя имя файла, то при каждом копировании файл будет перезаписываться.

Если маршрутизатор вашей сети поврежден и вы используете резервный автономный маршрутизатор, вам необходимо скопировать конфигурацию неисправного маршрутизатора на резервный. Чтобы сделать это, сначала необходимо сконфигурировать интерфейс для запасного маршрутизатора. Для того чтобы быстро настроить один интерфейс, обычно обращаются к диалогу базового конфигурирования (он изучался в главе 6).

После того как интерфейс сконфигурирован, можно скопировать конфигурационный файл с TFTP-сервера на автономный маршрутизатор:

```
Router#copy tftp startup-config
Address or name of remote host []? 10.4.16.152
Source filename [startup-config]?routera-running-config
…

Router#reload

…
```

Заметьте, что вместо того, чтобы сохранить резервную копию `running-config` в файле `running-config`, мы посылаем ее в `startup-config`. Это гарантирует, что маршрутизатор не потеряет свои настройки, если с ним что-нибудь случится. Но так как `startup-config` читается только в процессе начальной загрузки маршрутизатора, необходимо выполнить команду `reload`, которая перезагрузит устройство. Теперь ваш запасной маршрутизатор находится в рабочем состоянии.

К настоящему моменту мы уже изучили большую часть основных вопросов, необходимых для понимания маршрутизации Cisco. Пришло время перейти к обсуждению более сложных тем. Впереди глава 8 «Введение в маршрутизируемые протоколы».

Резюме

- Одним из наиболее важных занятий пользователя маршрутизатора Cisco является его профилактическое обслуживание. Необходимо постоянно осуществлять резервирование вашего маршрутизатора Cisco.
- Аппаратную часть маршрутизатора Cisco можно защитить, используя источники бесперебойного питания, автономные или работающие в сети резервные устройства или комбинацию вышеперечисленного.
- Программное обеспечение защищается посредством выполнения резервного копирования на TFTP-серверы.

Вопросы и ответы

Вопрос Зачем нужно хранить резервные копии как файла `running-config`, так и `startup-config`, если на момент начальной загрузки они идентичны?

Ответ Может быть сделано множество изменений, которые пропадут при аварийном отключении маршрутизатора, если проводивший изменения специалист забудет скопировать `running-config` в `startup-config`.

Тест

Вопросы

1. Какая команда используется для перемещения файлов с места на место?
2. Что отличает источники питания маршрутизаторов Cisco масштаба предприятия от источников питания маршрутизаторов других серий?
3. Правда ли, что файл `running-config` копируется поверх `startup-config` при каждом выключении маршрутизатора?
4. От чего может защитить ИБП?

Ответы

1. `copy`.
2. Возможность питания от источника постоянного тока.
3. Неправда. Файл `startup-config` копируется в `running-config` в процессе загрузки маршрутизатора.
4. ИБП может защитить от отключения электричества. Некоторые модели ИБП могут также защитить ваше оборудование от резких скачков напряжения.

Упражнение

1. Используя команду `copy`, отправьте копию конфигурации из IOS на TFTP-сервер и обратно, чтобы имитировать сбой в работе маршрутизатора и его замену резервным устройством.

Решение

```
Router#copy runnning-config tftp
Address or name of remote host []? 10.4.16.152
Destination filename [running-config]?routera-running-config

Router#copy tftp startup-config
Address or name of remote host []? 10.4.16.152
Source filename [startup-config]?routera-running-config
…

Router#reload

…
```

II

Протоколы и их функционирование

8

Введение в маршрутизируемые протоколы

В этой главе мы немного отвлечемся от конфигурирования маршрутизаторов Cisco. Для того чтобы успешно перемещать данные из одного места в другое, всем маршрутизаторам необходимо одно: протокол. Поэтому на страницах этой главы будет рассказано о работе протоколов и о том, как они связаны с маршрутизаторами. Оставшаяся часть книги (в той или иной форме) знакомит с конфигурированием работы маршрутизатора в отношении используемых протоколов. Чтобы приступить к конфигурированию, необходимо иметь четкое представление о том, как на самом деле работают протоколы.

В главе будут обсуждаться следующие вопросы:

- Распределение протоколов по категориям
- Модель OSI
- Типы и классы протоколов
- Инкапсуляция протоколов

Категории протоколов

Протоколы можно разделить на две категории: маршрутизируемые протоколы и протоколы маршрутизации. Маршрутизируемые протоколы – это то, что большинство людей понимает под словом «протокол». Маршрутизируемые протоколы могут маршрутизироваться другими устройствами. Такие общеизвестные протоколы, как IP и IPX, относятся к маршрутизируемым.

Протоколы маршрутизации – это протоколы, которые маршрутизаторы используют для взаимодействия друг с другом (когда маршрутизируют маршрутизируемые протоколы). К протоколам маршрутизации относятся RIP и OSPF. Эти протоколы помогают маршрутизаторам наиболее эффективно передавать данные.

Примечание

Полное понимание протоколов необходимо для того, чтобы разобраться в маршрутизации Cisco, поэтому в книге будет уделено внимание многим протоколам. Прежде чем перелистнуть последнюю страницу, вы познакомитесь с такими маршрутизируемыми протоколами, как:

- IP (Internet Protocol – интернет-протокол)
- IPX (Internetwork Packet Exchange – межсетевой пакетный обмен)

А также со следующими протоколами маршрутизации:

- RIP (Routing Information Protocol – протокол маршрутной информации)
- IGRP (Interior Gateway Routing Protocol – протокол маршрутизации внутреннего шлюза)
- EIGRP (Enhanced Interior Gateway Routing Protocol – усовершенствованный протокол маршрутизации внутреннего шлюза)
- OSPF (Open Shortest Path First – первоочередное открытие кратчайших маршрутов)
- BGP (Border Gateway Protocol – протокол граничного шлюза)

В данной главе мы будем изучать работу протоколов, использование их маршрутизаторами, а также поговорим о том, в каком случае и какому протоколу отдать предпочтение. Наиболее подробно будут рассмотрены протоколы IP и IPX. Именно эти два протокола вы, скорее всего, встретите при работе с маршрутизаторами Cisco.

Помните (особенно это касается новичков в маршрутизации), что не все протоколы можно маршрутизировать. То есть даже если протокол может применяться для передачи информации в сети, это не означает, что маршрутизаторы могут взаимодействовать с ним. Поэтому при планировании маршрутизируемой среды можно принять решение избегать использования некоторых протоколов. В конце главы будет представлено краткое описание немаршрутизируемых протоколов с пояснениями.

Чтобы иметь четкое представление о том, как работают протоколы, сначала необходимо понять, где они работают (имеется в виду не физическое местоположение, а способ взаимодействия протокола с компьютером или маршрутизатором). Разные протоколы работают на разных уровнях модели OSI; одно устройство может использовать набор протоколов для выполнения какой-либо задачи.

Примечание

Многие протоколы сгруппированы в «наборы», или «стеки». Дело в том, что большая часть протоколов была создана для выполнения каких-то специальных функ-

ций, и один протокол умеет делать то, что другой не может. Объединившись, протоколы дополняют друг друга и могут коллективно выполнять ряд задач.

Примером двух протоколов, работающих вместе, служат TCP и IP. Эти протоколы работают на разных уровнях модели OSI, поддерживая связь между компьютерами. В следующих разделах будет описана роль модели OSI и ее влияние на работу маршрутизаторов Cisco.

Примечание

Знание функций всех семи уровней модели OSI (даже тех, которые не имеют прямого отношения к маршрутизации) поможет вам лучше осознать перемещение данных. Вам также легче будет выявлять возможные проблемы. Например, зная, что World Wide Web работает на седьмом уровне, вы сможете определить источник связанной с Web ошибки. Так как седьмой уровень OSI управляет использованием приложений, то проблема с World Wide Web может быть связана с программой просмотра страниц.

Важно понимать, что протоколы служат определенной цели, и совсем не обязательно, что это маршрутизация. Одной из многих функций маршрутизируемых протоколов является сегментирование и инкапсулирование данных. Маршрутизаторы пользуются преимуществом этой функции, применяя ее для содействия процессу маршрутизации. Но маршрутизируемые протоколы могут существовать и в средах, где нет маршрутизаторов. (Поэтому бывают и немаршрутизируемые протоколы.)

Чтобы научиться безошибочно определять, какие протоколы будут работать с маршрутизаторами Cisco, а какие – нет, необходимо сначала понять различия между ними. Проиллюстрировать эти различия поможет модель OSI.

Модель OSI

В начале 1980-х годов Международная организация по стандартизации (International Organization for Standardization, ISO) разработала модель OSI (Open Systems Interconnect, взаимодействие открытых систем). В основе OSI лежит получившая дополнительное развитие эталонная модель DoD. ISO взяла четырехуровневую модель DoD и структурировала ее еще сильнее. В результате появились семь уровней OSI. Подразделение модели OSI на семь уровней помогает выделить некоторые области передачи данных для специализированных протоколов.

Уровни модели OSI пронумерованы снизу вверх. Чтобы перенести информацию с одного компьютера на другой, данные должны пройти через эти уровни как на передающем, так и на принимающем устройствах. Информация седьмого уровня одной машины должна:

1. Переместиться с седьмого на первый уровень машины-отправителя.

2. Быть передана протоколу (возможно, для маршрутизации) и доставлена на устройство назначения.
3. Перейти с первого на седьмой уровень машины-получателя.

Каждый уровень модели OSI выполняет свою определенную функцию. Информация, поступающая с одного уровня, слегка изменяется, чтобы стать удобочитаемой для следующего уровня. Когда данные достигают принимающего устройства, они переходят от уровня к уровню в обратном порядке, чтобы аннулировать изменения, сделанные исходным компьютером. Поэтому даже если маршрутизатор имеет дело с данными только после того, как они попали на третий уровень, но при этом уровни с 4 по 7 работают некорректно, то эти уровни будут пагубно воздействовать на функционирование маршрутизатора.

В следующих разделах для обеспечения полного понимания роли протоколов в маршрутизации рассмотрены все уровни и выполняемые ими задачи. Изучив модель OSI, мы сможем перейти к обсуждению принципов IP и IPX.

Уровень приложений

Уровень приложений (уровень 7) занимается координированием связи между приложениями. Этот уровень модели OSI синхронизирует данные, перемещающиеся между клиентами и серверами, заведуя передачей файлов, сетевым управлением и обработкой услуг. В круг обязанностей уровня приложений входят:

- World Wide Web (WWW)
- Шлюзы электронной почты
- Электронный обмен данными (Electronic Data Interchange, EDI)
- Чаты
- Утилиты навигации по Интернету

Уровень приложений можно рассматривать как первый шаг, предпринимаемый данными, покидающими ПК, для их последующей маршрутизации. Уровень приложений – это прямой доступ маршрутизируемой информации к программе, в которую она направляется. То есть большая часть маршрутизируемой информации (если не вся) или исходит от приложения, работающего на устройстве, или же для него предназначена. Уровень приложений модели OSI задает правила, управляющие обработкой такой информации.

Например, когда вы просматриваете веб-страницу, вы просматриваете данные на уровне приложений. Информация, образующая веб-страницу, проходит через семь уровней модели OSI на удаленном сервере, «пересекает» пространство Интернета и поднимается по уровням OSI вашего компьютера. В конечном счете данные становятся удобочитаемыми на уровне приложений.

Уровень представлений

Функция уровня представлений состоит в преобразовании информации уровня приложений в формат, который понимают другие уровни. Любые шифрования, дешифрования и сжатия данных производятся на шестом уровне модели OSI. Уровень представлений также отвечает за все функции аудио- и видеопредставлений. Доступные на уровне представлений службы включают:

- MP3
- RealAudio
- RealVideo
- JPEG
- GIF

Заметьте, что все эти сервисы требуют сжатия. Для MP3 необходимо чрезвычайно эффективное сжатие звука, а GIF использует сжатие изображений. Если не применять к данным сжатие, их объем будет слишком велик, чтобы маршрутизатор смог обработать поступившие данные без ошибок. Чем больше информации маршрутизируется в рамках одного сеанса связи, тем меньше вероятность того, что она достигнет нужного получателя в целости и сохранности. Поэтому сжатие данных (вместе с шифрованием и дешифрованием) играет важную роль в обеспечении маршрутизации.

Сеансовый уровень

Сеансовый уровень (уровень 5) координирует связь между сетевыми устройствами. Сеансовый уровень (работая в паре с сеансовым уровнем другого устройства) устанавливает сеанс связи между двумя приложениями. Два сеансовых уровня контролируют «разговор» и в нужный момент заканчивают сеанс. Также сеансовый уровень отвечает за:

- SQL (Structured Query Language, структурированный язык запросов)
- X Windows
- NFS (Network File System, сетевая файловая система)

Информация, отправляемая (или получаемая) этими тремя верхними уровнями (уровни 5, 6 и 7), называется *пользовательскими данными*. Пользовательские данные преобразовываются в другие формы данных, подходящие для обработки остальными уровнями.

Транспортный уровень

Транспортный уровень (уровень 4) отвечает за получение пользовательских данных от верхних уровней и разбиение их (или же обратную сборку) на удобные для передачи порции. Порции данных, фор-

мируемые транспортным уровнем, называются *сегментами*. Такие сегменты передаются нижним уровням для дальнейшей обработки.

Примечание

Работаете ли вы с сегментами, кадрами, дейтаграммами или ячейками – знание терминологии уровней модели OSI очень вам поможет. Для каждого уровня существует специальный формат данных, так что уровень (а обычно и протокол) можно идентифицировать по формату данных. Например, так как TCP работает на четвертом уровне, то все передаваемые по нему данные будут иметь форму «сегментов».

Транспортный уровень также предоставляет возможность управления потоками данных. Управление потоками помогает этому уровню обеспечить надежную передачу данных от одного устройства к другому (с установлением соединений). Транспортный уровень забирает пользовательские данные с верхних уровней и сегментирует их. Затем эти сегменты по одному передаются указанному получателю. Получив сегмент, получатель отправляет обратно уведомление. Если отправитель не получает уведомления, он заново передает сегмент. После нескольких попыток отправляющее устройство старается заново установить соединение с получателем. Если оказывается, что получатель не отвечает (принимающее устройство больше не работает в сети), то генерируется ошибка и остальные сегменты не отправляются. Поток данных, передаваемых с одного ПК на другой по TCP, изображен на рис. 8.1.

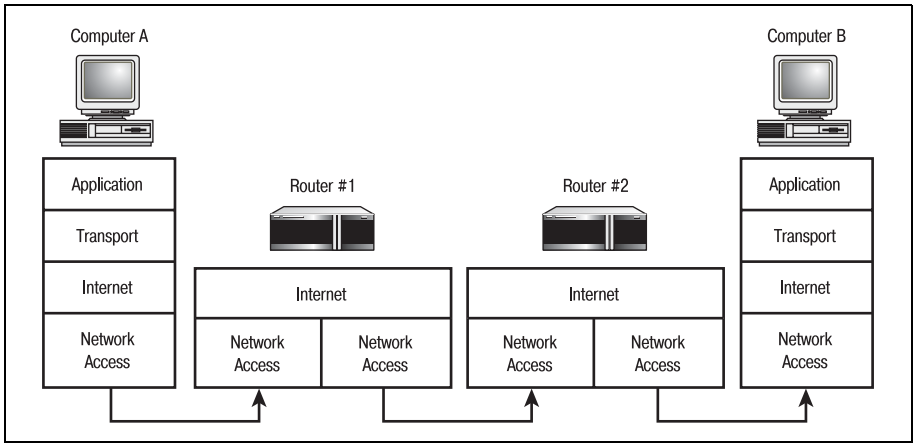

Рис. 8.1. *Поток данных TCP*

Сетевой уровень

Все уровни выполняют очень важные функции, но основная часть процесса маршрутизации проходит именно на уровне 3. Большая часть устройств межсетевого взаимодействия (маршрутизаторы, коммутаторы третьего уровня и мосты) работает на сетевом уровне модели OSI.

Для обеспечения правильности маршрутизации сетевой уровень создает логическую карту сети. Эта карта работает как проводник при направлении данных из одной части сети в другую. Хотя функция составления карты сети имеет очень важное значение для маршрутизации, используется она не только маршрутизаторами. Персональные компьютеры и другие устройства также используют службы сетевого уровня для определения местоположения маршрутизаторов в окружении. Это позволяет им посылать маршрутизаторам информацию, которая должна быть доставлена по сети.

Первым этапом создания карты сети является преобразование сегментов транспортного уровня в пакеты. Затем эти пакеты, к которым добавлена адресная информация, передаются канальному уровню.

Примечание

Так как маршрутизаторы практически всегда работают только на третьем уровне модели OSI, они перемещают данные только в виде пакетов. Однако когда мы дойдем до WAN-технологий, вы увидите, что бывают маршрутизаторы, которые передают кадры и ячейки. Помните, что формат данных во многом зависит от протокола.

Когда устройство получает пакет, из него извлекается информация об отправителе и помещается в таблицу. По мере роста таблицы сетевой уровень получает более явную картину сетевого окружения. Другие протоколы и устройства могут использовать эту информацию для эффективной маршрутизации данных.

Примечание

Данные, используемые сетевым уровнем, также сохраняются локально на маршрутизаторах Cisco в таблице маршрутов. Конкретная информация, содержащаяся в таблице маршрутов, зависит от используемого протокола маршрутизации. Более подробно таблицы маршрутов будут затронуты в следующих главах, посвященных протоколам маршрутизации.

Например, если устройство сети A хочет отправить данные устройству сети B, оно посылает широковещательный пакет через свою локальную сеть (A). Этот широковещательный пакет проводит «разведку», разыскивая адрес получателя. Так как принимающее устройство не находится в сети A, то «разведчик» не получит ответ. Тогда устройство сети A предположит, что получатель находится в сети B, и отправит данные туда.

Прежде чем достичь сети B, пакет попадает на маршрутизатор 1. Маршрутизатор просматривает свою собственную таблицу маршрутов (составленную из данных сетевых уровней) и определяет, что получатель действительно находится в сети B. Тогда маршрутизатор 1 пересылает пакет в соответствующие сети. Маршрутизатор, соединяющий две сети, отметит эту ситуацию в своей таблице маршрутов и в даль-

нейшем будет направлять все пакеты, предназначающиеся данному устройству, в сеть B.

Ключевым элементом сценария является знание адреса устройства, с которым вы пытаетесь поделиться информацией. Сбором и отслеживанием этих адресов занимается канальный уровень.

Канальный уровень

Сетевой уровень поддерживает карту сети, а канальный уровень (уровень 2) обеспечивает корректность информации в этой карте за счет адресации. Канальный уровень принимает пакеты от сетевого уровня и преобразует их в кадры данных. Кадры содержат следующую информацию:

- Преамбулу (указывающую начало кадра)
- Адрес назначения (получателя)
- Адрес отправителя
- Поле длины (в стандартном кадре Ethernet) указывает размер данных, содержащихся в кадре
- Поле типа (в кадрах Ethernet_II) указывает, какой протокол будет получать данные
- Данные
- Контрольная последовательность кадра (проверочный номер, соответствующий контрольной сумме кадра)

Канальный уровень также обеспечивает много других возможностей, которые доступны не всем устройствам. Поэтому уровень был разбит на два подуровня: MAC и LLC. В каждом подуровне существуют свои правила и атрибуты.

Подуровень MAC

Подуровень MAC (media access control, управление доступом к среде) отвечает за кадрирование пакетов сетевого уровня. Разбивая пакеты на кадры, подуровень MAC прикрепляет к пакету адресную информацию, в которую входит MAC-адрес.

Примечание

Каждое устройство, пригодное для работы в сети, имеет присвоенный ему при изготовлении адрес, однозначно идентифицирующий этот компонент в сети. Такой адрес называется MAC-адресом.

Еще одной функцией подуровня MAC является обслуживание верхних уровней без установления соединения. Такое обслуживание имеет место, когда данные посылаются на устройство без предварительного установления с ним соединения. Другими словами, отправляющее устройство отсылает данные по сети, заранее не извещая об этом получателя.

Таблица 8.1. «За» и «против» обслуживания без установления соединения

За	Против
Быстрее, чем обслуживание с установлением соединений. (Не тратится время на установление соединений и ожидание ответов)	Не гарантируется доставка кадров
Меньше нагрузка на сеть	Получатель не знает об отправке данных. Машина, которой вы отправляете информацию, может даже не работать в этот момент

Далее вы узнаете, что подуровень MAC имеет очень важное значение для маршрутизации. Так как MAC-адреса уникальны и распознаются почти всеми протоколами, то их можно встретить во многих аспектах маршрутизации.

Подуровень LLC

Одной из задач подуровня LLC (logical link control, управление логическим соединением) является предоставление обслуживания с установлением соединений (в то время как MAC не устанавливает соединение). В этом случае перед отправкой кадров устанавливается соединение с получателем, благодаря чему доставка кадров гарантируется получением уведомлений.

Физический уровень

Первый уровень модели OSI определяет физическое соединение между устройствами. Физический уровень принимает кадры от верхних уровней и передает их в виде битов по среде сети передачи данных. При работе с интерфейсами маршрутизаторов Cisco это становится гораздо более понятным. Все физические порты и интерфейсы оборудования Cisco для маршрутизации работают на физическом уровне модели OSI. Весь вышеописанный процесс представлен на рис. 8.2.

Осмысление модели OSI поможет вам понять, как работают протоколы, что, в свою очередь, поможет овладеть принципами маршрутизации, на постижение которых некоторые тратят долгие годы.

Закончив разговор о внутреннем устройстве модели OSI, обсудим, как маршрутизаторы используют эту информацию для перемещения данных из системы в систему.

Типы и классы протоколов

Примечание

Книга посвящена маршрутизации и маршрутизаторам Cisco, поэтому я считаю, что в подробном рассмотрении немаршрутизируемых протоколов нет необходимости.

Просто знайте, что существование протокола не означает, что он маршрутизируем. Примером немаршрутизируемого протокола служит NetBEUI. С помощью NetBEUI могут быть созданы целые сети, которые будут замечательно работать до тех пор, пока не будут использоваться маршрутизаторы.

Для того чтобы понять, как протоколы работают и как они перемещают данные, необходимо знать, что у них «внутри». Для каких целей протокол был создан, на каких уровнях модели OSI он работает и что от него можно ожидать? Рассмотрение различных типов протоколов поможет вам понять, как работают протоколы маршрутизатора и как маршрутизаторы Cisco с ними управляются.

Протоколы с установлением соединения и протоколы без установления соединения

Протоколы с установлением соединения разработаны так, чтобы способствовать установлению соединений между системами. То есть когда две системы используют протоколы с установлением соединений, то прежде чем передавать информацию, они устанавливают между собой соединение.

Протоколы с установлением соединения, такие как TCP (Transmission Control Protocol – протокол управления передачей), работают на транс-

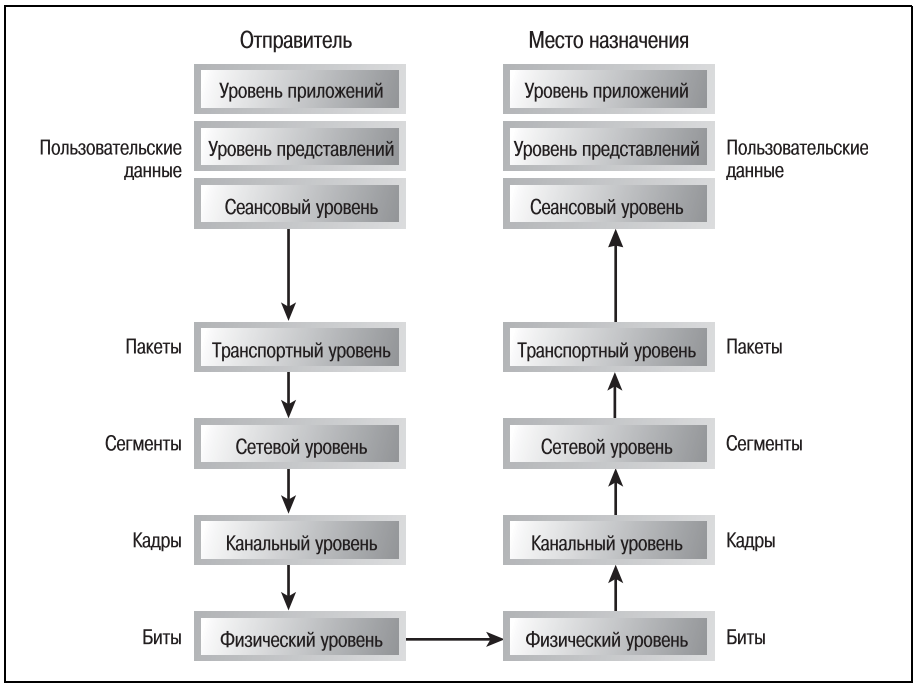

Рис. 8.2. *Схема прохождения данных через семиуровневую модель OSI*

портном уровне модели OSI. Так как многие маршрутизаторы работают на сетевом уровне (которому внутренне присуще отсутствие соединений), протоколы с установлением соединения не являются маршрутизируемыми. Но мы все же поговорим о них, так как у них имеется ряд полезных свойств, к тому же, они тесно связаны со многими дружественными маршрутизатору протоколами без установления соединений, с которыми мы их и сравним.

Когда одна система хочет отправить информацию в соседнюю, получателю отправляется запрос на установление соединения. Система-отправитель ждет уведомление о готовности к соединению. Получив его, система-отправитель устанавливает соединение, и две системы могут свободно обмениваться данными. По завершении соединения инициировавшая его система отправляет пакет «tear-down», который показывает, что соединение закончено и используемые для его поддержания ресурсы должны быть освобождены. Процесс установления соединения представлен на рис. 8.3.

Сам процесс (инициирование соединений перед обменом информацией) не создает особых проблем; в действительности это один из наиболее привлекательных аспектов данной категории протоколов. Проблемы для некоторых сетей может вызвать тот способ, которым соединение поддерживается.

Когда инициатор запроса посылает пакет установления соединения, к пакету добавляется запись о пути, который он проходит. Таким образом пакет создает маршрут, по которому будет происходить обмен информацией между двумя устройствами. Получающая пакет система отправляет свое уведомление (acknowledgment – ACK) по тому же пути, который прошел пакет установления соединения. И после открытия сеанса связи между двумя системами все пересылаемые ими пакеты следуют по тому же самому маршруту, в результате чего между устройствами возникает гарантированный канал связи.

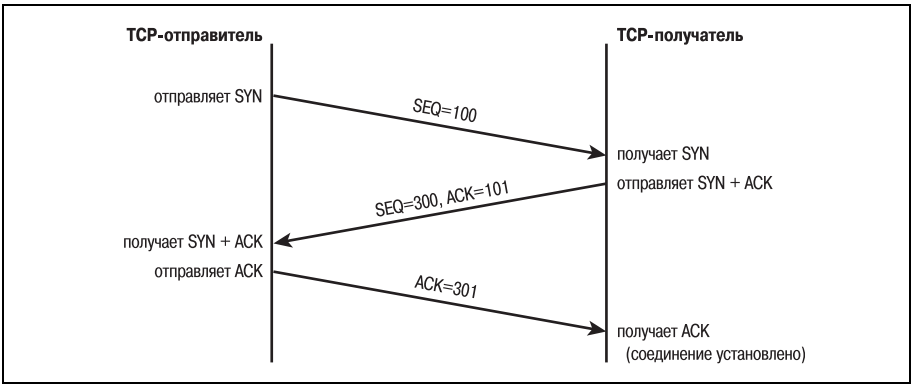

Рис. 8.3. *Процесс установления соединения*

Но установление соединения и поддержание назначенного пути требуют значительных ресурсов. Накладные расходы протоколов с установлением соединения выше, чем у протоколов без установления соединений. К тому же, из-за того что в основе работы маршрутизатора лежит случайность, он не может поддерживать назначенные пути.

Примечание

Маршрутизаторы не могут гарантировать, что два пакета будут отправлены по одному и тому же пути. Маршрутизатор всегда пытается выбрать «лучший путь» для отправки информации. А так как на этот выбор влияет множество связанных с состоянием сети обстоятельств, то от пакета к пакету маршрут может меняться.

Протоколы с установлением соединения имеют и ряд преимуществ для сетевого окружения. Так как прием любого пакета, отправленного той или иной системой, подтверждается, то доставка данных гарантирована. Когда две системы участвуют в сеансе обмена информацией, то пакет-подтверждение посылается, чтобы сообщить об успешном прибытии каждой порции данных. Поэтому первым очень приятным свойством протоколов с установлением соединения является гарантированная доставка информации.

Кроме того, протоколы с установлением соединения работают по заданному пути. То есть каждый пакет отправляется по тому же пути, что и предыдущий, что облегчает поиск неисправностей в случае проблем с соединением. Используя такие средства, как пакетные анализаторы или модули проверки текущего состояния LAN, специалист легко может определить маршрут, по которому взаимодействовали два устройства, и диагностировать возможные проблемы.

Протоколы без установления соединений, такие как IP, работают (за редким исключением) на сетевом уровне модели OSI. Маршрутизаторы Cisco (и все остальные маршрутизаторы, присутствующие на рынке) также работают на сетевом уровне. Из чего можно сделать логический вывод о том, что все маршрутизируемые протоколы относятся к протоколам без установления соединений и что маршрутизаторы поддерживают обмен информацией только без установления соединений.

Протоколы без установления соединений используют при доставке данных принцип «наибольших усилий». Когда устройство пытается отправить информацию другой системе, между ними не устанавливается специальное соединение. Использование протоколов без установления соединений несколько рискованно. С ними связаны такие возможные опасности, как:

- Несоблюдение порядка доставки пакетов
- Потеря пакетов
- Повышенная сложность при поиске и устранении неисправностей

Так как между устройствами нет выделенного соединения, каждый отправленный ими пакет может перемещаться по пути, не совпадающему с остальными. Поэтому вероятность того, что пакеты прибудут не в том порядке, в котором были отправлены (если вообще прибудут), гораздо больше, чем при использовании протоколов с установлением соединений.

Главным недостатком протоколов без установления соединений является неминуемая потеря данных, находящихся в недоставленных пакетах. И хотя в большинстве случаев потери минимальны и едва заметны, но риск все-таки есть.

Еще одним неудобством, вызванным отказом от использования выделенного пути, является то, что сложность поиска и устранения неисправностей возрастает экспоненциально при увеличении количества маршрутизаторов в сети. Так как пакеты могут использовать различные пути, чтобы попасть в один и тот же пункт назначения, то чрезвычайно сложно предсказать, где же может возникнуть проблема.

Один из способов контроля маршрутизатора Cisco за доставкой данных из одной системы в другую – это использование статических маршрутов (когда возможно). Хотя статические маршруты, конечно же, подходят не для каждого случая (на самом деле они в большей или меньшей степени лишают маршрутизацию смысла), они могут помочь в особенно ненадежных системах.

Есть и явные преимущества использования протоколов без установления соединений. Обычно они быстрее и требуют меньших ресурсов, чем протоколы с установлением соединений. Именно поэтому они идеально подходят для маршрутизации. В условиях полиморфной природы маршрутизации протоколы без установления соединения предоставляют возможность быстро перемещать данные из системы в систему.

Так как протоколы без установления соединения не требуют установления сеанса связи, они могут быть быстрыми. Время, необходимое для установления и завершения соединения, на отправку уведомлений и ответы на них, в больших сетях накапливается. Поэтому для современных маршрутизируемых сетей так важно быстродействие протоколов без установления соединения.

Классовые и бесклассовые протоколы

Объяснять отличие классовых протоколов от бесклассовых лучше всего на примере IP. В том, что касается протоколов, в списке ваших приоритетов на первом месте должен стоять IP.

Причина, по которой различия классовых и бесклассовых протоколов рассматриваются на примере IP, заключается в том, что в последние годы IP стал и тем и другим, то есть его можно сконфигурировать и как классовый, и как бесклассовый протокол.

Классовый IP

По своей природе IP относится к классовым протоколам. Когда IP разрабатывался (вместе с первыми сетевыми окружениями), он был задуман как классовый протокол. Это означает, что IP может быть разделен на классы, каждый из которых будет ориентироваться на нужды конкретной сети клиента.

Классы используются для определения предельного размера среды на основании соотношения количества машин и сетей. Каждый класс предлагает разное количество адресуемых машин для адресуемой сети. Один класс может предлагать 127 адресуемых сетей и более 16 миллионов адресуемых машин в сети, а другой – 2 миллиона сетей и 254 машины в сети.

В случае с IP протокол разделен на три (общепринятых) класса. Классы были разработаны так, чтобы соответствовать потребностям организаций разных размеров.

Примечание

Хотя общепринятым считается разделение IP на три класса (A, B и C), на самом деле их пять. Классы A, B и C наиболее популярны при сетевой адресации. Но классы D и E могут использоваться в других целях, например для многоадресной передачи.

В классовом протоколе один адрес используется для задания как машины, так и сети на переменой основе. То есть та часть адреса, которая определяет сеть (а не машину), может иметь разный размер в зависимости от используемого класса. Например, в IP-сети класса A адрес 16.10.20.6 может быть разбит на адрес сети – 16.0.0.0 и адрес машины – 0.10.20.6. В классе же C адрес 225.198.40.9 может быть разбит на сетевой – 225.198.40.0 и адрес машины – 0.0.0.9.

Маршрутизатор определяет, какая часть адреса использована для сети, применяя вторичный адрес или маску. Маски (уникальные для классовых протоколов) указывают маршрутизирующему устройству, какие биты протокольного адреса представляют сеть. Эти вычисления являются ключом к функциональности маршрутизатора.

Бесклассовый IP

Проблема классовых протоколов, таких как IP, в том, что они конечны в том смысле, что существует определенный предел количества адресов, которые могут быть присвоены в течение жизни протокола. Поэтому стал популярным бесклассовый способ использования IP.

Хотя бесклассовые протоколы тоже используют один адрес для представления как сети, так и машины, но их поведение неизменно. Часть адреса, относящаяся к машине, всегда имеет одинаковую длину. Поэтому маршрутизирующие устройства могут легко и быстро определять сетевые биты адреса, которые никогда не меняются.

CIDR

В последние несколько лет широкое распространение получил протокол CIDR (Classless Inter-Domain Routing – бесклассовая междоменная маршрутизация). Разработанный с целью помочь удовлетворить растущий спрос на IP-сети класса B, CIDR является бесклассовой формой IP.

Когда маршрутизаторы сконфигурированы для использования CIDR, сетевые IP-адреса сгруппированы в суперсети. Внутри структуры IP-адреса биты сетевой части передаются адресной части машины, что обеспечивает более гибкую схему адресации.

Эффект создания суперсетей заключается в том, что возникает возможность маршрутизации информации к любой машине группы сетей на основе адреса суперсети. Адрес суперсети используется для определения того, какие сети принадлежат к конкретной группе.

Инкапсуляция

Протоколы способствуют движению данных из системы в систему, используя процедуру под названием *инкапсуляция*. В процессе инкапсуляции данные, пересылаемые из одной системы в другую, упаковываются так, чтобы промежуточные устройства сетевого взаимодействия могли определить, куда направляются данные.

Инкапсуляция представляет собой процесс деления сплошного потока данных на небольшие сегменты одинакового размера. Эти маленькие одинаковые пакеты легче отправлять и получать, так как их размер легко спрогнозировать всем системам, использующим один и тот же протокол. Поэтому сеть может настраиваться на работу с сегментированными данными конкретного размера.

Во время инкапсуляции протокол присоединяет к каждому сегменту или пакету свой собственный заголовок.

Примечание

Имя, которое дается каждому меньшему пакету данных, зависит от протокола и от уровня модели OSI, на котором он работает. Одни протоколы работают с пакетами, другие используют сегменты, кадры или ячейки. Но все это разные имена для одного и того же: подвергнувшегося инкапсулированию сегмента данных, готового к маршрутизации.

Содержимое заголовка зависит от протокола. (Поля заголовка будут описаны при рассмотрении конкретных протоколов в следующей главе. Заголовки IP обсуждались в главе 5 «Перемещение данных маршрутизаторами».) Но несколько полей присутствуют в заголовках почти всех протоколов.

Два из них – это адрес отправителя и адрес получателя. Эти адреса сообщают маршрутизатору и другим устройствам сетевого взаимодействия, откуда отправлен пакет данных и куда его необходимо доставить.

Следующее поле – это порядковый номер. Данное поле очень важно для инкапсуляции. Так как протоколы берут большие порции данных и делят их на более мелкие, легко управляемые части, то система должна уметь определять, как вновь собрать из них поток данных. (Помните, что маршрутизаторы работают на сетевом уровне без установления соединений, поэтому пакеты почти всегда прибывают на место назначения не в том порядке, в каком были отправлены. Обеспечением того, чтобы пакеты были снова собраны в правильном порядке, занимаются маршрутизируемые протоколы.)

Еще одно необходимое поле содержит размер передаваемого пакета данных.

Последнее поле, о котором мы сейчас поговорим, – это контрольная сумма. По существу, контрольная сумма представляет (в краткой и легко воспроизводимой форме) содержимое передаваемого пакета. После получения пакета (но до его обратной сборки) устройство назначения на основе полученных данных вычисляет контрольную сумму заново и сравнивает ее с полученной контрольной суммой, чтобы определить, не был ли пакет поврежден во время пересылки. Эта функция маршрутизируемых протоколов также жизненно важна для правильной работы маршрутизаторов.

Этот короткий, но достаточно подробный рассказ о маршрутизируемых протоколах служит введением в оставшуюся часть книги, которая познакомит вас непосредственно с протоколами и расскажет, как маршрутизаторы Cisco их используют. Теперь же мы углубимся в конфигурирование маршрутизатора Cisco с тем, чтобы он смог использовать свой первый протокол – IP.

Резюме

- Протоколы являются главной движущей силой маршрутизации.
- Протоколы можно разделить на маршрутизируемые и немаршрутизируемые, протоколы с установлением соединения и без установления соединения, классовые и бесклассовые.
- Протоколы с установлением соединения используют соединения для создания выделенных путей между передающими устройствами.
- Протоколы без установления соединения предлагают быстрое, требующее небольшого количества ресурсов, маршрутизируемое решение.
- Классовые протоколы делятся на классы, удовлетворяющие конкретным требованиям среды, в которой они используются.
- Бесклассовые протоколы относятся к разряду «универсальных».

Вопросы и ответы

Вопрос Зачем нужны маршрутизируемые протоколы, если маршрутизаторы используют протоколы маршрутизации?

Ответ Маршрутизаторы – это не единственные устройства, участвующие в передаче данных. Другие устройства используют маршрутизируемые протоколы для перемещения данных и управления ими. Маршрутизаторы используют протоколы маршрутизации, чтобы перемещать информацию от одного маршрутизатора к другому. Большая часть других устройств в сети использует маршрутизируемые протоколы.

Вопрос Почему маршрутизаторы работают только с протоколами сетевого уровня?

Ответ Сетевой уровень отвечает за составление карт сетей и их окружений. Поэтому для перемещения информации с места на место маршрутизаторам необходимы службы сетевого уровня, чтобы понять, куда должны быть переданы данные.

Тест

Вопросы

1. Относится ли IP к протоколам с установлением соединения или же без установления соединения?
2. Какой уровень модели OSI занимается управлением потоком данных?
3. В чем смысл инкапсуляции протоколов?
4. Что такое CIDR и в чем ее важность?

Ответы

1. IP – это протокол без установления соединения (а вот TCP – протокол с установлением соединения).
2. Транспортный уровень.
3. Инкапсуляция помогает разбить потоки данных на легко управляемые фрагменты и упаковать их так, чтобы промежуточные устройства могли понять, куда их следует отправить.
4. CIDR – бесклассовая междоменная маршрутизация, которая помогает продлить жизнь IP-схем за счет использования суперсетей.

9

Изучение основ IP

TCP/IP (Transmission Control Protocol/Internet Protocol – протокол управления передачей данных/протокол Интернета) – это самый популярный на сегодняшний день стек сетевых протоколов. Он содержит протокол, который вы, вероятнее всего, встретите при конфигурировании маршрутизаторов Cisco. Но в этой главе мы пока не будем заниматься установкой IP на маршрутизаторе Cisco. Данная глава – это что-то типа букваря, который поможет вам понять, как работает IP. (Конфигурирование IP на маршрутизаторе Cisco является темой следующей главы «Настройка протокола IP на маршрутизаторе Cisco».)

Примечание

В предыдущей главе говорилось, что в терминах модели OSI TCP относится к протоколам транспортного уровня, в то время как IP – протокол сетевого уровня. Два протокола дополняют друг друга, и, чтобы осознать целостную картину, необходимо понять, как работают они оба. Но так как маршрутизатор работает только с одним протоколом стека TCP/IP – IP, то здесь мы не будем говорить о TCP.

Как работает TCP/IP и почему он так популярен? Для того чтобы познакомить вас со всеми базовыми компонентами IP, в этой главе будут рассмотрены такие темы:

- IP
- Организация подсетей
- Надсеть IP
- IP и маршрутизаторы Cisco

IP

Все знакомы с протоколом IP и применяемой в нем адресацией. (Об этом рассказывалось в главе 6 «Запуск маршрутизатора и работа с ним», когда устанавливались адреса для интерфейсов Ethernet.) Сегодня практически невозможно работать в компьютерной индустрии и не знать, что такое IP. Но понимаете ли вы на самом деле, как он работает и почему он настолько популярен? В этом разделе мы поговорим о работе IP, об адресации и организации подсетей.

Примечание

IP облегчает работу сетевого уровня по созданию карты сетевого окружения. Используя адресную IP-схему, сетевой уровень может составить подробную картину сети вокруг хоста. IP работает только на сетевом уровне.

Стандартный IP-адрес выглядит так 128.95.95.178 – четыре части, содержащие по байту каждая. IP-адрес состоит из 32 битов или 4 байтов. Так как биты двоичны, то максимальное значение, которое достижимо в пределах одного байта, – это 255. То есть возможный диапазон IP-адресов выглядит как 0.0.0.0–255.255.255.255. (Эти числа достижимы, но не все они могут быть использованы.) Будьте осторожны и не используйте для своих маршрутизаторов один из зарезервированных адресов.

Список IP-адресов, которые считаются зарезервированными, приведен в табл. 9.1.

Таблица 9.1. Зарезервированные IP-адреса

Адрес	Двоичный	Причина резервирования
0.0.0.0	00000000.00000000. 00000000.00000000	Адрес не может состоять из одних нулей. Используется протоколом RIP для маршрутизации
255.255.255.255	11111111.11111111. 11111111.11111111	Адрес не может состоять из одних единиц. Используется для широковещания
127.0.0.1	01111111.00000000. 00000000.00000001	Зарезервировано для внутреннего петлевого контроля

В зависимости от класса адреса, от одного до трех байтов заняты под идентификацию машины и от трех до одного байта используются для определения сети. Первая часть адреса задает сеть, а вторая – машину. Умение отличать хост от сети очень важно для маршрутизации.

Примечание

Адрес хоста (машины) также иногда называют адресом узла.

Для того чтобы отслеживать количество адресов, выделенных организациям различных размеров, IP-адреса разделены на три класса: A, B и C. Существует также малоизвестный (и еще меньше используемый) класс D, о котором мы не будем рассказывать. (Класс D главным образом используется для многоадресной передачи.) Зная класс адреса, вы сможете правильно определить маску подсети при конфигурировании интерфейса.

Адреса класса A

У адресов класса A первый байт адреса представляет собой сеть, а остальные три байта – узлы.

Разбиение адреса класса A на адрес сети и адрес узла изображено на рис. 9.1.

Сетевые адреса класса A имеют диапазон от 1 до 127. Следовательно, IP-адрес, начинающийся с числа, входящего в этот диапазон, относится к классу A. Поясним это на двоичном уровне. Первый бит первого октета адреса класса A – это всегда 0. Например, адрес 127.0.0.0 в двоичной записи выглядит так: 01111111.00000000.00000000.00000000. Заметьте, что первый бит равен 0.

Адреса класса B

В адресах класса B сеть представляют два первых байта адреса, а узел – два других. Первый бит адреса класса B имеет значение от 128 до 191. IP-адрес класса B, разделенный на адрес сети и адрес узла, изображен на рис. 9.2.

Исследуя двоичную запись адреса, мы обнаружим, что все адреса класса B начинаются с 10, то есть два первых бита первого октета – это всегда 10. Например, адрес 140.75.0.0 соответствует двоичному 10001100.01001011.00000000.00000000. Первые два бита – это 10.

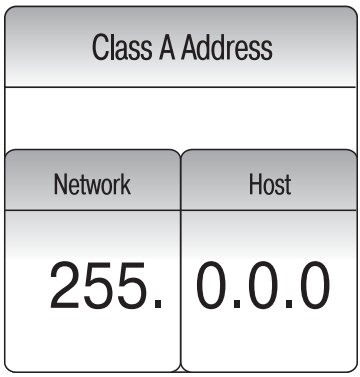

Рис. 9.1. *Адрес класса A*

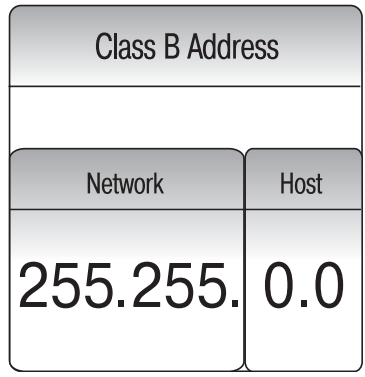

Рис. 9.2. *Адрес класса B*

Адреса класса C

В адресах класса C три первых байта определяют сеть, а последний – узел. Разрешенные значения первого байта сетевого адреса в данном случае входят в диапазон от 192 до 223. Разбиение адреса класса C на адрес сети и адрес узла изображено на рис. 9.3.

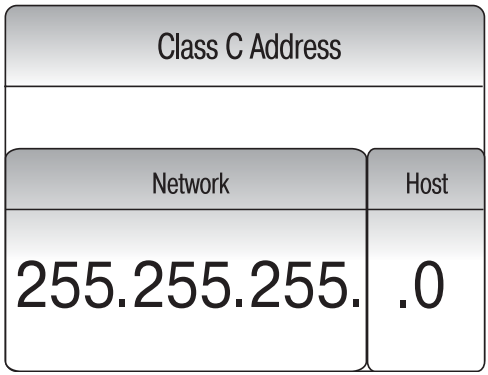

Рис. 9.3. *Адрес класса C*

Давайте посмотрим на адрес класса C в двоичном формате: 198.40.50.0 соответствует двоичному адресу 11000110.00101000.00110010.00000000. Все адреса класса C начинаются с трех цифр: 110.

Примечание

Вы могли обратить внимание на то, что верхняя граница разрешенных значений для адресов класса C равна 223, в то время как IP-адреса могут достигать значения 256. Такой зазор оставлен для класса D (223-239) и класса E (240-255). Не тратьте время на раздумья об этих адресах, они нечасто используются.

О существовании таких диапазонов полезно знать при поиске и устранении неисправностей. Например, если вы конфигурируете интерфейс маршрутизатора для сети класса C и присваиваете ему адрес 230.230.230.0 с маской подсети 255.255.255.0 (правильная маска подсети для сети класса C), то можете столкнуться с проблемами.

В данном конкретном случае устройство не сможет взаимодействовать ни с какими другими устройствами подсети 255.255.255.0. На первый взгляд адрес 230.230.230.0 выглядит вполне законно, хотя на самом деле это адрес класса D.

Маска подсети

Маска подсети используется IP для того, чтобы различать адрес сети и адрес узла. Чтобы понять, как работает маска подсети, преобразуем IP-адрес в двоичный формат. Возьмем, к примеру, адрес класса C – 198.68.85.114:

```
11000110.01000100.01010101.01110010
```

Маска подсети для IP-адреса класса C выглядит как 255.255.255.0, переведем и ее в двоичный формат:

```
11111111.11111111.11111111.00000000
```

Как она может помочь протоколу IP узнать, где сетевой адрес, а где адрес узла? Если вы посмотрите на двоичное представление адреса и маски, то увидите, что сеть обозначена единицами, в то время как часть, относящаяся к адресу узла, состоит из нулей. Теперь все кажется очевидным, но когда вы начинаете организовывать подсети в своей сети, оказывается, что есть и сложности.

Примечание

Используемый класс IP-адресов определяет, какую маску вам следует применять. Приведем существующие по умолчанию маски подсети для трех классов адресов:

Класс A = 255.0.0.0

Класс B = 255.255.0.0

Класс C = 255.255.255.0

Организация подсетей – это техническая операция, которой вам точно не удастся избежать при работе с маршрутизаторами Cisco. Чем больше вы этим занимаетесь, тем понятнее становится, как организовывать подсети, но такая задача может быть и сложной, а в некоторых случаях и мучительной. Я бы предложил вам сначала нарисовать схемы своих подсетей на бумаге, а потом уже реализовывать их. Но давайте все же немного поговорим о процессе разбиения сети на подсети, чтобы познакомить вас с данным вопросом.

Организация подсетей

При разделении сети на подсети биты из той части адреса, которая задает адрес узла, передаются в сетевую часть адреса. Благодаря этому одна лицензия на IP-сеть может использоваться для адресации более чем одной сети (подсети). На рис. 9.4 показана сеть, в которой организация подсетей принесла бы пользу.

Требуется разбить лицензию 10.0.0.0 так, чтобы получилось три подсети. С чего начать? Начнем с того, что посмотрим на маску подсети. (Даже если вы создаете три сети, у них должна быть общая маска подсети.) Если преобразовать маску подсети в двоичный формат, то легче увидеть, как организуются подсети.

Маска подсети для 10.0.0.0 – 255.0.0.0 или (в двоичном виде)

```
11111111.00000000.00000000.00000000.
```

Теперь следует определить количество бит, которые будут переданы сетевому адресу, чтобы предусмотреть еще три сети. Предположим,

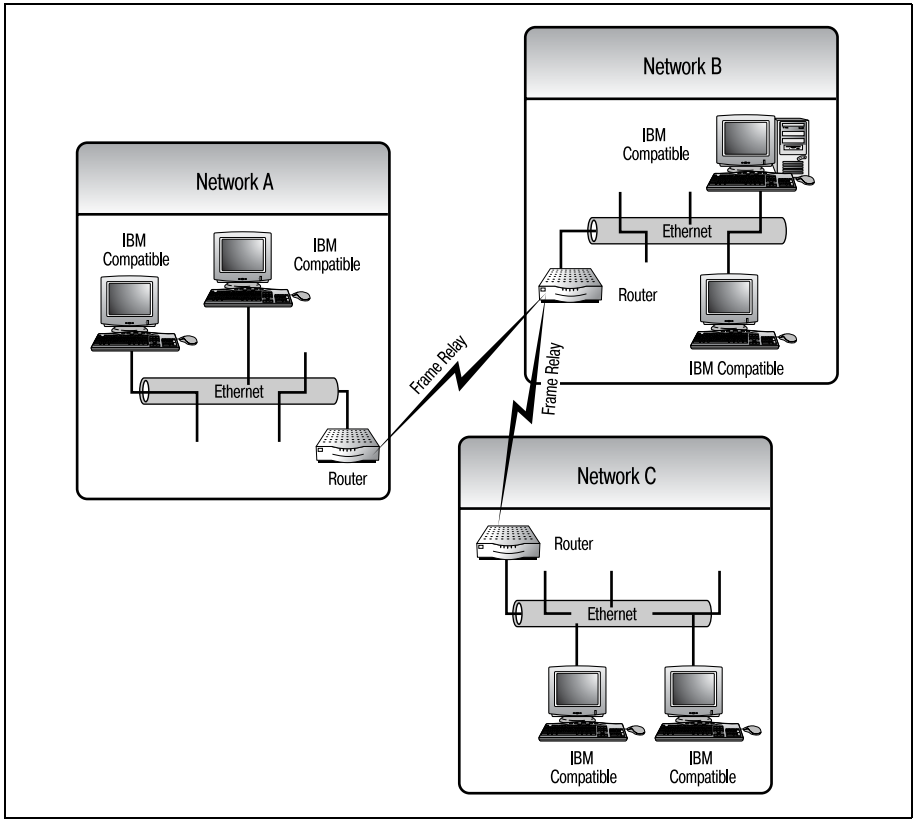

Рис. 9.4. *Сеть, не разбитая на подсети*

например, что мы передаем два бита из той части маски, которая соответствует адресу узла, в сетевую часть маски:

```
11111111.11000000.00000000.00000000
```

Получается новая маска подсети – 255.192.0.0.

При подсчете количества бит, которые должны быть переданы в сетевую часть адреса, чтобы обеспечить создание требуемого количества подсетей, можно использовать такое уравнение:

$2^x - 2 =$ количество адресов (где x – количество бит в адресе)

Во втором байте рассмотренного примера мы использовали два бита для сетевого адреса. Следовательно, количество сетей равно $2^2 - 2 = 2$. Для адресов узлов осталось всего 22 бита, значит, их общее количество равно $2^{22} - 2 = 4\,194\,302$.

Примечание

Почему следует вычесть 2 из общего количества адресов? Дело в том, что использование адресов, состоящих из одних единиц или нулей, не разрешено, поэтому необходимо вычесть два эти варианта.

Второе уравнение применяется для вычисления сетевых адресов. Размер интервала между сетевыми адресами равен разности 256 и новой маски подсети. Каждый сетевой адрес должен быть кратен этой разности, при этом быть положительным и не превышать значение маски подсети:

i-й сетевой адрес $\quad yi = i * (256 - x),\ i = 1, 2, \ldots,$ [меньше чем] x

где x – это новая маска подсети. В нашем примере маска подсети равна 192, значит

$$256 - 192 = 64$$

Следовательно, допустимыми являются сети 10.**64**.0.0 и 10.**128**.0.0 (128 = 64 + 64). Если еще раз прибавить 64, то получится 192, то есть наша маска подсети.

Чтобы успешно разбить на подсети пример, представленный на рис. 9.4, необходимы три сети. Два бита, добавленные к сетевому адресу, не привели к образованию достаточного количества сетей. Если же добавить три бита в маску подсети, чтобы получить 255.224.0.0 (224 = = 128 + 64 + 32), то образуется шесть сетей ($2^3 - 2$), по 2 097 150 узлов в каждой. Очевидно, что нам не потребуется так много узлов, но прежде всего нас интересует количество сетей.

С помощью организации подсетей можно лучше управлять трафиком между ними. Если не создавать подсети, то трафик (порождаемый широковещательными пакетами и CSMA/CD-пакетами) начинает отрицательно влиять на производительность сети. После же организации подсетей любой поток информации, у которого и отправитель и адресат находятся в одной подсети, остается в пределах этой подсети, «не лихорадя» сетевое окружение.

При создании подсетей биты из адреса узла передаются в сетевую часть адреса, при организации надсетей этот процесс происходит с точностью до наоборот. Образуется «надсеть» высшего уровня, которой принадлежат все ваши сети. Создание надсети применяется при бесклассовой междоменной маршрутизации (CIDR).

Надсеть IP

В связи с растущей популярностью CIDR появилась необходимость объединения группы сетей в надсеть. Так как административные про-

цедуры выделения адресов мучительно медленно работают с большими классами IP-адресов, потребовалось найти способ продлить жизнь оставшихся адресов. Так и пришли к созданию надсетей.

Когда группа адресов объединяется в надсеть, создается маска, которая показывает, что отдельные сети принадлежат к одной большой супергруппе. Например, если окружению присвоены два сетевых адреса: 215.50.25.0 и 215.50.26.0, то можно создать надсеть, которая бы объединила и связала две сети для использования в рамках одной и той же физической среды.

Двоичная запись маски подсети 255.255.255.0 (маска класса C для обеих упомянутых сетей) выглядит как 11111111.11111111.11111111.00000000. При организации подсетей биты передаются из адреса узла в сетевую часть адреса, а при создании надсети биты «покидают» сетевую часть адреса, чтобы уменьшить количество сетей.

Если передать два бита из сетевой части адреса в адрес хоста, то маска превратится в 11111111.11111111.11111100.00000000 (или 255.255.252.0). Будучи наложенной на сетевой адрес 215.50.25.0, эта маска сообщит маршрутизатору, что две сети включены в надсеть. Если адреса сетей начинаются с 215.50.25.0, то маска распространяется на две сети: 215.50.25.0 и 215.50.26.0.

Вспомните, если использовать те же вычисления, которые были представлены в начале раздела, то получится, что передача двух битов из адреса узла в сетевую часть адреса включит в состав вашей надсети две (то есть $2^2 - 2$) сети.

Примечание

Если вы беспокоитесь о том, как маршрутизатор узнает, что новая маска 255.255.252.0 относится не к подсети класса B, а к надсети класса C, то ответ очень прост. Первый октет сетевого адреса выглядит как 215.x.x.x. Только адрес сети класса C может начинаться с 215; поэтому новая маска должна относиться к надсети класса C.

Легче понять надсети, если рассматривать их во взаимосвязи с подсетями. Чтобы поднатореть в этом вопросе, попрактикуйтесь на бумаге в создании как подсетей, так и надсетей.

IP и маршрутизаторы Cisco

Для работы маршрутизаторов Cisco необходимо использование IP. Чем лучше вы разберетесь в IP на начальном этапе обучения маршрутизации Cisco, тем лучше будете осознавать дальнейшие темы, представленные в этой книге. Вскоре вы узнаете, что многие протоколы, как маршрутизируемые, так и протоколы маршрутизации, или базируются на IP, или же зависят от него.

Все маршрутизаторы поставляются с возможностью маршрутизировать IP. То есть функциональный пакет Cisco IOS «Basic IP» по умолчанию поставляется со всеми маршрутизаторами Cisco. Другие маршрутизируемые протоколы, такие как IPX, требуют установки дополнительного функционального пакета. Поэтому мы потратили чуть больше времени на обсуждение этого протокола.

В мире маршрутизаторов Cisco избежать использования IP невозможно. Это не только главный протокол Интернета; инструментальные средства, основанные на IP, являются неотъемлемой частью обеспечения работы маршрутизаторов Cisco. Такие средства, как ping, traceroute и telnet, являются продуктами стека протоколов TCP/IP и имеют важное значение для работы маршрутизатора.

В следующей главе мы изучим конфигурирование маршрутизатора для работы с IP, а также связанные с IP команды и программы. Итак, давайте займемся конфигурированием наших маршрутизаторов Cisco.

Резюме

- IP является важным элементом работы маршрутизаторов Cisco.
- Сетевые адреса IP класса A можно идентифицировать по первому нулевому биту первого октета в двоичной записи.
- Два первых бита первого октета сетевых адресов класса B содержат 10.
- Узнать сетевые адреса класса C можно по цифрам 110 в первых двух битах первого октета адреса, записанного в двоичном формате.
- Создание подсетей – это процесс, в котором биты заимствуются у адреса узла IP-адреса в пользу сетевой части адреса, в результате чего сетевой администратор может работать с несколькими сетями.
- Создание надсетей – это процесс, в котором биты передаются из сетевой части IP-адреса в адрес узла. Образуется меньше сетей, с которыми администратор может работать, но зато в каждой сети появляется возможность организации большего количества узлов.

Вопросы и ответы

Вопрос Если TCP и IP входят в один и тот же стек протоколов, то почему маршрутизаторы работают только с IP, но не с TCP?

Ответ Хотя два протокола входят в один стек, они были созданы для выполнения различных задач. TCP был разработан для нужд транспортного уровня модели OSI, в то время как IP занимается адресацией на сетевом уровне, где работают и маршрутизаторы.

Вопрос Почему IP поддерживается всеми функциональными пакетами Cisco IOS?

Ответ IP – это основной протокол всех маршрутизаторов. Вне зависимости от того, состоит ваша среда маршрутизации из двух маленьких сетей или из группы веб-серверов, IP будет тем протоколом, который необходим вам для маршрутизации.

Тест

Вопросы

1. Сколько классов было создано для IP?
2. Какая маска подсети является стандартной для IP-адресов класса C?
3. К какому классу относится IP-адрес 126.45.30.1?
4. Какой IP-адрес зарезервирован для петлевого контроля?

Ответы

1. Пять (от A до E)
2. 255.255.255.0
3. К классу A
4. 127.0.0.1

10

Настройка протокола IP на маршрутизаторе Cisco

В главе 9 мы обсуждали один из наиболее популярных сегодня протоколов в маршрутизации – IP. Но существует также ряд других протоколов и инструментальных средств, непосредственно связанных с IP, о которых еще не говорилось. Стек протоколов TCP/IP включает в себя не только те два протокола, которые дали ему имя (TCP и IP). В этой главе будут рассмотрены два таких дополнительных средства: ICMP (Internet Control Message Protocol – протокол управления сообщениями в Интернете) и Telnet. После знакомства с ними и изучения команд конфигурирования IP на маршрутизаторах Cisco можно будет считать, что вы полностью освоили использование и работу IP. Чтобы достичь этой цели, мы рассмотрим такие темы:

- IP и интерфейсы Cisco
- ICMP
- Использование утилит ICMP
- Telnet
- Удаленное администрирование с использованием Telnet
- `rlogin`

ICMP – это протокол, тесно связанный с IP. ICMP является мощным инструментом диагностики в мире маршрутизации. Освоение ICMP и его утилит поможет вам обнаружить большую часть проблем, которые могут случиться с маршрутизаторами Cisco, например таких, как ошибки при соединении типа «точка-точка». Часть этой главы посвящена непосредственно ICMP.

Еще один широко используемый протокол, который также напрямую связан с IP, – это Telnet. Если вы имеете опыт работы с компьютерами, Telnet может быть вам знаком. Маршрутизаторы Cisco используют этот протокол в двух целях. Во-первых, ПК может использовать Telnet для взаимодействия с маршрутизатором Cisco, чтобы получить доступ к командной строке. Второе (и наиболее важное) предназначение Telnet заключается в том, что он предоставляет маршрутизаторам Cisco возможность удаленно администрировать другие маршрутизаторы Cisco. То есть, используя Telnet, один маршрутизатор Cisco может удаленно администрировать несколько других маршрутизаторов Cisco. Возможность администрировать несколько серверов, не меняя при этом местоположения, становится бесценной при работе в больших и очень больших средах маршрутизации.

Эта глава завершает наш разговор об IP. Разделы главы – это кусочки той большой картины, которой является стек протоколов TCP/IP. IP – это, безусловно, самый важный протокол в современной маршрутизации, и осмысление каждого аспекта его работы необходимо для понимания внутренних механизмов маршрутизации. Когда вы дочитаете главу до конца, то будете обладать хорошим запасом базовых знаний.

Теперь, когда все вводные слова сказаны, давайте перейдем к конфигурированию маршрутизатора для работы с IP. Затем мы сможем погрузиться в изучение связанных с ним программ и их использование.

IP и интерфейсы Cisco

IP как протокол уже должен быть установлен на вашем маршрутизаторе. Все функциональные пакеты Cisco IOS, кроме IPX Basic, содержат IP. Поэтому для того, чтобы IP заработал, не требуется устанавливать никакого программного обеспечения. В этой главе мы поговорим об использовании возможностей IP.

Все настройки IP производятся в режиме конфигурирования интерфейсов. Чтобы войти в режим конфигурирования интерфейсов маршрутизатора Cisco, выполните такие команды:

```
Router>enable
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-if)#
```

Команда `enable` активирует привилегированный режим работы командного интерпретатора Exec. В нем мы выполняем команду `configure terminal`. Маршрутизатор переходит в режим глобального конфигурирования.

Используем команду `interface ethernet 0` для перехода от глобального конфигурирования к конфигурированию интерфейсов (будем настраи-

вать интерфейс Ethernet 0). В режиме конфигурирования интерфейсов достаточно одной простой команды, и IP уже может использоваться на вашем маршрутизаторе:

```
Router(config-if)#ip address 10.156.4.16 255.255.0.0
```

Команда `ip` имеет ряд параметров. Ключевое слово `address` указывает, что будет введен IP-адрес для интерфейса и маска подсети. Такая команда разрешает работу IP для одного интерфейса маршрутизатора. Повторите ее для всех физических интерфейсов, которые должны работать с IP.

Когда интерфейсы настроены, вы можете включить маршрутизацию, опять-таки, посредством одной простой команды:

```
Router(config)#ip routing
```

Команда `ip routing` включает маршрутизацию IP-пакетов между всеми сконфигурированными интерфейсами. Может также возникнуть необходимость разрешить бесклассовую маршрутизацию IP с помощью отдельной команды.

Если вы знаете, что в вашей сети будет использоваться бесклассовый IP, то следует разрешить маршрутизатору работу с ним. (Бесклассовый IP обсуждался в главе 8 «Введение в маршрутизируемые протоколы» и главе 9 «Изучение основ IP».) Команда, разрешающая маршрутизатору работать в бесклассовом режиме, выглядит так:

```
Router(config)#ip classless
```

Заметьте, что команда `ip classless` выполняется в режиме глобального конфигурирования, так как возможность бесклассовой маршрутизации IP затрагивает все интерфейсы.

Когда маршрутизатор получает пакет, он пытается установить соответствующую сеть по своей таблице маршрутов. Если в таблице для пункта назначения пакета не найдено соответствия, он пересылается в надсеть. Например, если у маршрутизатора существуют определенные маршруты для сетей 128.46.69.0 и 128.45.77.0 и поступает пакет с адресом назначения 128.46.68.15, то он будет перенаправлен в сеть 128.46.68.0, так как 128.46.x.x – это надсеть наиболее близкого соответствия.[1]

Наконец, если ваш маршрутизатор не является маршрутизатором «последней надежды» сети, то следует его задать. То есть конфигурируемый маршрутизатор должен знать, что ему делать с пакетами, для которых не определены маршруты. Чаще всего такие пакеты отправляют в другую сеть, для чего необходимо использовать шлюз. Поэтому

[1] Непонятно, что имелось в виду по поводу перенаправления в сеть 128.46.68.0, так как у маршрутизатора нет такого маршрута. – *Примеч. науч. ред.*

маршрутизатор последней надежды также может быть шлюзом по умолчанию.

Примечание

Маршрутизатор последней надежды – это тот, на который все остальные маршрутизаторы сети пересылают пакеты, для которых не заданы маршруты. Такой маршрутизатор чаще всего является единственным, имеющим прямое соединение с Интернетом.

Можно с уверенностью предположить, что в большинстве случаев, когда на маршрутизатор приходит пакет, для которого нет предопределенного маршрута, его следует переслать в Интернет. Поэтому маршрутизатор, соединяющий вашу сеть с Интернетом, должен быть настроен как маршрутизатор последней надежды для всех остальных устройств сети.

Чтобы установить адрес шлюза по умолчанию, используйте команду `ip default-gateway`:

```
Router(config)#ip default-gateway 198.2.65.1
```

Отметьте, что и эта команда выполняется в режиме глобального конфигурирования, следовательно, будет применена ко всем интерфейсам. В заключение используйте команду `ip route` для определения статических путей IP:

```
Router(config)#ip route 198.52.2.0 255.0.0.0 Ethernet 0 100 perm
```

Команда `ip route` определяет для маршрутизатора прямой путь к указанной сети. Устанавливая статические пути, вы можете управлять потоком данных вне вашего маршрутизатора. Но команда `ip route` требует указания нескольких параметров. Давайте немного поговорим о них.

За `ip route` должно следовать определение статического пути, то есть должны быть указаны IP-адрес сети и маска подсети.

Так как эти настройки производятся в режиме глобального конфигурирования, необходимо указать, для какого интерфейса маршрутизатора определяется путь. Поэтому следующий параметр задает физический интерфейс, на который будут пересылаться все пакеты, соответствующие данному статическому пути: `Ethernet 0`.

Следующий параметр выбирается в некоторой степени случайно, и, если хотите, то можете его не включать. Этот параметр, `100`, определяет административную метрику маршрута. Административная метрика используется в тех случаях, когда для некоторой сети определено несколько путей. Тогда используется путь с наименьшей метрикой, а другие становятся резервными.

Последний параметр, `perm`, определяет весь статический путь как постоянный (permanent). Поэтому он будет сохранен в конфигурационном файле `startup-config`. Чтобы определить временные пути, не при-

меняйте параметр perm. Временные пути не сохраняются в конфигурационном файле, и после перезагрузки маршрутизатор о них забывает.

Давайте проработаем несколько сценариев IP-маршрутизации. Например, как следует сконфигурировать маршрутизатор сети, изображенной на рис. 10.1, чтобы он использовал IP и маршрутизировал его между сетями A и B?

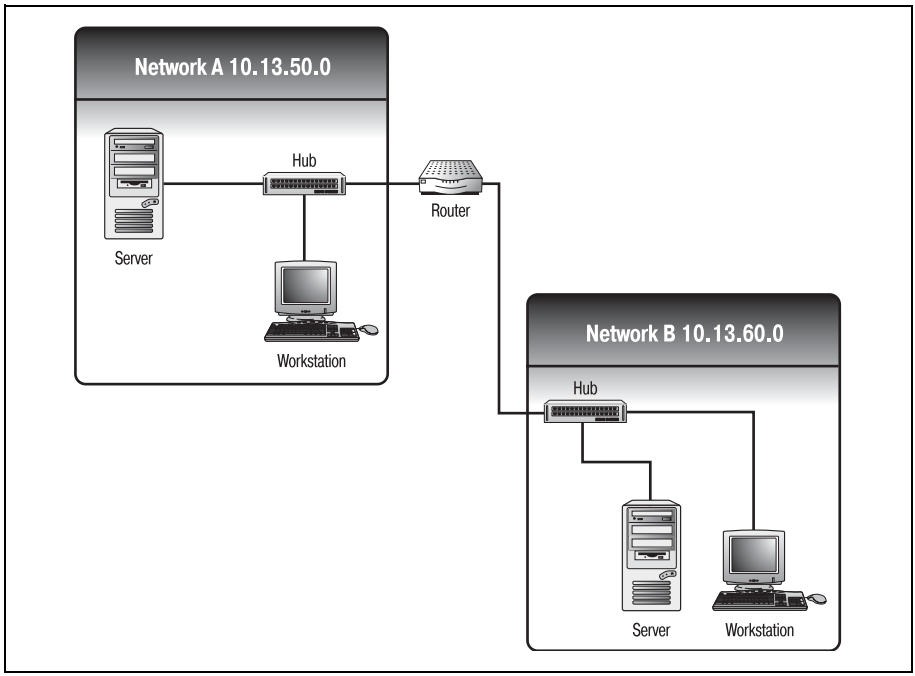

Рис. 10.1. *Пример IP-сети*

Чтобы разрешить маршрутизацию, необходимо выполнить такие команды:

```
Router>enable
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-if)#ip address 10.13.50.1 255.255.255.0
Router(config-if)#^Z
Router#configure terminal
Router(config)#interface ethernet 1
Router(config-if)#ip address 10.13.60.1 255.255.255.0
Router(config-if)#^Z
Router#configure terminal
Router(config)#ip routing
Router(config)#ip route 10.13.50.0 255.255.255.0 e0 perm
Router(config)#ip route 10.13.60.0 255.255.255.0 e1 perm
```

Хотя такой план действий может показаться слишком простым с практической точки зрения, по существу, именно он лежит в основе функционирования любой сети.

Однако для того чтобы обладать всеми базовыми знаниями, необходимыми для конфигурирования IP на маршрутизаторах Cisco, вам не помешает знать о существовании еще одного сценария. Следующий пример иллюстрирует часто встречающийся вариант сетевой среды. На рис. 10.2 изображены три сети, объединенные вместе.

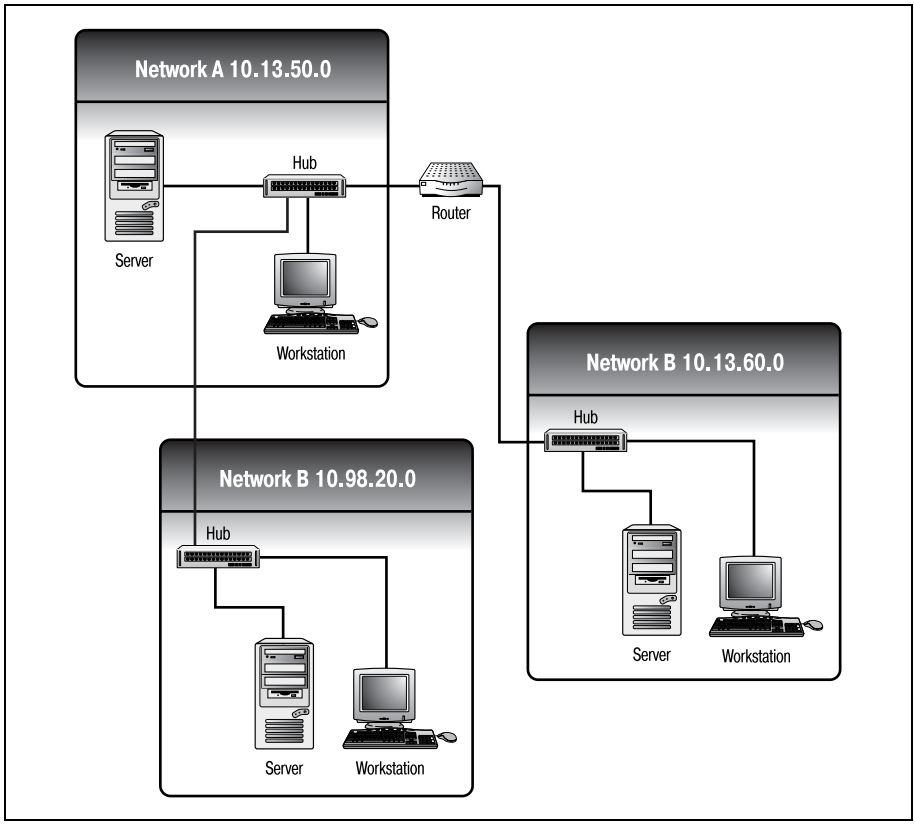

Рис. 10.2. *Сеть, объединяющая три подсети*

В данном случае существующая сеть приобретает третий, более мелкий сегмент. Проблема в том, что два физических интерфейса маршрутизатора уже использованы.

Решение состоит в использовании параметра `secondary` команды `ip address` для указания вторичного IP-адреса для одного из интерфейсов.

Примечание

Интерфейсам маршрутизатора Cisco можно установить вторичный адрес, что будет способствовать разрешению ситуаций, подобных описанной выше. Но любой пакет,

отправленный с этого интерфейса, будет иметь в качестве адреса отправителя первичный адрес интерфейса.

Чтобы сконфигурировать вторичный адрес для маршрутизатора, изображенного на рис. 10.2, используйте такую команду:

```
Router(config-if)#ip address 10.98.20.1 255.255.0.0 secondary
```

Итак, некоторое количество команд, необходимых для настройки IP на маршрутизаторе, вы уже изучили. Теперь можно спокойно приступить к обсуждению остальных связанных с IP программ и функций.

ICMP

ICMP очень тесно связан с IP. Многие устройства используют сервисы ICMP для текущего контроля над сетью и получения информации о сетевом окружении. Тесная связь с IP имеет свои плюсы и минусы. Преимущество в том, что благодаря распространенности IP многие устройства допускают применение ICMP. С другой стороны, все же остается немного персональных компьютеров и других устройств, которые не используют IP, и, к сожалению, ICMP не в состоянии обнаружить и продиагностировать такие устройства.

Большая часть ПК и других пригодных для работы в сети устройств обладает способностью использовать такие средства, как `ping` и `traceroute`. Эти программы применяют ICMP-пакеты для проверки возможности соединения между устройствами. Если вы будете знать, как работают данные программы, и сумеете использовать их с выгодой для себя, это поможет содержать вашу сеть и маршрутизатор в хорошим состоянии.

Далее в разделе рассказывается о технологии и принципах ICMP. Для установки ICMP на маршрутизаторе не требуется никакого дополнительного конфигурирования, но вы должны знать, как им пользоваться.

Примечание

Хотя для установки ICMP или сервисов ICMP на маршрутизаторе не требуется никакого дополнительного конфигурирования, необходимо наличие хотя бы одного работающего интерфейса, использующего IP. Если ни один из интерфейсов не сконфигурирован для работы с IP, вы все равно получите доступ к командам ICMP, только они не будут работать до тех пор, пока протокол IP для интерфейса не будет включен.

Утилиты ICMP обеспечивают возможность достаточно простой диагностики сетевых проблем. Поэтому для того чтобы научиться работать с маршрутизатором Cisco, вам необходимо знать эти команды и понимать, что они выводят.

Использование утилит ICMP

ICMP – это протокол, который используется почти исключительно для проверки связности узлов сети и диагностики возможных проблем. Из-за сложности маршрутизации без соответствующих средств тяжело разобраться даже с простыми вопросами. Давайте рассмотрим на примере, каким сложным может оказаться выявление проблем в маршрутизируемой среде.

Сетевые окружения становятся все больше и сложнее. Уже нередки случаи, когда в одной сети работают от 5 до 10 независимых маршрутизаторов. Но многие сети не ограничены стенами комнат. Соединения с Интернетом добавляют в среду маршрутизации сотни, если не тысячи маршрутизаторов. Типичная маршрутизируемая среда представлена на рис. 10.3.

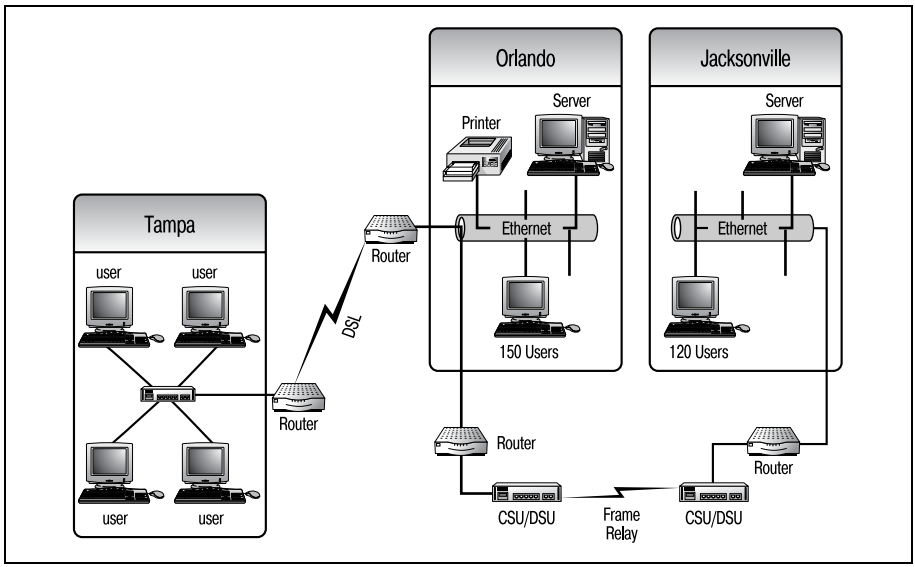

Рис. 10.3. *Характерная среда маршрутизации*

Уровень сложности сети возрастает с увеличением количества маршрутизаторов. Одно часто встречающееся осложнение, возникающее в результате добавления новых маршрутизаторов, – это необходимость отслеживания пакетов внутри сети. Используя ICMP, администратор может следить за маршрутизацией пакетов от начала и до конца. Это не только помогает инженерам и проектировщикам определить, есть ли проблемы с соединениями в сети, но и обеспечивает предоставление информации о том, правильные ли пути используют маршрутизаторы.

Маршрутизаторы Cisco могут применять две утилиты ICMP (`ping` и `traceroute`) для всеобъемлющей оценки маршрута. Обе эти программы обладают мощными диагностическими возможностями.

Ping

Команда `ping` используется для проверки существования конечной системы. Пусть, например, вы пытаетесь передать по TFTP конфигурационный файл от маршрутизатора к TFTP-серверу и в ответ получаете такое сообщение:

```
%Error opening tftp://10.16.4.153/
```

Вам сообщают, что сервер, с которым вы пытаетесь вступить в контакт, не отвечает. Очевидно, что существует миллиард причин, по которым сервер может не отвечать. Чтобы уменьшить это количество, можно применить команду `ping`.

Программа `ping` использует запросы отклика ICMP для проверки доступности сервера, аналогично тому, как гидролокатор подводной лодки использует звуковые импульсы (которые долго называли «pings») для проверки наличия объектов в близлежащих водах. Обращаясь к IP-адресу сервера с запросом отклика и ожидая ответа, вы можете узнать, работает ли машина и правильно ли функционирует протокол. Так вы сужаете круг возможных проблем.

Примечание

Ping работает, передавая выбранному IP-адресу запросы отклика ICMP. Когда устройство с указанным адресом получает ICMP-пакеты, оно отражает их обратно отправителю.

Однако единственное, в чем вы можете быть уверены, используя `ping`, — это в том, что устройство включено и работает правильно; кроме этого мало что можно узнать. Например (возвращаясь к последнему примеру), если вы посылаете ping-запрос на IP-адрес TFTP-сервера и не получаете ответа, это может указывать на несколько проблем. Сервер может быть выключен, сервер может быть не сконфигурирован для работы с IP, а может быть, дело совсем не в нем, и неправильно сконфигурирован маршрутизатор, с которого вы посылаете запрос. Узнать, что именно произошло, пока невозможно, но, по крайней мере, есть с чего начать.

Будучи командой Cisco IOS, `ping` имеет ряд дополнительных возможностей, которыми может не обладать стандартная версия. Именно благодаря этим дополнительным свойствам `ping` относится к тем немногим командам, которые можно исполнять и в пользовательском, и в привилегированном режимах. Давайте посмотрим на команду `ping` в пользовательском режиме.

Примечание

Помните, что для использования любых программ ICMP на вашем маршрутизаторе должен работать хотя бы один интерфейс с установленным IP. Если не будет до-

ступна ни одна линия передачи данных, то при попытке выполнить команду вы увидите такое сообщение об ошибке:

```
% Unrecognized host or address, or protocol not running.
```

ping (пользовательский режим)

Стандартная версия команды ping доступна в пользовательском режиме работы командного процессора. В результате ввода ping после приглашения на ввод пользовательского режима появится сообщение:

```
Router>ping
% Incomplete command.
```

Дело в том, что команде ping необходим как минимум один параметр. Используя справочную систему Cisco IOS, можно вывести список параметров ping:

```
Router>ping ?
 WORD Ping destination address or hostname
 ip  IP echo
 tag  Tag encapsulated IP echo
```

Примечание

С двумя дополнительными параметрами: ip и tag мы не будем иметь дела. Первый из них, ip, указывает на то, что при отправке ping-запроса будет использоваться протокол IP. Cisco разрешает использовать для этой цели и другие протоколы, указывая параметры apollo, appletalk, clns, decnet, ipx, vines и xns. Мы сосредоточимся на IP (значение по умолчанию).

Параметр tag применяется для определения инкапсуляции пакета с тегами. Это используется главным образом при коммутации и не будет описано в данной книге.

Чтобы команда была полной, следует указать IP-адрес устройства, которому вы хотите отправить ping-запрос:

```
Router>ping 10.16.4.152
```

Выходные данные стандартной команды ping приведены ниже. На первый взгляд, сообщение может показаться непонятным, но если разделить его на части, то все прояснится.

```
Router>ping 10.16.4.152

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.16.4.152, timeout is 2 seconds:
!!!!!
Success rate is 100 percent, round-trip min/avg/max = 1/3/4 ms
```

Первая строка вывода (Type escape sequence to abort.) на самом деле генерируется IOS до того, как выполняется команда. Это сообщение о

состоянии, информирующее пользователя о том, что процесс выполнения команды можно прервать. Чтобы прекратить выполнение ping, используйте комбинацию клавиш <Ctrl>+<Shift>+<6>.

Вторая строка – это тоже сообщение о состоянии, генерируемое Cisco IOS. Строка повторяет смысл команды ping. В нашем случае команда ping собирается отправить пять 100-байтных пакетов по IP-адресу 10.16.4.152. Тайм-аут установлен в 2 секунды.

Тайм-аут, или TTL (Time To Live, время жизни), – это часы, используемые для определения момента, когда можно отбросить пакет. Когда время жизни пакета истекает, он удаляется хранящим его устройством. Поэтому отправляющее устройство, не получив ответа в течение 2 секунд, делает вывод о том, что адрес назначения недостижим. Иначе отправитель бесконечно ждал бы ответ, который мог бы не прийти никогда.

Следующая строка выведенного сообщения иллюстрирует исполнение ping. Каждый символ (в нашем примере – восклицательный знак) представляет один пакет, отправленный маршрутизатором по адресу назначения. Восклицательные знаки означают, что ICMP-пакеты были успешно возвращены обратно маршрутизатору. Если бы вместо восклицательных знаков стояли точки, это бы означало потерю пакетов; точка показывает, что адрес назначения не откликнулся на команду ping.

В данной строке могут появиться восемь символов:

- ! (успех)
- . (неудача)
- U (конечный узел недостижим)
- N (сеть конечного узла недостижима)
- P (несовместимость протоколов)
- Q (замедление источника)
- M (пакет слишком велик для маршрутизации и не может быть фрагментирован)
- ? (неизвестный ответ)

Последняя выводимая строка – сводная информация. IOS известит вас об успешном выполнении ping и о том, через какое время был получен отклик (если получен).

Примечание

Хотя мы еще не затрагивали понятие «перехода» (hop), отметим, что ping имеет ограничение на количество переходов. Маршрутизаторы в сети часто называют переходами (говоря о количестве маршрутизаторов, через которое должен пройти элемент информации, чтобы достичь своего конечного назначения). Например, говорят, что пакет, который перед доставкой прошел через три маршрутизатора, преодолел три перехода.

Команда ping (в пользовательском режиме) проверяет не более девяти переходов. Поэтому для любого проверяемого вами адреса, который находится от вас на

расстоянии более девяти маршрутизаторов-переходов, будет выведен ответ «адрес назначения недостижим», даже если он полностью функционален.

Переходы еще будут обсуждаться далее в главах, связанных с протоколами маршрутизации.

Может показаться, что выводится масса полезной информации, но на самом деле это самое элементарное использование команды. Чтобы получить доступ к более мощной версии ping, войдите в привилегированный режим.

ping (привилегированный режим)

Cisco IOS содержит немного более мощную, чем стандартная, версию команды ping, доступную в привилегированном режиме. Версия привилегированного режима позволяет администраторам оценить несколько переменных, а не просто проверить, отвечает ли узел.

Сначала можно подумать, что версии ping пользовательского и привилегированного режимов совпадают. Если вы воспользуетесь справочной системой Cisco IOS, то увидите, что две команды принимают одни и те же параметры:

```
Router#ping ?
 WORD Ping destination address or hostname
 ip  IP echo
 tag  Tag encapsulated IP echo
```

(Пояснения параметров были приведены в предыдущем разделе.) Но выполнение ping пользовательского режима без параметров приводит к ошибке:

```
Router>ping
% Incomplete command.
```

А если попытаться выполнить без параметров ping привилегированного режима, то вы будете вовлечены в расширенный диалог ping. Приведем отрывок из такого диалога:

```
Router# ping
Protocol [ip]:
Target IP address: 10.16.5.152
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.16.5.152, timeout is 2 seconds:
!!!!!
Success rate is 100 percent, round-trip min/avg/max = 2/3/4 ms
```

Просматривая параметры строка за строкой, легче понять, какую информацию собирает ping в привилегированном режиме.

Первый вопрос, задаваемый диалогом: какой протокол вы хотите использовать для отправки ping-запроса. По умолчанию предлагается значение IP, на которое и следует согласиться. (Ранее уже говорилось, что ping Cisco может использовать и другие протоколы, но в данной книге об этом не будет рассказано.) После того как в ответ на первый вопрос будет принято значение по умолчанию, появится вторая строка диалога.

Теперь ping интересуется адресом пункта назначения. Это самый важный элемент диалога. Если вы введете только адрес пункта назначения и пропустите все остальные вопросы, команда ping все равно будет выполнена. (Очевидно, что для проведения проверки адрес необходим.)

Далее вам следует ввести число повторений, то есть указать количество эхо-пакетов, которые вы хотите отправить получателю. По умолчанию их будет пять, но бывают ситуации, когда хотелось бы, чтобы их было больше или меньше. Например, если проводится проверка на неисправность соединения или его отсутствие, то пяти пакетов может быть недостаточно для выявления проблемы. Поэтому для периодически возникающих проблем с линией, может быть, лучше отправить не пять, а больше пакетов. Посылая больше пакетов, вы сможете дольше «пинговать» нужный адрес, повышая свои шансы на столкновение с периодической проблемой.

Затем ping-диалог предлагает задать размер эхо-пакета. Это полезно в сетях с большой рабочей нагрузкой. Хотя и кажется, что 100 байтов – это совсем немного данных, но вы можете захотеть уменьшить размер пакета. Например, если вы проверяете чрезвычайно загруженную линию (вероятно, чтобы определить причину перегруженности), пропускной способности может не хватить для эффективной отсылки пяти 100-байтных эхо-пакетов. Поэтому вы можете решить использовать пакеты меньшего размера.

Следующая строка диалога связана с установкой времени жизни эхо-пакетов. Обычно нет причин изменять значение по умолчанию, равное 2 секундам. Так у маршрутизатора будет достаточно времени для того, чтобы определить, откликается устройство или нет.

Потом ping-диалог спрашивает, хотите ли вы разрешить расширенные команды. Ответ "yes" откроет новый диалог; если же вы ответите "no", то будет продолжен текущий диалог. В текущем диалоге будет предложено ввести диапазон изменения размеров.

Когда указан диапазон изменения размеров, ping меняет размер отправляемых эхо-пакетов. Это помогает определить, влияет ли размер пакета на возможность соединения ваших конечных систем. Обычно на выполнение ping воздействует размер дейтаграммы, принуждая все пакеты быть одинакового размера, но если указан данный параметр, то создаются пакеты разных размеров.

После этого команда ping выполняется. Если же вы разрешите расширенные команды, то увидите такие приглашения:

```
Extended commands [n]: y
Source address or interface: 10.153.16.4
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: Record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.153.16.5, timeout is 2 seconds:
Packet has IP options: Total option bytes= 39, padded length=40
```

Расширенные команды позволяют получить еще больший контроль над командой ping и ее выполнением.

Первый вопрос диалога расширенных команд касается адреса отправителя запросов отклика. Получив возможность указать адрес источника, вы сможете следить за тем, какой интерфейс маршрутизатора выполняет команду. Такая возможность особенно полезна, если у вас несколько интерфейсов и все они потенциально могут обращаться к одним и тем же адресатам. Задание адреса интерфейса отправителя поможет вам уменьшить количество проблем с соединениями. В сценариях с несколькими интерфейсами маршрутизатора, имеющими возможность контактировать с некоторым адресом пункта назначения, можно выполнять расширенную команду ping для каждого интерфейса. Так вы определите, какой путь к устройству создает проблемы.

Все остальные приглашения относятся к более сложным вещам. Они связаны со способом форматирования пакетов, то есть спрашивается, какая структура будет использоваться в отдельных частях пакета. Это очень специфические параметры, но о них стоит поговорить.

Бит DF в заголовке IP указывает, что пакет не подлежит фрагментированию (don't fragment – DF). Маршрутизаторы и другие IP-устройства могут быть сконфигурированы так, чтобы перемещать только пакеты определенных размеров. Поэтому когда поступает пакет с размером, превышающим установленный, устройство просто фрагментирует его на более мелкие пакеты, которые оно может обрабатывать. Обычно это делается незаметно для пользователя. Устанавливая бит DF, вы сообщаете всем остальным устройствам сети: «Если вы сконфигурированы так, что можете перемещать только более мелкие, чем данный, пакеты, оставьте этот пакет и отправьте мне сообщение об ошибке». Такая возможность может быть полезна, если вы пытаетесь найти устройство, которое может быть ошибочно сконфигурировано для работы с маленькими пакетами, но обычно нет необходимости изменять эту настройку.

Два следующих параметра – это «validate replies» (подтверждать ответы) и «set data pattern» (определить структуру данных). Подтверждение ответов может быть избыточным, так как оно делает только одну вещь: создает ответ на ответ. Однако если у вас возникает периодическая проблема, этот параметр может быть полезен для определения того, правильные ли ответы вы получаете.

Определение структуры данных позволяет изменить существующую структуру битов эхо-пакета. Изменив структуру битов пакета, вы можете использовать анализатор пакетов для исследования структуры отправленного пакета в сравнении с полученным ответом. Это помогает при проверках на наличие помех в линиях и перекрестных наводок. Если структура битов ответа отличается от структуры отправленного пакета, значит, вы улавливаете какие-то электронные шумы.

Следующая строка выглядит немного загадочно. Требуется ввести один из перечисленных параметров:

```
Loose, Strict, Record, Timestamp, Verbose[none]:
```

Но непонятно, что означают эти параметры. «Loose», «Strict», «Timestamp» и «Verbose» – это те способы, которыми маршрутизатор может исследовать заголовок пакета (пояснения даны в табл. 10.1). Ответ по умолчанию – «none».

Примечание

Помните, что параметры `Loose`, `Strict`, `Timestamp`, `Record` и `Verbose` вводятся в поле «options» IP-заголовка. Нет гарантии, что каждый маршрутизатор сети сможет понять, как нужно обрабатывать эти команды. Это главным образом относится к не-Cisco-маршрутизаторам.

Таблица 10.1. Дополнительные параметры ping

Параметры	Описание
Loose source routing (свободное исполнение маршрута отправителя)	Указывает список устройств, которые должны участвовать в работе трассировщика. Другими словами, трассировщик может проводить пакеты через разные маршрутизаторы, при этом те, для которых указан параметр «loose», должны быть включены в путь
Strict source routing (строгое исполнение маршрута отправителя)	Аналогичен параметру «loose», но в путь трассировщика должны быть включены только указанные устройства
Record (запись маршрута)	Позволяет указать требуемое количество переходов
Timestamp (временные метки)	Позволяет указать требуемое количество временных меток

Параметры	Описание
Verbose (подробное описание)	Выбор любого из параметров (кроме «none») переводит пакет в режим «verbose». Результаты применения дополнительных параметров выводятся в пользовательском интерфейсе

Установив для пакета параметр «Record», вы будете получать эхо от каждого устройства, которое пакет встретит на своем пути к месту назначения. Если выбран этот параметр, выводится такое приглашение:

```
Number of hops [ 9 ]:
```

В указании количества переходов, которые вы хотите записать, проявляется отличие между трассировкой и записью маршрута. Тогда как ping ограничивается максимум девятью переходами, для трассировщика такого ограничения нет. Данный параметр позволяет указать количество переходов, не превосходящее 9 (значение по умолчанию).

Далее версия ping привилегированного режима выполняется точно так же, как и версия пользовательского режима. Команда ping хороша для определения доступности адресата, а для выявления проблем с путями необходима команда traceroute.

traceroute

Как вы уже знаете, команда ping успешно определяет, отвечает ли конечная система на запросы отклика ICMP. Но она не может объяснить, почему какой-то узел не отвечает.

Есть еще одна утилита, использующая ICMP-пакеты для проверки доступности конечного узла, – traceroute. Она посылает эхо-пакеты не только с адреса назначения, но и со всех устройств, которые встречаются на пути продвижения к конечной цели.

Команда traceroute, как и ping, имеет в Cisco IOS две версии: для пользовательского и для привилегированного режимов. Пользовательская версия имеет те же параметры, что и пользовательская версия ping (и работает аналогично). В пользовательской версии traceroute вы можете указать цель, путь к которой хотите отследить. Будучи выполненной, traceroute возвращает имена и адреса всех устройств, через которые она прошла, чтобы достичь указанного объекта:

```
Router>trace 10.16.4.153

Type escape sequence to abort.
Tracing the route to 10.16.4.153
1 Router.testnode.com (10.16.4.199) 62 msec 82 msec 78 msec
2 RouterB.testnode.com (10.16.4.189) 80 msec 99 msec 117 msec
3 RouterC.testnode.com (10.16.4.177)100 msec 110 msec 124 msec
```

Обратите внимание на то, что в конце каждого ответа стоит время, которое потребовалось на получение отклика от данного устройства. Как видите, по умолчанию traceroute посылает три эхо-пакета (в отличие от пяти для ping).

Что касается параметров выполнения программ, пользовательская версия traceroute мало отличается от пользовательской версии ping. Для обеих команд можно определить адрес пункта назначения и протокол, все остальное они делают сами. Если вы хорошо разобрались в пользовательской версии команды ping, то пользовательская версия traceroute не захватит вас врасплох, поэтому мы не будем останавливаться на ней и перейдем к версии привилегированного режима.

traceroute (привилегированный режим)

Как и в случае ping, все самые важные конфигурационные свойства traceroute доступны в привилегированном режиме. Так администраторы маршрутизаторов могут сохранить более полный контроль над сетью.

Выполнение команды traceroute в привилегированном режиме без указания адресата приведет к выводу такого диалога:

```
Router#trace
Protocol [ip]:
Target IP address: 10.16.4.153
Source address:
Numeric display [n]: n
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 10.16.4.153
```

Сразу бросается в глаза, что многие приглашения на ввод повторяют приглашения версии ping привилегированного режима. Так как их назначение аналогично параметрам ping, то Protocol, Target, Timeout, Probe count, Loose, Strict, Record, Timestamp и Verbose не будут рассматриваться в данном разделе. Чтобы вспомнить возможные значения и свойства этих параметров, вернитесь к разделу «ping (привилегированный режим)».

Давайте поговорим о тех параметрах, которые доступны только в версии traceroute привилегированного режима. Первый из них (хотя может показаться, что этот параметр тоже подобен своему аналогу в ping) – это адрес источника.

В отличие от ping, в параметре адреса источника нельзя указать имя интерфейса. Адрес источника должен быть протокольным адресом,

назначенным физическому интерфейсу. Не считая этого отличия, в остальном параметр адреса источника используется так же, как и в команде ping.

Два следующих параметра, «Minimum Time to Live» (минимальное время жизни) и «Maximum Time to Live» (максимальное время жизни) связаны со временем жизни пакета. Так как при использовании traceroute мы имеем дело с многочисленными устройствами, то время задержки может меняться от одного к другому. Поэтому вместо установки статического TTL для всех устройств traceroute создает для времени жизни пороговые значения.

Утилита traceroute может охватывать больше переходов, чем ping, поэтому ей необходимо больше времени для того, чтобы пакеты достигли конечного адресата. В результате двухсекундный тайм-аут, используемый ping, может оказаться недостаточным для traceroute. Устанавливая минимальное и максимальное значения, traceroute делает поправку на задержки между переходами. По умолчанию пороги равны 1 и 30 секундам. Этого времени должно хватить для компенсации незначительных сетевых задержек.

Последний параметр traceroute – это номер порта. Утилита traceroute позволяет пользователю указать, с какого IP-порта (протокольного порта) он хотел бы отправлять traceroute-пакеты. Это может быть очень важно для диагностики возможных проблем с безопасностью. Указывая порты для отправки traceroute-пакетов, вы можете определить, открыты ли порты, которые должны быть открыты, и, наоборот, закрыты ли порты, которые не должны быть открыты. Так как многие сети работают по незащищенным протокольным портам, отсылая traceroute-пакеты с разных портов, вы сможете уточнить архитектуру вашей сети.

Эти простые команды (ping и traceroute) очень помогают в диагностике и поиске неисправностей при маршрутизации. Не жалейте времени на изучение этих команд, их параметров и форматов вывода. Зная, как работают ping и traceroute, вы сможете значительно сократить время, необходимое для проверки и создания новых маршрутов и схем сети.

Теперь, когда мы поговорили об «ICMP-родственниках» IP, пришло время познакомиться с «родной» утилитой IP – Telnet. Даже если вы уже работали с IP, материал следующего раздела будет полезен для вас.

Telnet

Имея некоторый опыт работы с ПК и сетями, вы, наверное, уже использовали Telnet (или хотя бы слышали о нем). Telnet – это мощное средство для регистрации на удаленных устройствах и выполнения на них команд, особенно это касается работы в рамках операционной системы UNIX.

Маршрутизаторы Cisco могут использовать возможности Telnet двумя способами. Во-первых, Telnet позволяет администраторам получить удаленный доступ к маршрутизаторам (командным интерпретаторам и конфигурационным файлам). У администратора пропадает необходимость физически находиться у консолей маршрутизаторов для работы с ними. Наиболее распространенная форма администрирования маршрутизатора – это доступ к одному или нескольким устройствам по Telnet с персонального компьютера.

Вторая возможность (чем-то напоминающая первую) использования Telnet заключается в доступе администраторов, работающих с консолью маршрутизатора, к другим маршрутизаторам Cisco. То есть сами маршрутизаторы Cisco способны использовать Telnet-клиент для удаленного администрирования других маршрутизаторов Cisco. Это означает, что инженер Cisco может войти в систему на одном маршрутизаторе, а затем, используя командную строку, зайти на другие маршрутизаторы Cisco.

Однако эта невероятно полезная возможность представляет собой одну из самых больших проблем Cisco в плане обеспечения безопасности. Во многом так же, как римская система дорог была обращена захватчиками против римлян («все дороги ведут в Рим»), сетевые злоумышленники, получившие доступ к одному маршрутизатору, могут легко переместиться и на другие, используя Telnet. Поэтому будьте внимательны при защите своих соединений. Позже мы обсудим основы безопасности Cisco.

Удаленное администрирование с использованием Telnet

Для работы Telnet требуется некоторое дополнительное конфигурирование. То есть если вы хотите удаленно администрировать маршрутизатор по Telnet, необходимо настроить его соответствующим образом. Если же вы хотите просто использовать маршрутизатор в качестве Telnet-клиента (соединяться с него с другими маршрутизаторами и администрировать их), в дополнительной настройке нет необходимости.

По умолчанию сервисы Telnet-сервера на маршрутизаторах Cisco запрещены. Это сделано для предотвращения использования маршрутизаторов в то время, когда администраторы и не подозревают об активности протокола. Если вы хотите включить Telnet на маршрутизаторе, разрешив Telnet-клиентам удаленно администрировать маршрутизатор, вам необходимо выполнить ряд операций.

До этого наши попытки конфигурирования относились к интерфейсам (Ethernet0 или Ethernet1). Telnet же необходимо настраивать как линию передачи данных. Используйте команду `line` для конфигурирования линии (не забудьте указать, какую линию или линии вы хотите

сконфигурировать; введя `line 0 5`, вы одновременно будете настраивать линии с 0 по 5).

Войдя в режим конфигурирования линии, запросите справку по Telnet, чтобы посмотреть, какие параметры конфигурирования Telnet можно задать на вашем маршрутизаторе:

```
Router#configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
Router(config)#line 0 5
Router(config-line)#telnet ?
  break-on-ip          Send break signal when interrupt is received
  ip-on-break          Send interrupt signal when break is received
  refuse-negotiations  Suppress negotiations of Telnet Remote Echo and
SuppressGo Ahead options
  speed                Specify line speeds
  sync-on-break        Send a Telnet Synchronize signal after receiving a
Telnet Break signal
  transparent          Send a CR as a CR followed by a NULL instead of a CR
followed by a LF
```

Для каждого параметра даны пояснения. Но вам в настоящее время нужен один параметр – `speed`. Параметр `speed` должен подготовить ваш маршрутизатор к принятию удаленных соединений.

Примечание

В этом примере использована команда `line 0 5`, которая настраивает все доступные линии передачи данных на прием Telnet-соединений. Вы можете сделать еще один шаг и установить для каждой из линий отдельный пароль, но это может быть и не совсем продуктивно.

При обращении к маршрутизатору Cisco по Telnet нет возможности указать терминальную линию, которую вы хотели бы использовать. Поэтому если вы установите шесть разных паролей для шести линий, то может оказаться, что вам придется перепробовать их все, прежде чем получить доступ к системе. На это накладывается еще тот факт, что Cisco IOS предоставляет только три попытки для ввода правильного пароля, а затем блокирует соединение.

Чтобы завершить конфигурирование Telnet, введите следующую команду в режиме конфигурирования линии:

```
Router(config-line)#telnet speed 9600
```

Эта команда разрешит (в нашем случае) линиям 0-5 принимать Telnet-соединения на скорости 9600. Теперь вы можете как с удаленного ПК, так и с другого маршрутизатора, установить соединение по Telnet с любой из этих линий.

После успешной настройки маршрутизатора, вы можете захотеть проверить возможности Telnet. Для этого сначала убедитесь, что хотя бы один интерфейс маршрутизатора подключен к сети. Затем попытайтесь подключиться к маршрутизатору с Telnet-клиента.

Для этого попробуйте с Telnet-клиента подключиться к одному из установленных адресов маршрутизатора. Вы должны увидеть такое сообщение:

```
C:\Telnet 10.16.4.153

Line 0
Password required but none set

Disconnecting
```

Как видите, недостаточно просто разрешить линиям работу с Telnet; Cisco не разрешает использовать незащищенное Telnet-соединение. Поэтому последний шаг в настройке Telnet заключается в установке пароля для тех линий, которым разрешено принимать соединения.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#line 0 5
Router(config-line)#password telnet
Router(config-line)#^Z
```

Снова попробуем подключиться к маршрутизатору. Вас должно встретить приглашение на вход в систему:

```
C:\telnet 10.16.4.153

login

User Access Verification

Password:  telnet
Router>
```

Маршрутизатор успешно сконфигурирован для принятия входящих Telnet-соединений. Давайте перейдем к рассмотрению использования вашего маршрутизатора в качестве Telnet-клиента.

Использование маршрутизатора в качестве Telnet-клиента не требует дополнительного конфигурирования (не надо настраивать маршрутизатор на прием Telnet-соединений). Доступ по Telnet с одного маршрутизатора Cisco на другой очень прост, он выполняется посредством одной из трех команд. Для открытия сеансов связи между маршрутизаторами (в административных целях) можно использовать команды `telnet`, `connect` и `rlogin`.

Примечание

Помните, что все три команды `telnet`, `connect` и `rlogin` – это стандартные команды Unix. Поэтому их использование в рамках Cisco не ограничивается соединением с маршрутизаторами. Администраторы могут с маршрутизатора Cisco обращаться по Telnet к Unix-серверу (или почти любым другим серверам, совместимым с Telnet). Но в данной книге использование Telnet будет рассмотрено только в связи с маршрутизаторами Cisco.

Команды `telnet` и `connect` работают одинаково и являются взаимозаменяемыми. Чтобы инициировать сеанс Telnet с маршрутизатора `Router` на другой маршрутизатор `RouterB`, используйте такой формат команды:

```
Router>telnet RouterB
Translating "RouterB"...domain server (10.16.4.188) [OK]
Trying Server3-RouterB.STYSCisco.com (10.16.4.188)... Open

login: userforRouter
Password:
RouterB)
```

Ввод команды `connect` приводит к такому же результату. Обе команды доводят вас до приглашения на вход в систему на нужном сервере. Если вы знакомы с командой Telnet в Unix, знайте, что она отличается от Cisco Telnet. Cisco Telnet не поддерживает доверительные соединения. То есть Cisco Telnet не может аутентифицировать Telnet-соединение, основанное на данных авторизации текущего соединения. Поэтому каждый инициированный сеанс Telnet выводит приглашение на регистрацию.

Установить Telnet-соединение можно и вообще без использования команд. Например, если ввести в командной строке несуществующую команду, то в ответ будет выведено такое сообщение:

```
Router>RouterB
Translating "RouterB"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address
```

Такое сообщение об ошибке генерируется Cisco IOS при попытке установить Telnet-соединение с указанным сервером (не осознавая, что вы хотели выполнить команду). В данном случае не существовало Telnet-сервера с именем `RouterB` (как и команды `RouterB`), поэтому было сгенерировано сообщение об ошибке. Если бы Telnet-сервер с именем `RouterB` существовал, вы увидели бы следующее сообщение:

```
Router>RouterB
Translating "RouterB"...domain server (10.16.4.188) [OK]
Trying Server3-RouterB.STYSCisco.com (10.16.4.188)... Open

login:
```

Необходимо сделать одно замечание: если имя Telnet-сервера совпадает с названием команды Cisco, то для обращения к такому серверу необходимо предварить его имя ключевым словом `telnet` или `connect`. Например, чтобы подключиться к Telnet-серверу `Enable`, нужно использовать такое обозначение:

```
Router> connect enable
Translating "enable"...domain server (10.16.4.189) [OK]
Trying Server3-enable.STYSCisco.com (10.16.4.189)... Open
```

После того как Telnet-соединение успешно установлено, вы увидите новое приглашение на ввод команды, которое состоит из имени удаленного сервера и правой скобки:

```
RouterB)
```

Такое приглашение показывает, что вы выдаете команды удаленному устройству. Каждый удаленный маршрутизатор, с которым вы установите соединение, будет иметь свое уникальное приглашение на ввод. Но большое количество приглашений на ввод может сбивать с толку, а переключение с одного на другое может быть затруднительным. Поэтому Cisco создала команду `resume`. Чтобы перейти от одного сеанса к другому, используйте в любой командной строке синтаксис `resume <connection name>`:

```
Router>resume RouterB
RouterB)
```

Команда `resume` переместит вас в указанную сессию. Чтобы завершить сессию, используйте команду `exit`:

```
RouterB)exit
Router>
```

Есть еще две команды, которые могут оказаться очень полезными для тех, кто планирует использовать несколько Telnet-соединений. Это команды `notify` и `refuse-message`.

Команда `notify` (после того как она будет разрешена) будет извещать вас о том, что выходные данные удаленного сеанса (отличного от того, в котором вы работаете в данный момент) требуют вашего внимания. В приведенном ниже примере показано, как разрешить использование свойства `notify`:

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#line 0 5
Router(config-line)#notify
Router(config-line)#^Z
Router#
```

Команда `refuse-message` будет уведомлять всех, пытающихся установить Telnet-соединение с маршрутизатором, о том, что линия уже используется:

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#line 0 5
Router(config-line)#refuse-message ! line in use !
Router(config-line)#^Z
Router#
```

Команда refuse-message имеет такие же параметры, как и большинство баннерных сообщений.

Команды telnet и connect могут использоваться для доступа к удаленным маршрутизаторам (и другим Telnet-серверам). Для этого же можно использовать и еще одну команду, имеющую ограниченное (специализированное) применение: rlogin.

rlogin

Команда rlogin работает так же, как telnet и connect. Основное ее отличие заключается в том, что не все устройства, поддерживающие Telnet, поддерживают и rlogin. Изначально rlogin создавалась для BSD Unix. Поэтому, если вы хотите инициировать сеанс rlogin, то сервер, принимающий соединение, должен быть сконфигурирован для работы с rlogin.

Синтаксис команды очень напоминает синтаксис telnet:

```
Router>rlogin RouterB
```

Преимуществом rlogin является то, что она поддерживает возможность отладки вывода. Указав, что соединение должно быть установлено в режиме отладки, вы получите более полный контроль над потоком вывода rlogin.

```
Router>rlogin RouterB debug
```

Использование таких команд, как ping, traceroute, Telnet, connect и rlogin, предоставляет вам возможность управления вашей средой маршрутизации. Если вы освоите эти команды, то будете готовы к восприятию более сложных свойств, с которыми еще предстоит познакомиться.

Резюме

- Поддержка IP по умолчанию устанавливается на маршрутизаторах Cisco в каждом из функциональных пакетов Cisco IOS, кроме IPX Basic Feature Pack.
- IP-адреса устанавливаются в режиме конфигурирования интерфейсов привилегированного режима работы командного интерпретатора.
- Чтобы войти в режим конфигурирования интерфейсов, используйте команду interface.
- Статические IP-пути записываются в файл startup-config при помощи параметра perm.
- В состав стека протоколов TCP/IP входит несколько утилит ICMP.

- Команда `ping` проверяет существование объекта, отправляя ему запросы отклика ICMP.
- Команда `traceroute` работает аналогично `ping`. Но `traceroute` посылает эхо-пакеты всем устройствам на пути ее следования к пункту назначения.
- Еще одной важной утилитой IP является Telnet.
- Маршрутизаторы Cisco могут устанавливать несколько Telnet-соединений и как клиенты, и как серверы.

Вопросы и ответы

Вопрос Поддержка IP по умолчанию предоставляется на всех маршрутизаторах Cisco. Можно ли ее отменить?

Ответ Нет, поддержку IP отменить невозможно. Если вы не хотите работать с IP на вашем маршрутизаторе, просто не включайте его. (Или, если IP включен, используйте команду `no ip routing`.) Даже базовые функциональные пакеты IPX предоставляют поддержку IP.

Тест

Вопросы

1. Если вы хотите проверить путь между шестью конкретными маршрутизаторами, какой тип исполнения маршрута отправителя вы зададите для команды `ping`?
2. Сколько линий доступно для входящих Telnet-соединений?
3. Какой результат работы команды `ping` или `traceroute` обозначается символом `M`?

Ответы

1. Strict (строгий).
2. 6.
3. Используемый для проверки пакет слишком велик для маршрутизации.

Упражнения

1. Сконфигурируйте интерфейс Ethernet1 с IP-адресом 198.5.42.1 и маской подсети 255.0.0.0.

Решение

```
Router(configureconfig)#int e1
Router(configureconfig-interface)#ip address 198.5.42.1 255.0.0.0
```

2. Что означают приведенные ниже результаты работы команды `ping`?

```
1) Sending 5, 100-byte ICMP Echos to 10.16.5.152, timeout is 2 seconds:
. . . . .
2) Sending 5, 100-byte ICMP Echos to 10.16.5.152, timeout is 2 seconds:
! ! ! ! !
3) Sending 5, 100-byte ICMP Echos to 10.16.5.152, timeout is 2 seconds:
U U U U U
```

Ответ

1) Отклик ICMP от адресата не получен.

2) Команда выполнена успешно.

3) Узел назначения недостижим.

3. В чем заключается основное отличие между режимом записи маршрута для версий привилегированного режима команд `ping` и `traceroute`?

Ответ

Команда `ping` отслеживает не более девяти переходов (маршрутизаторов).

4. Какая ошибка сделана при попытке установления Telnet-соединения с маршрутизатором `show`?

```
Router>show
```

Ответ

Telnet-соединение не будет обработано, так как `show` – это команда Cisco IOS. Необходимо использовать одно из ключевых слов: `rlogin`, `connect` или `telnet`.

11

Введение в сегментированные сети

Данная глава введет вас в мир проектирования сетей. Работа с маршрутизаторами Cisco подразумевает не только их конфигурирование и грамотное сопровождение, но и выбор наиболее эффективного расположения маршрутизаторов в сети. Можно считать, что правильное размещение маршрутизатора в значительной степени определяет эффективность работы всего оборудования.

К этому моменту мы уже рассмотрели все основные принципы, лежащие в основе успешного выбора, конфигурирования и сопровождения простого маршрутизатора IP. Однако мир не ограничивается приведенными в предыдущих главах примерами. Реальные маршрутизируемые сети намного сложнее и запутаннее, чем те, с которыми мы успели познакомиться.

В оставшейся части этой книги мы рассмотрим технические решения, с которыми чаще всего приходится иметь дело специалистам по Cisco. Сегментация сетей – одно из таких решений. Сегментированные сети, в силу своей запутанности и сложности, могут стать камнем преткновения для многих профессионалов в области сетей и маршрутизации.

Имейте в виду, что хотя эта глава и посвящена маршрутизации IP между сегментированными сетями, в ней не затрагиваются вопросы, связанные с протоколами маршрутизации, так как их мы еще не изучали. Сейчас мы рассмотрим маршрутизацию только с точки зрения протокола IP. Понимание маршрутизации в целом требует знания как маршрутизируемых протоколов, так и протоколов маршрутизации, и мы, двигаясь последовательно, займемся сначала основами маршрутизации IP. В последующих главах мы рассмотрим различные протоколы маршрутизации.

Глубокое понимание предмета сегментированных сетей и маршрутизации в них потребует обсуждения таких тем, как:

- Определение потребности в сегментировании
- Конфигурирование статических маршрутов между подсетями

Освоение сегментированных сетей потребует от вас углубленного понимания протокола IP и его маршрутизации. До сих пор мы рассматривали процессы, происходящие в «плоской» среде, то есть в статичной и имеющей лишь один путь, соединяющий две сети. Если вам уже приходилось работать с компьютерными сетями, вы знаете, что большинство компаний, использующих маршрутизаторы Cisco, не подходят под это описание.

Сегментированная IP-среда на сегодняшний день является самым распространенным вариантом конфигурации локальных и глобальных сетей. Сегментирование IP-сети предполагает ее разделение на подсети, в результате чего образуются десятки небольших IP-сетей. Каждая из этих подсетей связана с главной сетью и с другими подсетями. Зачастую такое многообразие маршрутов и схем адресации сбивает с толку даже бывалых профессионалов.

В главах 9 и 10 мы вкратце рассмотрели процесс создания IP-подсети и получения ее маски в двоичном виде. В первом разделе этой главы данная тема рассмотрена более подробно, показана физическая сторона процесса сегментирования сети и представлены некоторые принципы, которые могут пригодиться при сегментировании вашей собственной сети.

Определение потребности в сегментировании

До сих пор наши рассуждения о сегментировании сводились к тому, что биты, позаимствованные у адреса устройства, передаются адресу сети. Хотя этот способ и увеличивает количество адресов сетей, он одновременно уменьшает количество устройств в каждой из них. В действительности существует много веских причин для разделения IP-сети на подсети. Одна из самых серьезных причин, способная побудить администратора к сегментированию сети, – это постоянная нехватка IP-адресов. Применяемая ныне схема адресации IP-4 ограничена и скоро будет исчерпана.

Примечание

Хотя IP-4 является наиболее широко используемой версией IP, на горизонте уже появилась IP-6. Версия IP-6 предлагает значительно больший диапазон адресов и способна обеспечить нужды компьютерных сетей в обозримом будущем.

С наступлением эры сетевых технологий статический пул IP-адресов быстро исчерпался. Сетевые администраторы, специалисты по марш-

рутизации и интернет-провайдеры вынуждены дорожить доставшимися им адресами.

Деление IP-сети на подсети

Когда сеть спроектирована и построена, на администратора ложится обязанность разработать действующую схему IP-адресации. Эта схема должна обеспечивать количество IP-адресов, адекватное потребностям вычислительной среды. Каждое устройство, которое потребует сетевого взаимодействия – персональные компьютеры, серверы, маршрутизаторы, – нуждается, по крайней мере, в одном адресе. Разработка схемы адресации требует способностей к планированию и предвидению.

Примечание

IP-схема представляет собой множество IP-адресов (как для сетей, так и для устройств), которые присваиваются и используются внутри единой рабочей среды. IP-схема может состоять из диапазона адресов одного класса, а может включать в себя несколько диапазонов адресов разных классов.

Например, на рис. 11.1 изображена планируемая сетевая среда. Для этой среды определены географическое расположение объектов и предполагаемые требования к штату сотрудников. Имея такую информацию, администратор может вычислить приблизительное количество IP-адресов, необходимых для удовлетворения имеющихся требований.

Анализируя рисунок, администратор видит, что необходимо 695 IP-адресов. К этому количеству компания хотела бы добавить еще адреса, которые могут быть зарезервированы для последующего расширения: приема новых сотрудников или покупки оборудования. В данном примере администратор и компания договорились о том, что IP-схема сети должна содержать как минимум 750 адресов узлов и один адрес сети.

Такому требованию удовлетворяет IP-лицензия класса B. IP-лицензии класса B позволяют разметить 65 534 узла. Может показаться, что это оружие избыточной мощности для сети, которой требуется всего 750 адресов, но следующий, более мелкий класс C поддерживает только 254 узла. Поэтому организация должна приобрести лицензию класса B и начать готовиться к созданию своей новой сети.

Примечание

За последние несколько лет очень возрос спрос на лицензии IP. В связи с этим сегодня вряд ли можно представить себе, что организации, состоящей из 750 человек, будет выдана лицензия на целый класс. По всей вероятности, компания из нашего примера получит подсеть интернет-провайдера, имеющего лицензию на класс B. Однако для наглядности процесса сегментирования будем считать, что компании выдана полная лицензия.

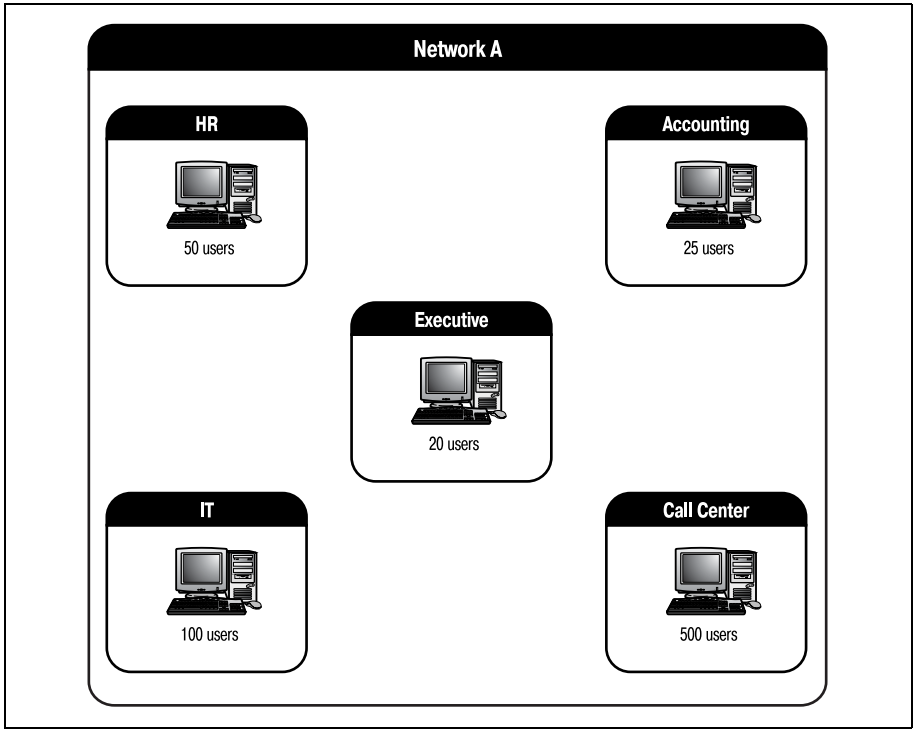

Рис. 11.1. *Планируемая сеть*

Пусть нашей фирме выдана лицензия класса B – 135.55.0.0, с маской подсети 255.255.0.0. Помните, что такая лицензия обеспечивает наличие одной сети (135.55) и 65 534 узлов (от 0.1 до 254.254). Администратор назначает новые адреса в сети. Получившаяся сетевая среда изображена на рис. 11.2.

На рис. 11.2 мы видим, что администратор рассматривал всю сеть как единый объект (каковым она технически и является) и присваивал адреса узлов, взяв за основу физическое расположение ПК. Компьютеры отдела HR получили адреса узлов с 1.1 по 1.51; департаменту IT присвоены адреса с 1.52 по 1.152, и т. д. Но это не самый эффективный способ распределения адресов.

Например, если у компании появится новый офис или она приобретет другую маленькую компанию, то им уже не останется сетевых адресов (используется всего один сетевой адрес – 135.55). Любым новым сетям, добавляемым в среду, должны присваиваться адреса, чтобы они могли участвовать в работе среды. Сетевому администратору необходимо каким-то способом создать новые адреса сетей, в то же время сохранив достаточное количество адресов узлов для назначения устройствам по всей сетевой среде. Другими словами, необходимо сегментировать лицензию класса B.

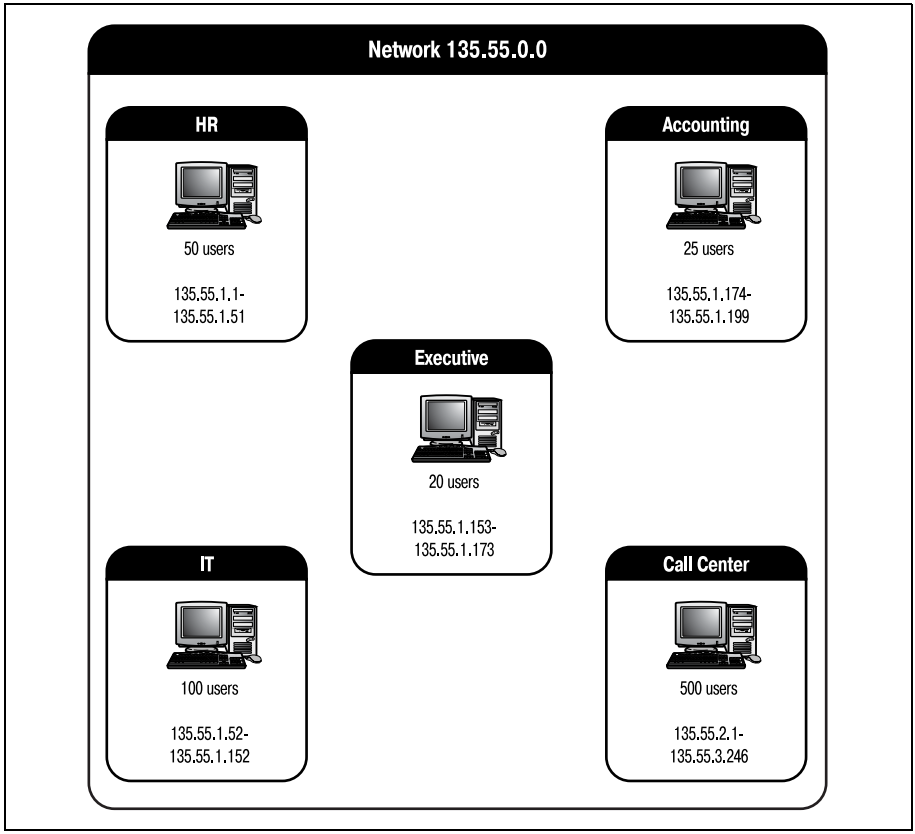

Рис. 11.2. *Сетевая среда с IP-схемой*

Сегментирование лицензии класса B разделяет ее на несколько сетей. Разделив один сетевой адрес (135.55) на несколько сетевых адресов, администратор получает возможность использовать один из них для текущей сети и сохранить остальные на будущее. Тогда сетевая среда будет наиболее масштабируемой.

Однако сегментирование сетевого IP-адреса имеет и свои недостатки. Получая дополнительные сети, вы теряете узлы. То есть, создавая больше адресов сетей, вы уменьшаете количество адресов узлов, которые могут быть назначены в каждой из сетей. Для многих организаций потребность в сетях перевешивает потребность в узлах каждой сети. Давайте посмотрим, как сегментировать данную сеть так, чтобы использовать IP-адреса с максимальной пользой. Затем мы обсудим маршрутизацию в новой среде.

Так как сегментирование увеличивает количество сетей в IP-схеме, администратор должен спрогнозировать, сколько сетей может понадобиться компании в будущем, и при этом еще сохранить необходимое

количество узлов. Администратор уже знает, что компании нужна как минимум одна сеть и 695 узлов.

Теперь следует решить, сколько бит IP-адреса должно быть передано из части, соответствующей адресу узла, сетевому адресу для обеспечения необходимого количества сетей. Формулы расчета количества сетей и узлов были приведены в главе 9. Количество сетей и узлов, возникающее в результате передачи каждого бита, представлено в табл. 11.1.

Таблица 11.1. Количество сетей, появляющихся в результате сегментирования

Кол-во бит	Двоичный адрес	Кол-во сетей	Узлов в сети
2	11000000.00000000	2	16382
3	11100000.00000000	6	8190
4	11110000.00000000	14	4094
5	11111000.00000000	30	2046
6	11111100.00000000	62	1022
7	11111110.00000000	126	510

Если администратор примет решение передать 6 бит адреса узла сетевому адресу, то он получит 62 возможные сети с 1022 узлами в каждой. Этого будет достаточно для того, чтобы всей среде были присвоены адреса, при этом она будет работать должным образом.

После того как администратор определил, сколько бит необходимо использовать для получения правильного количества сетей и узлов, эти адреса должны быть назначены сети. Первым шагом в присвоении новых адресов является определение сетевых адресов, которые будут использоваться, и соответствующей им маски подсети.

Для вычисления нового сетевого адреса требуется немного математики, мы уже говорили об этом в главе 9. Формула для вычисления сетевых адресов нашей сегментированной сети выглядит так:

256 – маска подсети = интервал между сетевыми адресами

Адрес первой сети в подсети – это просто интервал между сетевыми адресами. Все последующие адреса определяются последовательным добавлением интервала до тех пор, пока сумма не достигнет значения маски подсети, которое уже не годится для адреса. Звучит достаточно сложно, но после того как вы выполните приведенный выше пример, все станет понятно.

Первой переменной в уравнении для определения сетевых адресов является маска подсети. Соответственно, прежде чем вычислить сетевой адрес, вы должны определить, какой будет маска подсети для новой IP-схемы.

Примечание

Маска подсети остается неизменной в рамках IP-схемы. То есть несмотря на то что мы разделили IP-адрес между несколькими сетями, все эти сети будут совместно использовать общую маску подсети. Это будет та нить, которая свяжет сети вместе.

Чтобы вычислить маску подсети, просто запишите единицы (в двоичном формате) во все разряды сетевой части адреса. Так вы получите маску подсети, которая будет применяться во всей вашей сетевой среде. Различные маски подсети, доступные для адресов класса B, представлены в табл. 11.2.

Таблица 11.2. Маски подсети класса B

Количество бит	Двоичная маска	Маска подсети
1	11111111.11111111.10000000.00000000	255.255.128.0
2	11111111.11111111.11000000.00000000	255.255.192.0
3	11111111.11111111.11100000.00000000	255.255.224.0
4	11111111.11111111.11110000.00000000	255.255.240.0
5	11111111.11111111.11111000.00000000	255.255.248.0
6	11111111.11111111.11111100.00000000	255.255.252.0
7	11111111.11111111.11111110.00000000	255.255.254.0

Согласно таблице, для нашей сети подойдет маска 255.255.252.0 – маска для адреса класса B, в котором 6 бит адреса узла были переданы адресу сети. Если бы было решено передать 5 бит, то использовалась бы маска 255.255.248.0.

После того как маска сети определена, вставим ее значение в уравнение и вычислим значения сетевых адресов. Напоминаем, что уравнение для вычисления сетевых адресов нашей сегментированной среды выглядит следующим образом:

256 – маска подсети = интервал между сетевыми адресами = первый сетевой адрес

Определив маску подсети (255.255.252.0), подставляем ее значение в уравнение:

$256 - \mathbf{252} = 4$

Применив формулу, мы узнали, что первая сеть будет иметь адрес 135.55.4.0 с маской подсети 255.255.252.0. Используя этот сетевой адрес, можно приступить к перераспределению адресов в сети, рассматриваемой нами в качестве примера. Первые 1022 адреса сети 135.55.4.0 начинаются с 135.55.4.1 и заканчиваются 135.55.7.254. На рис. 11.3 изображена сеть, используемая нами в качестве образца, которой назначены адреса из нашей подсети.

Сравните сети на рис. 11.2 и 11.3. План, представленный на рис. 11.2, использовал только один сетевой адрес и терял до 64 тысяч адресов узлов, в то время как новый проект использует одну из 63 сетей и имеет приблизительно 300 резервных узлов. Такое решение гораздо более эффективно и «дружественно» маршрутизатору.

Теперь предположим, что после того как сеть построена, у компании появляется второй офис. Адресация в новой сети должна быть построена по той же IP-схеме, что и в первой. Кроме того, между двумя сетями должен быть помещен маршрутизатор Cisco, который свяжет их. Новая сеть представлена на рис. 11.4.

Вспомните, как был вычислен первый сетевой адрес нашей подсети:

256 – 252 = 4 (135.55.**4**.0)

Чтобы получить следующий сетевой адрес, добавьте к первому сетевому адресу приращение. Если взять приращение (4) и сложить его с первым сетевым адресом, получится второй сетевой адрес (135.55.**8**.0).

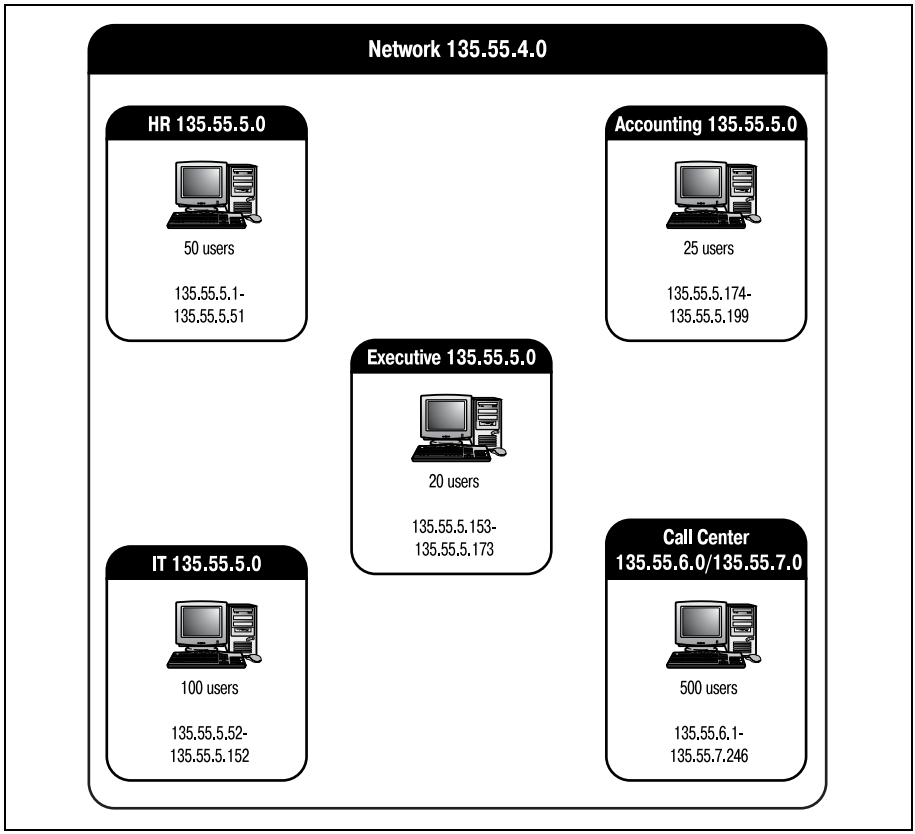

Рис. 11.3. *Пример сети, поделенной на подсети*

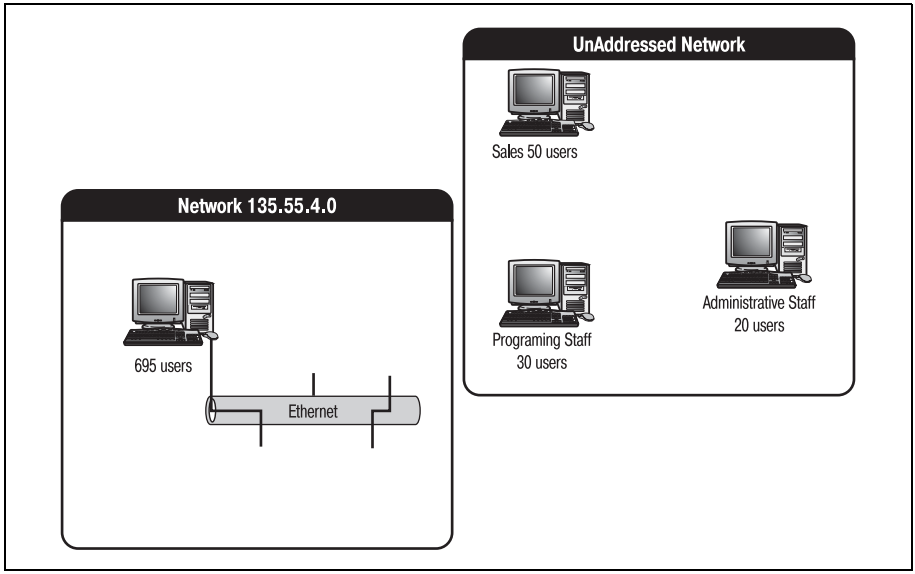

Рис. 11.4. *Второй сегмент сети*

Вторая сеть имеет точно такие же характеристики, как и первая. То есть в ней может быть 1022 узла, и она использует маску подсети 255.255.252.0.

Процедура добавления первого сетевого адреса для получения следующего сетевого адреса может повторяться до тех пор, пока сумма не станет равна адресу самой маски подсети, с которым сетевой адрес не может совпадать. Таким способом можно определить все доступные сети. Оставшиеся доступные сети нашей подсети класса B представлены в табл. 11.3.

Таблица 11.3. Оставшиеся сетевые адреса подсети

Маска подсети	Сеть подсети	Начало диапазона адресов узлов	Конец диапазона адресов узлов
255.255.252.0	135.55.4.0	135.55.4.1	135.55.7.254
	135.55.8.0	135.55.8.1	135.55.11.254
	135.55.12.0	135.55.12.1	135.55.15.254
	135.55.16.0	135.55.16.1	135.55.19.254
	135.55.20.0	135.55.20.1	135.55.23.254
	135.55.24.0	135.55.24.1	135.55.27.254
	135.55.28.0	135.55.28.1	135.55.31.254
	135.55.32.0	135.55.32.1	135.55.35.254
	135.55.36.0	135.55.36.1	135.55.39.254

Маска подсети	Сеть подсети	Начало диапазона адресов узлов	Конец диапазона адресов узлов
	135.55.40.0	135.55.40.1	135.55.43.254
	135.55.44.0	135.55.44.1	135.55.47.254
	135.55.48.0	135.55.48.1	135.55.51.254
	135.55.52.0	135.55.52.1	135.55.55.254
	135.55.56.0	135.55.56.1	135.55.59.254
	135.55.60.0	135.55.60.1	135.55.63.254
	135.55.64.0	135.55.64.1	135.55.67.254
	135.55.68.0	135.55.68.1	135.55.71.254
	135.55.72.0	135.55.72.1	135.55.75.254
	135.55.76.0	135.55.76.1	135.55.79.254
	135.55.80.0	135.55.80.1	135.55.83.254
	135.55.84.0	135.55.84.1	135.55.87.254
	135.55.88.0	135.55.88.1	135.55.91.254
	135.55.92.0	135.55.92.1	135.55.95.254
	135.55.96.0	135.55.96.1	135.55.99.254
	135.55.100.0	135.55.100.1	135.55.103.254
	135.55.104.0	135.55.104.1	135.55.107.254
	135.55.108.0	135.55.108.1	135.55.111.254
	135.55.112.0	135.55.112.1	135.55.115.254
	135.55.116.0	135.55.116.1	135.55.119.254
	135.55.120.0	135.55.120.1	135.55.123.254
	135.55.124.0	135.55.124.1	135.55.127.254
	135.55.128.0	135.55.128.1	135.55.131.254
	135.55.132.0	135.55.132.1	135.55.135.254
	135.55.136.0	135.55.136.1	135.55.139.254
	135.55.140.0	135.55.140.1	135.55.143.254
	135.55.144.0	135.55.144.1	135.55.147.254
	135.55.148.0	135.55.148.1	135.55.151.254
	135.55.152.0	135.55.152.1	135.55.155.254
	135.55.156.0	135.55.156.1	135.55.159.254
	135.55.160.0	135.55.160.1	135.55.163.254
	135.55.164.0	135.55.164.1	135.55.167.254
	135.55.168.0	135.55.168.1	135.55.171.254
	135.55.172.0	135.55.172.1	135.55.175.254

Таблица 11.3 (продолжение)

Маска подсети	Сеть подсети	Начало диапазона адресов узлов	Конец диапазона адресов узлов
	135.55.176.0	135.55.176.1	135.55.179.254
	135.55.180.0	135.55.180.1	135.55.183.254
	135.55.184.0	135.55.184.1	135.55.187.254
	135.55.188.0	135.55.188.1	135.55.191.254
	135.55.192.0	135.55.192.1	135.55.195.254
	135.55.196.0	135.55.196.1	135.55.199.254
	135.55.200.0	135.55.200.1	135.55.203.254
	135.55.204.0	135.55.204.1	135.55.207.254
	135.55.208.0	135.55.208.1	135.55.211.254
	135.55.212.0	135.55.212.1	135.55.215.254
	135.55.216.0	135.55.216.1	135.55.219.254
	135.55.220.0	135.55.220.1	135.55.223.254
	135.55.224.0	135.55.224.1	135.55.227.254
	135.55.228.0	135.55.228.1	135.55.231.254
	135.55.232.0	135.55.232.1	135.55.235.254
	135.55.236.0	135.55.236.1	135.55.239.254
	135.55.240.0	135.55.240.1	135.55.243.254
	135.55.244.0	135.55.244.1	135.55.247.254
	135.55.248.0	135.55.248.1	135.55.251.254

Хотя новой сети нашей среды может быть присвоен любой сетевой адрес из этого списка, мы остановимся на 135.55.8.0. Теперь можно распределить новые адреса сети 8.0. Повторяя процедуру, выполненную для первой сети, будем назначать новые адреса в соответствии с функциями подразделений. То есть отдел продаж (Sales) получит адреса с 135.55.8.1 по 135.55.8.101 и т. д. Полностью снабженная адресами сеть изображена на рис. 11.5.

Две сети нашей среды не имеют возможности общаться друг с другом. Так как адреса им были присвоены как двум разным сетям, они не могут обмениваться данными. Чтобы обеспечить перемещение данных, необходимо поместить маршрутизатор так, чтобы он мог обслуживать обе части единой среды.

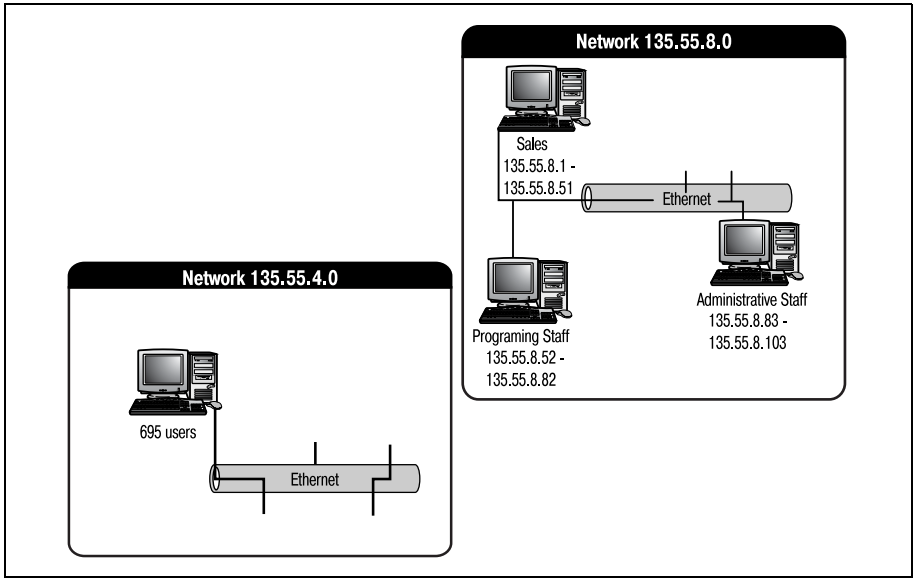

Рис. 11.5. *Полностью адресованная сегментированная среда, состоящая из двух сетей*

Размещение маршрутизаторов Cisco в сегментированных сетях

Многих сетевых проблем можно избежать за счет грамотного проектирования сети. Изучение маршрутизаторов Cisco побуждает к принятию на себя некоторой ответственности за всеобъемлющий план сети (включающий в себя и физическое оборудование для маршрутизации и программное обеспечение протоколов). Из данного раздела вы узнаете, как правильно размещать маршрутизаторы Cisco в сегментированных сетях.

Маршрутизатор, помещенный в сеть без учета окружающей его архитектуры, не будет работать эффективно. Необходимо уделить особое внимание таким элементам, как сетевой трафик и количество доступных интерфейсов. Помещая маршрутизатор Cisco между сегментированными сетями, необходимо учитывать несколько факторов:

- Сосчитайте доступные интерфейсы вашего маршрутизатора. Большая часть маршрутизаторов Cisco имеет два интерфейса LAN. Один маршрутизатор может соединить две сети, но для конфигурирования необходимых путей между тремя сетями может потребоваться до трех маршрутизаторов.
- Размещайте маршрутизатор так, чтобы наилучшим образом обслуживались самые загруженные части сети. Например, можно решить использовать отдельный маршрутизатор для наиболее загру-

женной части сети, даже если маршрутизация осуществляется только между двумя сетями.

- Если вы соединяете критически важные сетевые сегменты, разместите маршрутизаторы так, чтобы это привело к созданию избыточных соединений.
- Физически размещайте маршрутизаторы вместе с другим оборудованием, вблизи других устройств связи. Это упростит процедуру расширения, особенно если будет задействован Интернет.
- При размещении маршрутизаторов необходимо принимать во внимание то, какой именно протокол маршрутизации вы собираетесь использовать. Некоторые протоколы маршрутизации требуют, чтобы вы размещали маршрутизаторы в определенных местах или не выходили за рамки определенных ограничений. (Например, если вы планируете работать с RIP, то вы не сможете «дотянуться» до сетей, находящихся от вас на расстоянии, превышающем 16 переходов между маршрутизаторами.)

Примечание

Хотя необходимо всегда принимать во внимание используемый в сети протокол маршрутизации, но в данной главе мы откажем себе в этом удовольствии, так как ни один протокол маршрутизации еще не был изучен. Поэтому в этой главе при принятии решения о размещении маршрутизаторов ограничимся другими факторами. (О протоколах маршрутизации поговорим в оставшихся главах.)

Помня обо всех обстоятельствах, влияющих на выбор места, давайте выберем наилучшее расположение нашего маршрутизатора. Очевидно, что нужно поместить его между двумя сетями так, чтобы физически он был близок к ним обеим. Логичное размещение маршрутизатора изображено на рис. 11.6.

Работая с двумя сетями, можете при выборе места для вашего маршрутизатора положиться на логику. Помните, что временами работа с маршрутизаторами может быть достаточно сложной. Если возникает логичное решение, воспользуйтесь им. Так как в рассматриваемом сценарии присутствуют всего две сети, можно без сомнений расположить маршрутизатор Cisco между ними.

После того как место для маршрутизатора выбрано, остается сконфигурировать его. Давайте рассмотрим этапы настройки маршрутизатора Cisco, который должен соединить две сети.

1. Установить для каждого интерфейса адрес, соответствующий сети, к которой он подключен.
2. Включить интерфейсы.
3. Определить маску подсети.
4. Разрешить маршрутизацию IP.
5. Создать статические маршруты для соединения двух сетей.

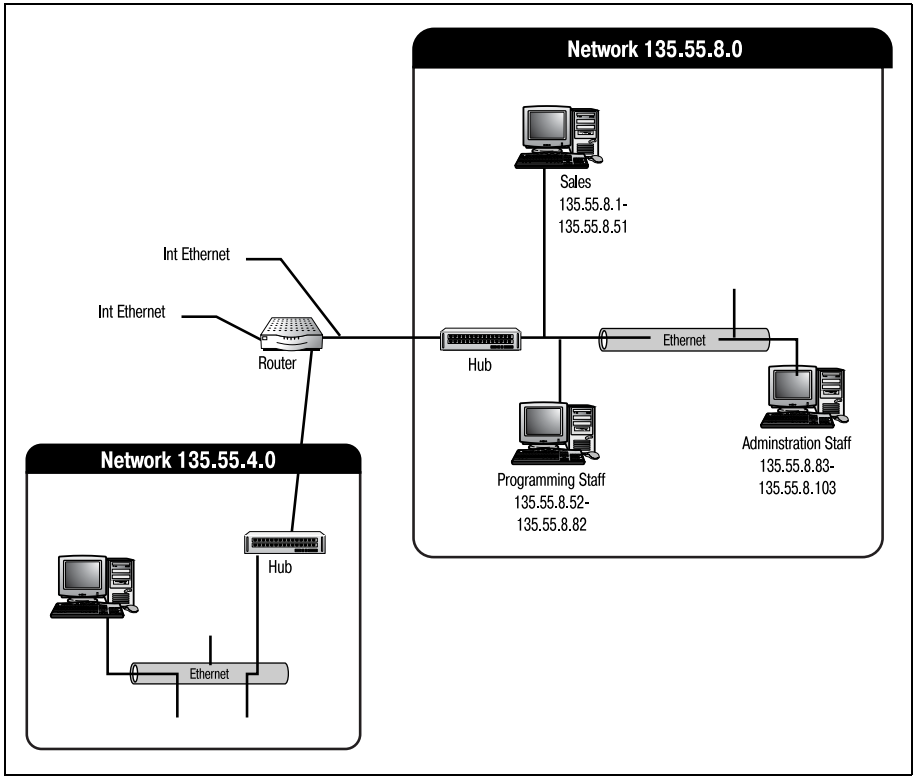

Рис. 11.6. *Маршрутизатор Cisco, помещенный между двумя сетями*

Первый шаг заключается в конфигурировании каждого интерфейса для работы с IP. Это означает, что каждому интерфейсу маршрутизатора нужно назначить адрес, соответствующий сети, с которой он связан. В той же самой последовательности команд можно определить маску подсети для каждого интерфейса и включить интерфейсы. Таким образом, три первых этапа из списка реализует одна серия команд, которая и представлена ниже (помните, что по умолчанию все интерфейсы имеют статус «shutdown», поэтому их необходимо включить вручную):

```
Router>enable
Router#configure terminal
Router(configure)#interface ethernet 0
Router(configure-interface)#ip address 135.55.4.1 255.255.252.0
Router(configure-interface)#no shutdown

Router#configure terminal
Router(configure)#interface ethernet 1
Router(configure-terminal)#ip address 135.55.8.1 255.255.252.0
Router(configure-interface)#no shutdown
Router(configure-terminal)#^Z
```

Давайте посмотрим, к какому из вышеупомянутых этапов конфигурирования относится каждая команда.

Первые три строки последовательности команд переводят маршрутизатор в режим конфигурирования интерфейсов. Чтобы попасть в этот режим, нужно сначала войти в привилегированный режим, а затем в режим глобального конфигурирования:

```
Router>enable
Router#configure terminal
Router(configure)#interface ethernet 0
```

Примечание

Цифра в конце команды `interface ethernet 0` указывает, какой именно интерфейс мы хотим конфигурировать.

Перейдя в режим конфигурирования интерфейсов, можно задать IP-адрес и маску подсети (четвертая строка) для нужного интерфейса. В данном случае интерфейс Ethernet 0 подключен к сети 135.55.4.0; поэтому устанавливаем для него адрес 135.55.4.1 (первый шаг в списке) и подсеть 255.255.252.0 (шаг 3):

```
Router(configure-interface)#ip address 135.55.4.1 255.255.252.0
```

Адрес и маска подсети определены, и пришло время включить интерфейсы (шаг 2 в списке). Следующая команда включает интерфейс и сохраняет конфигурацию в файле `running-config` (<Ctrl>+<Z> во второй строке обеспечивает выход из режима конфигурирования интерфейсов):

```
Router(configure-interface)#no shutdown
Router(configure-interface)#^Z
```

Обратите внимание на структуру команды. Вместо того чтобы сказать маршрутизатору, чтобы он включил интерфейс, мы указываем, что маршрутизатор должен не выключать его. Затем нажатием <Ctrl>+<Z> возвращаем маршрутизатор в привилегированный режим. Теперь можно повторить процесс для второго интерфейса:

```
Router#configure terminal
Router(configure)#interface ethernet 1
Router(configure-terminal)#ip address 135.55.8.1 255.255.252.0
Router(configure-interface)#no shutdown
Router(configure-terminal)#^Z
```

Примечание

Необязательно выходить из режима конфигурирования интерфейсов после выполнения каждой «порции» настроек. В данном примере <Ctrl>+<Z> было использовано для того, чтобы проиллюстрировать весь процесс целиком. Обычно оба интерфейса можно сконфигурировать в рамках одной сессии.

Шаг 4 нашего списка – это разрешение IP-маршрутизации. Фактически разрешается передача пакетов от одного интерфейса к другому. Прежде чем появится возможность использовать маршрутизатор между двумя подсетями, необходимо разрешить IP-маршрутизацию. Этот шаг выполняют две команды:

```
Router#configure terminal
Router(configure)# ip routing
```

Так как маршрутизация IP относится ко всему маршрутизатору, то команда выполняется в режиме глобального конфигурирования. По логике вы не можете разрешать IP-маршрутизацию для одного интерфейса и не разрешать для другого, ведь тогда данные все равно не будут никуда перемещаться, поэтому все параметры маршрутизации относятся к глобальному конфигурированию.

Последний этап – это конфигурирование статических маршрутов для направления потоков данных между подсетями. Так как наш проект не предусматривает использования протоколов маршрутизации, то нет возможности воспользоваться преимуществами динамической маршрутизации. Поэтому несмотря на то что маршрутизация происходит в пределах одного маршрутизатора (от одного интерфейса к другому), необходимо задать статические маршруты:

```
Router#configure terminal
Router (configure)#ip route 135.55.4.0 255.255.252.0 Ethernet 0 perm
Router (configure)#ip route 135.55.8.0 255.255.252.0 Ethernet 1 perm
```

Эти два выражения сообщают маршрутизатору, что сеть 135.55.4.0 подключена к интерфейсу Ethernet 0, а сеть 135.55.8.0 – к Ethernet 1. Но команда `ip route` обладает гораздо более мощными возможностями, чем использованные в данном примере. По мере изложения (особенно после изучения главы 13 «Введение в протоколы глобальных сетей») вы будете все более полно использовать `ip route`.

Конфигурирование статических маршрутов между подсетями

Маршрут – это карта или правило, используемые маршрутизатором для перемещения данных из одной сети в другую. Маршрутизаторы определяют движение информации в вашей сети. Маршрутизаторы Cisco могут использовать два разных типа маршрутов: статические и динамические. О динамических путях мы поговорим в следующих главах, а в этом разделе сосредоточимся на изучении статических путей. Говоря буквально, статические маршруты – это предопределенные пути для передачи данных из одной сети в другую, которые были жестко запрограммированы в памяти маршрутизатора.

Читая этот раздел, помните, что статические маршруты хороши только в некоторых случаях. Существует три основные сетевые ситуации, в которых оправдано использование статических IP-маршрутов:

- Маршрутизаторы сети не используют протокол маршрутизации.
- Правила безопасности требуют, чтобы определенные маршрутизаторы пропускали только определенный трафик.
- Среда маршрутизации не изменяется.

Примечание

О динамических маршрутах будет рассказано в последующих главах, так как их поддерживают только протоколы маршрутизации.

Команда конфигурирования статических маршрутов между подсетями на самом деле достаточно проста, хотя у нее есть ряд необязательных параметров (табл. 11.4), которые обеспечивают выполнение различных задач. Формат командной строки для команды `ip route` таков:

```
#ip route <Destination Network> <Destination Subnet> <Next Hop | Interface |
Null> <Next Hop | perm>
```

Таблица 11.4. Параметры команды ip route

Параметр	Описание
`<Destination Network>`	IP-адрес сети адресата (то есть путь, по которому должны отправляться данные)
`<Destination Subnet>`	Маска подсети сети адресата
`<Next Hop | Interface | Null>`	Параметр, указывающий, как маршрутизировать пакеты, направляемые адресату
`Next Hop`	IP-адрес маршрутизатора, которому должны пересылаться пакеты для сети адресата
`Interface`	Внутренний интерфейс, которому должны пересылаться пакеты для сети адресата
`Null`	Указание на то, что статический путь будет использоваться в другой команде (этот параметр чаще используется протоколами маршрутизации, такими как BGP)
`<Next Hop | perm>`	Необязательный параметр
`Next Hop`	IP-адрес маршрутизатора, которому должны отправляться пакеты, относящиеся к определеному
`Next Hop`	интерфейсу (необязателен, если в предыдущей позиции задан интерфейс)
`perm`	Указание маршрутизатору записать маршрут в файл `startup-config`, тем самым сделать маршрут постоянным

Давайте определим статические пути для более сложного случая сегментации сети, используя табл. 11.4 в качестве инструкции. На рис. 11.7 изображена часть большой сегментированной сети.

В этом сценарии у нас есть четыре взаимосвязанных маршрутизатора, каждый из которых обслуживает небольшой сегмент большой сегментированной сети. Чтобы обеспечить перемещение данных из сети 198.10.0.0 в сеть 198.13.0.0, необходимо задать следующие маршруты:

- Router 198.10.1.1:

```
RouterA(configure)#ip route 198.13.0.0 255.255.0.0 ethernet 1 198.11.1.1 perm
```

- Router 198.11.1.1:

```
RouterB(configure)#ip route 198.13.0.0 255.255.0.0 ethernet 1 198.12.1.1 perm
```

- Router 198.12.1.1:

```
RouterC(configure)#ip route 198.13.0.0. 255.255.0.0 ethernet 1 198.13.1.1 perm
```

- Router 198.13.1.1:

```
RouterD(configure)#ip route 198.13.0.0 255.255.0.0 ethernet 1 perm
```

Если эти четыре пути сконфигурированы, то информация будет пересылаться с одного маршрутизатора на другой, пока не достигнет адресата. Основу команды составляет информация о том, что данные для сети 198.13.0.0 должны покинуть маршрутизатор через определенный интерфейс и (в трех случаях) быть переданы другому маршрутизатору.

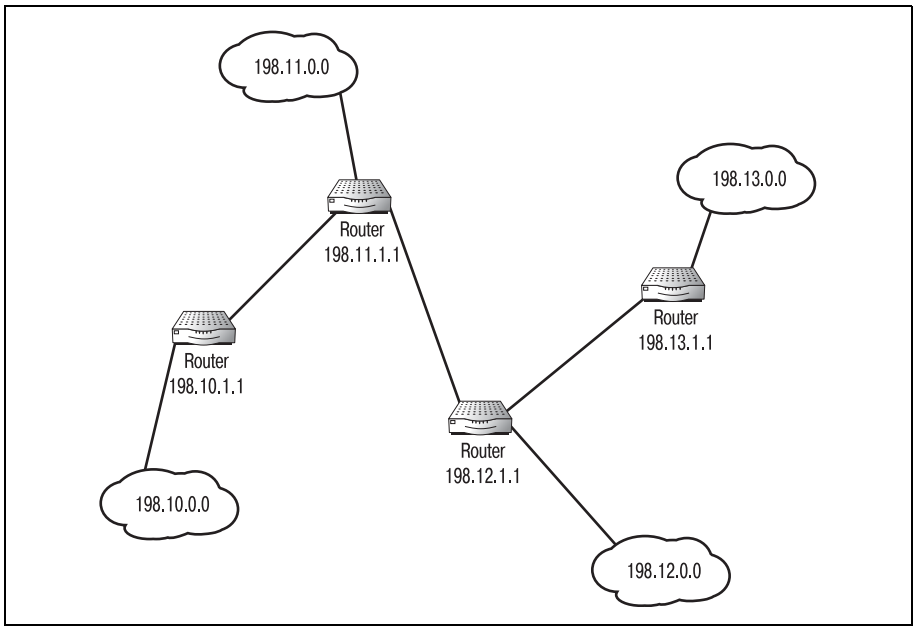

Рис. 11.7. *Сегмент сети*

Материал, представленный в этой главе, пригодится, когда вы дойдете до изучения таких протоколов маршрутизации, как RIP и OSPF. Для создания полнофункциональной среды маршрутизации мы будем комбинировать маршрутизируемые протоколы и протоколы маршрутизации. На одном маршрутизаторе можно использовать как динамические маршруты (посредством протоколов маршрутизации), так и статические IP-маршруты. Данная глава будет особенно полезна при работе с крупными действующими сетями.

Пройдя материал этой главы, вы должны были получить достаточно полное представление о маршрутизации и сегментировании IP. И маршрутизация и сегментирование будут использоваться далее в книге при изучении более сложных вопросов.

Резюме

- Одной из главных причин деления сетей на подсети является желание рационально использовать уменьшающийся пул доступных IP-адресов.
- Сегментирование производится за счет передачи битов из адреса узла сетевой части IP-адреса.
- Все сети сегментированной среды имеют общую маску подсети.
- Маршрутизаторы необходимо размещать только в тех сетях, которые совместимы с протоколом маршрутизации, который вы планируете использовать.
- Статическая маршрутизация должна применяться только в тех сетях, где не используются протоколы маршрутизации, где для обеспечения безопасности требуется, чтобы определенные данные отправлялись в определенные места, или где не меняется среда маршрутизации.

Вопросы и ответы

Вопрос Если сегментированный мною IP-адрес изначально принадлежал к классу B, почему я не могу использовать маску подсети 255.255.0.0?

Ответ Маска подсети 255.255.0.0 вмещает только одну сеть класса B. Чтобы пояснить это, посмотрим на двоичное представление маски. Два октета маски 11111111.11111111.00000000.00000000 зарезервированы под адреса узлов сети. После того как адрес сегментирован, некоторые из этих бит уже более не доступны. Поэтому использовать первоначальную маску больше нельзя.

Вопрос Почему для включения интерфейса используется команда `no shutdown`?

Ответ Чтобы сделать Cisco IOS не громоздкой и максимально простой в использовании, многие команды были исключены. Вместо того чтобы создавать две разные команды – одну для выполнения действия, а вторую для его отмены, – Cisco решила использовать одно ключевое слово для обозначения отрицания любой команды. Очевидным выбором в такой ситуации стало слово `no`. Чтобы выполнить действие, обратное любой команде (а не только `shutdown`), просто поставьте перед ней `no`.

Тест

Вопросы

1. Сколько сетей доступно в адресе класса B с маской подсети 255.255.224.0?
2. Сколько бит нужно передать сетевому адресу, чтобы вместить 2000 узлов?
3. Какой параметр статического маршрута используется для сохранения его в файле `startup-config`?
4. После того как для интерфейса установлен IP-адрес, какая команда может включить этот интерфейс?

Ответы

1. Шесть.
2. Пять.
3. `perm`.
4. `no shutdown`.

Упражнение

1. Сегментируйте адрес класса C 220.156.50.0 так, чтобы создать как минимум 20 сетевых адресов.

Решение

Передав пять бит из адреса узла в сетевой адрес, мы получим $2^5 - 2 = 30$ сетей с 6 узлами в каждой. Новая маска подсети выглядит так: 255.255.255.248.

12

Настройка протокола IPX

IPX (Internetwork Packet Exchange, межсетевой пакетный обмен) – это не так широко используемый, как IP, но не менее важный маршрутизируемый протокол. IPX используется в основном в сетях под управлением Novell NetWare и необходим каждому, кто планирует работать с операционной системой Novell. В этой главе вы познакомитесь с IPX как с протоколом и научитесь конфигурировать маршрутизатор Cisco для работы с IPX-адресами.

Представим основные темы данной главы:

- Введение в IPX
- Адресация IPX
- Маршрутизация IPX

К концу главы вы будете лучше представлять себе маршрутизируемый протокол IPX, чем значительно расширите свои знания по маршрутизации: ведь теперь вам будут знакомы два из наиболее популярных сегодня маршрутизируемых протоколов. Этим вы повысите свою квалификацию в области Cisco-технологий, что окажет вам неоценимую помощь во многих случаях.

Введение в IPX

До появления пятой версии Novell NetWare протоколом по умолчанию для сетей Novell был IPX (Internetwork Packet Exchange) – маршрутизируемый протокол, который идейно похож на IP. Оба протокола работают на сетевом уровне модели OSI и относятся к протоколам без установления соединений.

Компания Novell разработала IPX в начале 80-х годов. Он создавался как протокол маршрутизации для линии серверов Novell NetWare. IPX (и его аналог SPX, работающий на транспортном уровне) должен был стать запатентованным протоколом, который заменил бы TCP/IP в архитектуре NetWare.

Примечание

Novell разработала IPX на основе более раннего протокола Xerox, XNS.

У маршрутизаторов Cisco не возникает проблем при работе с IPX, так как он является протоколом третьего уровня без установления соединений. Маршрутизаторы Cisco легко и свободно маршрутизируют IP, IPX или оба эти протокола в одной и той же сети (и даже на одном маршрутизаторе).

В отличие от IP, IPX не используется маршрутизаторами Cisco по умолчанию. Поэтому чтобы использовать IPX-маршрутизацию, необходимо установить на вашем маршрутизаторе функциональный пакет IPX. На большинстве моделей для поддержки IPX устанавливают функциональный пакет IPX/IP Basic IOS. Этот пакет обеспечивает поддержку как IPX, так и IP, позволяя маршрутизатору работать с обоими протоколами.

Если вы никогда не работали с IPX и не знакомы с этим протоколом, то не обнаружите больших различий между IP и IPX. Наиболее очевидное отличие заключается в формате адреса IPX: любому, кто ранее не встречался с IPX, покажутся странными его адреса. Для того чтобы работать с IPX-адресами, вам придется перестроиться с разделенных точками целых чисел (адресов IP) на шестнадцатеричные числа.

Хотя IPX и не самый распространенный маршрутизируемый протокол, но он используется в достаточном количестве производственных сетей, что служит основанием для подробного рассмотрения этого протокола. Одной из наиболее распространенных конфигураций маршрутизаторов Cisco (включающих в себя IPX) является соединение сетей IPX с сетями IP. Например, если IPX-сети требуется выход в Интернет, IPX-пакеты нужно будет преобразовывать в IP-пакеты. Большинство маршрутизаторов Cisco могут обеспечить такую высокоуровневую протокольную маршрутизацию.

Оставшаяся часть главы посвящена особенностям маршрутизируемого протокола IPX и его конфигурированию на маршрутизаторах Cisco. Важный аспект архитектуры IPX – это применяемая им схема адресации. Начнем с рассмотрения формата адресов IPX и поговорим о том, как они назначаются.

Адресация IPX

IPX-адреса сходны с IP-адресами тем, что один адрес используется для распознавания и сети и узла. Но если IP-адреса имеют сетевую составляющую переменной длины (сетевая часть адреса может быть представлена одним, двумя или тремя октетами в зависимости от класса адреса), то IPX-адреса имеют фиксированную длину. IPX-адрес всегда придерживается структуры: *сеть.узел*.

Адрес IPX представляет собой 32-битный сетевой адрес, за которым следует 48-битный адрес узла. Весь IPX-адрес состоит из 80 бит (10 байт), то есть он значительно больше, чем 4-байтный IP-адрес. Больший размер адреса является преимуществом: значительно больше IPX-адресов доступно для назначения устройствам. Типичный IPX-адрес изображен на рис. 12.1.

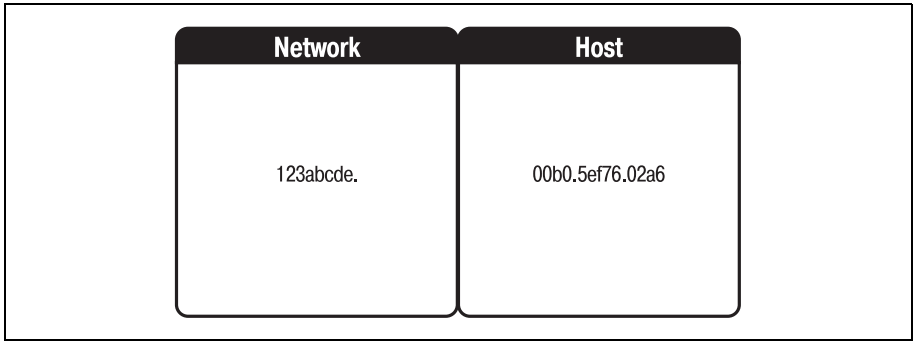

Рис. 12.1. *IPX-адрес*

Примечание

IPX-адреса всегда представляются в шестнадцатеричном формате.

Сетевая часть IPX-адреса – это значение, присваиваемое администратором. Сетевой администратор назначает уникальное число в качестве сетевого IPX-адреса для всего сетевого окружения. Присваиваемая величина должна быть записана восемью шестнадцатеричными цифрами (4 байта). Этот адрес должен оставаться уникальным для всей среды маршрутизации, чтобы не создавать путаницы при маршрутизации.

Если вы никогда не имели дела с шестнадцатеричными числами, вам может потребоваться какое-то время, чтобы к ним привыкнуть. Шестнадцатеричные числа относятся к системе счисления с основанием 16 (двоичные числа относятся к системе счисления с основанием 2 и поэтому используют только две цифры: 0 и 1). Шестнадцатеричные цифры перечислены в табл. 12.1.

Таблица 12.1. Шестнадцатеричные цифры

Шестнадцатеричная цифра	Двоичное число	Десятичное число
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Если администратор назначает сетевой адрес, длина которого меньше 8 шестнадцатеричных разрядов, маршрутизатор автоматически добавляет в начало адреса 0 до тех пор, пока он не достигнет нужного размера. Например, если администратор присваивает IPX-сети сетевой адрес 76b8, то маршрутизатор Cisco преобразует этот адрес в 0x000076b8.

Примечание

Условное обозначение 0x показывает, что значение записано в шестнадцатеричном формате. Если перед значением нет 0x, можете считать, что это не шестнадцатеричная запись. Например, «bad» – это английское слово, а «0xbad» – шестнадцатеричное число, соответствующее десятичному 2989.

Адрес узла не так произволен, как адрес сети. Особенностью IPX, отличающей его от таких протоколов, как IP, является динамическое определение адресов узлов. То есть если администратор не укажет иное, IPX динамически назначает адрес узла тому устройству, на котором он работает.

Адрес узла IPX-устройства обычно составляется из его MAC-адреса (MAC, media access control – протокол управления доступом к среде). Так как MAC-адреса по сути своей глобально уникальны, для любого узла всегда есть в наличии IPX-адрес. Поскольку протокол имеет доступ к сетевому адаптеру устройства, MAC-адрес может использоваться в качестве адреса узла.

Примечание

MAC-адрес – это число, присваиваемое каждому сетевому устройству его производителем. Центральный орган управления выдает каждому производителю сетевых устройств ряд номеров, которые он может присваивать своим продуктам. Эти адреса относятся к подуровню MAC канального уровня модели OSI. Хотя некоторые производители и разрешают замену MAC-адресов устройств, обычно они являются постоянными.

Уникальность и общедоступность MAC-адреса делает его особенно привлекательным для использования в качестве протокольного адреса. У каждого устройства есть такой адрес, и при этом MAC-адрес каждого устройства уникален. Поэтому, так как IPX применяет MAC-адрес для адреса узла, теоретически количество доступных для использования IPX-адресов не ограничено.

Конфигурирование IPX на маршрутизаторах Cisco

Прежде чем пытаться настроить IPX на вашем маршрутизаторе, необходимо понять, установлен ли на нем соответствующий функциональный пакет. Простейший из функциональных пакетов, который может иметься на маршрутизаторе и при этом предоставлять поддержку IPX, – это IPX/IP Basic. Один из способов узнать, какой функциональный пакет установлен на вашем маршрутизаторе Cisco, – использовать команду `show version`:

```
""Router#show version
Cisco Internetwork Operating System Software
IOS (tm) 1600 Software (C1600-NY-M), Version 12.0(3), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Mon 08-Feb-99 20:21 by phanguye
Image text-base: 0x02005000, data-base: 0x024A8D24

ROM: System Bootstrap, Version 12.0(3)T, RELEASE SOFTWARE (fc1)
ROM: 1600 Software (C1600-RBOOT-R), Version 12.0(3)T, RELEASE SOFTWARE (fc1)

Router uptime is 57 minutes
System restarted by reload
System image file is "flash:c1600-ny-mz.120-3.bin"
```

Эта команда не сообщает вам в явном виде имя установленного функционального пакета, но выводит имя образа IOS. Последняя строка приведенного выше вывода указывает, что образ IOS называется `c1600-ny-mc.120-3.bin`. Теперь вы можете обратиться к веб-сайту Cisco Connection Online (CCO) (в главе 3 «Введение в Cisco IOS» рассказывалось, что для получения доступа к сайту необходимо заключить сервисное соглашение), где увидите, что такое имя файла соответствует функциональному пакету IP/IPX Basic для маршрутизатора Cisco серии 1600, работающего под управлением версии 12.0.3 операционной системы IOS.

Однако вместо того, чтобы разыскивать эту информацию, вы могли просто попробовать настроить IPX на маршрутизаторе. Если соответствующий функциональный пакет не установлен, то маршрутизатор сообщил бы вам об этом при попытке конфигурирования.

Каким бы путем вы ни пошли, определение того, имеется ли на маршрутизаторе нужный функциональный пакет, должно быть вашей первоочередной задачей. Определив, что маршрутизатор может работать с IPX, вы можете начать конфигурирование. Использование IPX можно разрешить в режиме глобального конфигурирования.

Примечание

Многие из этих шагов аналогичны конфигурированию IP. Cisco умышленно сделала команды настройки протоколов схожими, чтобы сделать IOS максимально понятной.

Для того чтобы включить IPX-протокол, необходимо разрешить маршрутизацию IPX, что и делается в режиме глобального конфигурирования при помощи команды `ipx routing`. Если вы запросите справку Cisco IOS по команде `ipx routing`, то увидите, что она принимает один параметр. IOS позволяет указать адрес узла, который должен использоваться маршрутизатором:

```
Router(config)#ipx routing ?
  H.H.H  IPX address of this router
```

Если же вы введете команду без параметра, то IOS использует MAC-адрес маршрутизатора в качестве адреса узла. Для обеспечения уникальности не указывайте адрес узла вручную:

```
Router(config)#ipx routing
Router(config)#^Z
```

Команда разрешит маршрутизацию IPX на маршрутизаторе Cisco. (Помните, что отключить маршрутизацию IPX можно, используя команду `no`, например: `no ipx routing`.) На одном маршрутизаторе можно разрешить маршрутизацию и IP и IPX. Выполнение этой операции продемонстрировано ниже:

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ipx routing
Router(config)#ip routing
Router(config)#^Z
```

Причиной включения двух протоколов на одном маршрутизаторе может быть желание соединить две сети. Например, IPX не используется в Интернете, поэтому любая IPX-сеть, которой необходимо взаимодействие с Интернетом, должна быть подключена к маршрутизатору. Один интерфейс такого маршрутизатора должен быть настроен на работу с IPX, а второй – на работу с IP.

Маршрутизатор Cisco готов маршрутизировать IPX и любые другие протоколы, которые вы сконфигурировали. Но прежде чем он на самом деле сможет перемещать IPX-данные с одного интерфейса на другой, необходимо задать еще некоторые ключевые параметры:

- Номер сети
- Адреса интерфейсов
- Тип инкапсуляции

Все эти данные вводятся в программу конфигурирования интерфейсов `config-if`, поэтому для завершения настройки IPX необходимо войти в режим конфигурирования интерфейсов. Так как интерфейсы можно конфигурировать только по одному, начнем с Ethernet 0:

```
Router#configure terminal
Router(config)#interface ethernet0
Router(config-if)#
```

Все три порции информации из приведенного выше списка можно ввести, используя одну команду. Команда `ipx network` позволяет указать адрес сети, адрес узла и тип инкапсуляции для интерфейса. Команда имеет такой формат:

```
#ipx network <network number> encapsulation <encapsulation type>
```

Примечание

Обратите внимание на то, что в структуре команды отсутствует параметр определения адреса узла для интерфейса. Так как IPX динамически назначает адрес узла, используя в качестве основы MAC-адрес, нет необходимости указывать его. Однако до тех пор, пока команда `ipx network` не выполнена, адрес узла не присваивается. Команда `ipx network` сообщает маршрутизатору Cisco, что можно присваивать адрес узла устройству.

Первый параметр команды `ipx network` – это адрес сети. Этот номер должен быть установлен на всех IPX-устройствах внутри одной среды маршрутизации. Сетевой адрес указывает маршрутизатору, куда он должен (или не должен) направлять пакеты.

Длина сетевого адреса не может превышать 8 шестнадцатеричных разрядов. Но, как уже говорилось ранее, если вы укажете адрес, длина которого меньше 8 разрядов, то маршрутизатор добавит в его начало столько нулей, сколько необходимо для того, чтобы длина стала равна 8. В нашем примере указан адрес сети 1234. (Внутри маршрутизатора Cisco этот номер будет сохранен как 0x00001234.)

Сразу за сетевым адресом следует ключевое слово `encapsulation`. Оно используется для определения конкретного типа инкапсуляции, который будет применяться маршрутизатором для IPX-пакетов.

Тип инкапсуляции – это очень важная составляющая конфигурации IPX. В предыдущих главах рассказывалось о том, что когда порция

данных передается протоколу, он ее инкапсулирует. При инкапсуляции к данным добавляются поля, содержащие основную информацию об этих данных. Но в зависимости от используемого типа сети может потребоваться внести некоторые изменения в инкапсуляцию для того, чтобы данные маршрутизировались корректно.

Инкапсуляция IPX

Одним из ключевых элементов настройки IPX на маршрутизаторе Cisco является выбор метода инкапсуляции. Метод инкапсуляции определяет типы и порядок полей заголовка протокола. При выборе метода инкапсуляции для IPX необходимо учитывать ряд факторов. Во-первых, для всей среды маршрутизации должен быть выбран один и тот же метод инкапсуляции. То есть каждое устройство, соприкасающееся с IPX, должно использовать один и тот же метод инкапсуляции. Во-вторых, инкапсуляция должна соответствовать типу сети, например ethernet-инкапсуляция для ethernet-сетей. Параметры инкапсуляции IPX, доступные пользователям Cisco, приведены в табл. 12.2.

Таблица 12.2. Типы инкапсуляции IPX

Тип сети	Тип инкапсуляции
Ethernet	Ethernet II
	Ethernet 802.3
	Ethernet 802.2
	Ethernet 802.2 SNAP
FDDI	FDDI 802.2 LLC
	FDDI 802.2 LLC SNAP
	FDDI raw
Token Ring	Token Ring
	Token Ring SNAP

Примечание

Заметьте, что в табл. 12.2 методы инкапсуляции сгруппированы по типам сетей. Выбирая метод для вашей сети, учитывайте ее тип.

Примечание

Модель маршрутизатора, используемого в вашей сети, также должна соответствовать типу сети. Другими словами, если вы работаете с маршрутизатором Cisco 1605R (это маршрутизатор Ethernet), то не выбирайте метод инкапсуляции Token Ring.

Тип сети является основным фактором для выбора типа инкапсуляции IPX. Хотя для Ethernet существует четыре типа инкапсуляции, все они представляют различные версии сети. Выбранный тип инкап-

суляции определяет, какие поля будут присоединены к IPX-пакету (разные типы сетей могут читать разные поля). Чтобы пояснить ситуацию, рассмотрим поля для Ethernet-метода инкапсуляции.

Поля Ethernet-инкапсуляции

В процессе конфигурирования маршрутизации IPX на маршрутизаторе Cisco, инженеру необходимо выбрать метод инкапсуляции, соответствующий типу используемой сети. Существуют методы для таких сетей, как Ethernet, Token Ring и FDDI, но мы не будем подобно рассматривать их все. Рассмотрим инкапсуляцию на примере полей, сопоставленных различным версиям сетей Ethernet.

Самым распространенным типом инкапсуляции Ethernet для IPX является Ethernet_II. Ethernet версии II – это последнее воплощение широко используемого сетевого стандарта. Если вы конфигурируете IPX для Ethernet-среды, то, вероятнее всего, будете использовать инкапсуляцию Ethernet_II. Нижеследующий листинг приводит ключевые слова Cisco, соответствующие различным метолам инкапсуляции Ethernet:

```
Router(config)#interface ethernet0
Router(config-if)#ipx network 1234 encapsulation ?
  arpa          IPX Ethernet_II
  hdlc          HDLC on serial links
  novell-ether  IPX Ethernet_802.3
  novell-fddi   IPX FDDI RAW
  sap           IEEE 802.2 on Ethernet, FDDI, Token Ring
  snap          IEEE 802.2 SNAP on Ethernet, Token Ring, and FDDI
```

Обратите внимание на то, что ключевое слово Cisco для инкапсуляции Ethernet_II – `arpa`. Чтобы настроить для данного интерфейса инкапсуляцию Ethernet_II, используйте такую команду:

```
Router(config-if)#ipx network 1234 encapsulation arpa
Router(config-if)#^Z
```

Команда задает для текущего интерфейса маршрутизатора Cisco адрес IPX-сети 0x00001234 и инкапсуляцию Ethernet_II. Выбор метода Ethernet_II добавляет в IPX-пакет некоторые поля, которые необходимы для маршрутизации пакета по сети указанного типа. На рис. 12.2 изображен пакет IPX с инкапсуляцией Ethernet_II.

Для сравнения на рис. 12.3 приведены поля заголовка, добавляемые к IPX-пакету при инкапсуляции Ethernet_SNAP.

Имея дело с типами инкапсуляции IPX, необходимо помнить о двух вещах. Во-первых, выбранный тип инкапсуляции не меняет структуру IPX-пакета. Вместо этого он добавляет поля в его начало. Когда устройство получает инкапсулированный IPX-пакет, оно просто «счищает» инкапсулированные поля, оставляя IPX-пакет нетронутым.

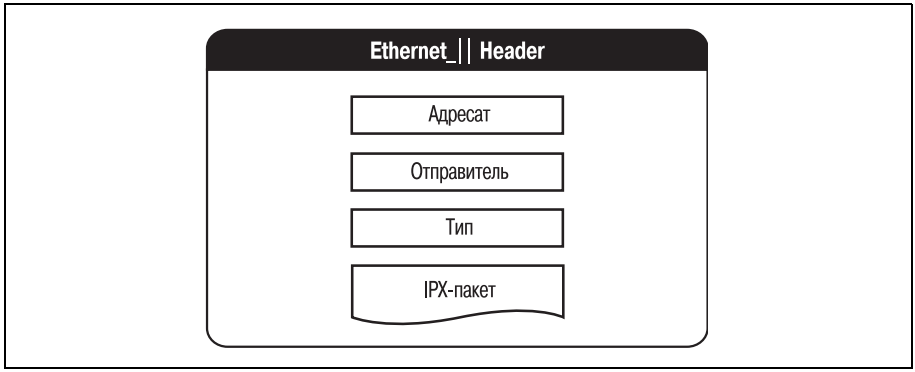

Рис. 12.2. *Поля заголовка Ethernet_II*

На рис. 12.4 изображен стандартный IPX-пакет. На рисунке представлены поля пакета до инкапсуляции.

Когда такой пакет отправляется, он инкапсулируется для используемого типа сети. На рис. 12.5 изображен тот же пакет, что и на рис. 12.4, но после инкапсуляции.

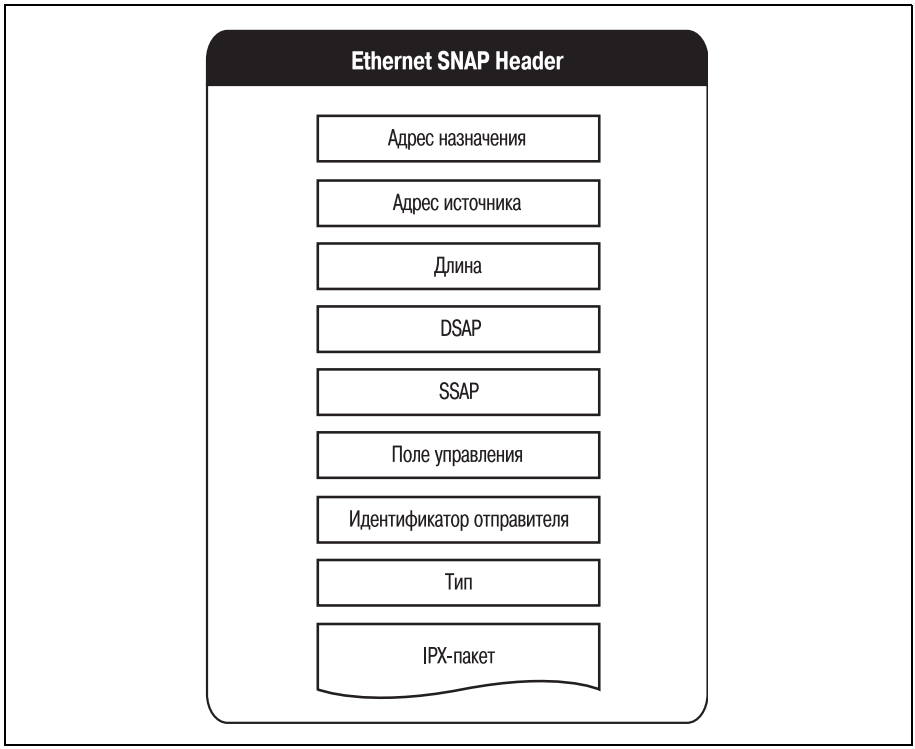

Рис. 12.3. *Поля заголовка Ethernet_SNAP*

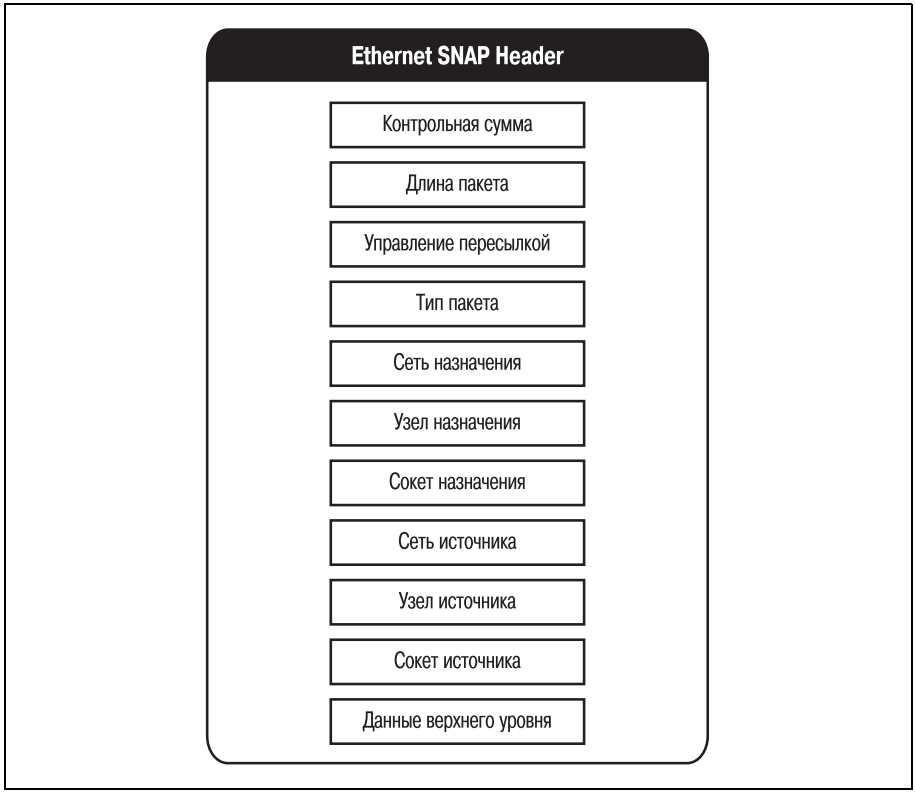

Рис. 12.4. *IPX-пакет до инкапсуляции*

Заметьте, что исходные поля IPX-пакета не изменены. Когда адресат получает пакет, он отделяет поля, добавленные при инкапсуляции. Остается исходный IPX-пакет.

Второе, о чем следует помнить при инкапсуляции, – для одного интерфейса можно определить различные типы инкапсуляции. Если маршрутизатор подключен к нескольким сетям, относящимся к различным типам, администратор может назначить соответствующие методы инкапсуляции. Поговорим об этом в следующем разделе.

Маршрутизация IPX

IPX-маршрутизация несколько сложнее, чем маршрутизация IP. Одна из причин этого в том, что в IPX-сетях может использоваться несколько типов инкапсуляции. Хотя для всех устройств сети должен быть определен один тип инкапсуляции, но один маршрутизатор Cisco может соединять множество сетей. Чтобы маршрутизатор Cisco мог успешно работать с любой средой IPX, может понадобиться настроить на нем несколько типов инкапсуляции IPX.

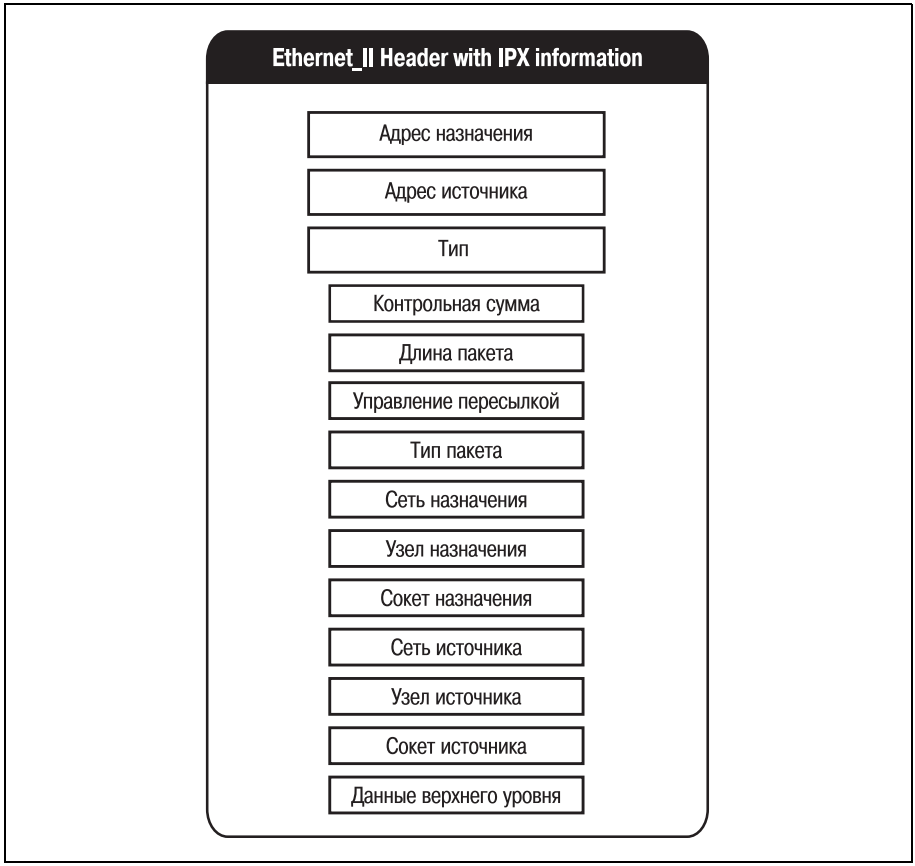

Рис. 12.5. *IPX-пакет после инкапсуляции*

Чтобы пояснить вышесказанное, рассмотрим два разных сценария. Первый – это простейшая среда IPX, состоящая из двух сетей Ethernet разных типов (рис. 12.6).

Первая сеть этой среды – стандартная Ethernet_II IPX-сеть. Можно предположить, что это более новая часть сетевой среды, которая была создана для того, чтобы расширить возможности второй сети на всю среду. Вторая сеть – это старая Novell Ethernet-сеть с архитектурой IEEE 802.3. Маршрутизатор Cisco был помещен между двумя сетями для того, чтобы позволить информации перемещаться из одной сети в другую.

Примечание

Две сети на рис. 12.6 не смогут взаимодействовать самостоятельно, так как они используют разные типы инкапсуляции. Для обеспечения взаимодействия необходимо промежуточное устройство.

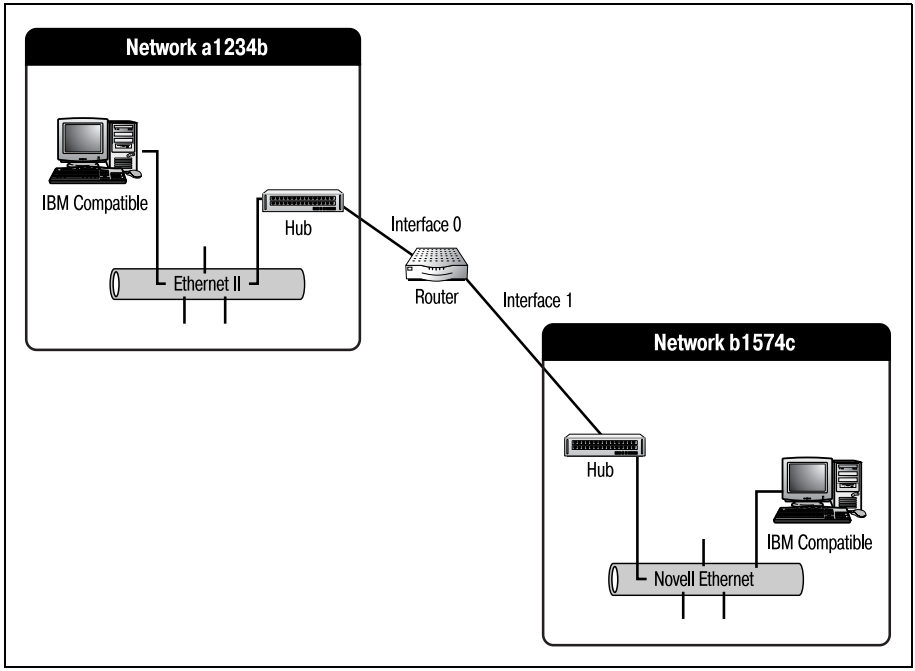

Рис. 12.6. *IPX-среда, состоящая из двух сетей*

Сконфигурируем маршрутизатор Cisco, изображенный на рис. 12.6. Сначала необходимо разрешить на устройстве маршрутизацию IPX. Затем двум интерфейсам присваиваем соответствующие номера сетей и типы инкапсуляции. Полный набор команд, необходимых для настройки маршрутизатора Cisco на работу в таком сценарии, выглядит так:

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ipx routing
Router(config)#^Z
Router#
1d00h: %SYS-5-CONFIG_I: Configured from console by console
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface ethernet 0
Router(config-if)#ipx network a1234b encapsulation arpa
Router(config-if)#no shutdown
Router(config-if)#interface ethernet 1
Router(config-if)#ipx network b1574c encapsulation novell-ether
Router(config-if)#no shutdown
Router(config-if)#^Z
Router#
1d00h: %SYS-5-CONFIG_I: Configured from console by console
Router#
```

Интерфейс ethernet 0 сконфигурирован с сетевым адресом 0x00a1234b и инкапсуляцией Ethernet_II, а для ethernet 1 определен адрес сети 0x00b1574c и инкапсуляция Ethernet 802.2. Теперь интерфейсы корректно настроены и могут перемещать пакеты из одной сети в другую. Но, как и в случае с IP, прежде чем маршрутизатор сможет пересылать данные, для интерфейсов необходимо определить статические маршруты.

Статические маршруты нужны для того, чтобы маршрутизатор знал, какие пакеты на какие интерфейсы и с каких интерфейсов можно пересылать. Используем команду ipx route для конфигурирования статических маршрутов, которые в нашем случае определят перемещение данных от интерфейса ethernet 0 к ethernet 1 и обратно.

Формат команды определения статических путей для IPX несколько отличается от такой же команды для IP. Структура команды и параметры статических маршрутов IPX приведены ниже:

```
#ipx route <network> <network-mask> <destination> | <tick> | <hop count> |
<float>
```

Хотя может показаться, что у команды ipx route какие-то непонятные параметры, но вы в них разберетесь. Назначение параметров команды поясняется в следующем списке:

- Параметр «network» (сеть) – это адрес источника для всех пакетов, предназначающихся для данного статического пути.
- Параметр «network mask» – сетевая маска (подобно «subnet mask» – маска подсети) используется для того, чтобы сообщить маршрутизатору, сколько бит содержится в сетевом адресе. Вам очень редко придется использовать что-либо, отличное от «FFFFFFFF».
- Параметр «destination» (адресат) – это IPX-адрес (адрес сети и узла) интерфейса, на который вы хотите пересылать пакеты.

Наряду с перечисленными выше необходимыми параметрами команда ipx route принимает также три необязательных параметра. Необязательные параметры позволяют управлять тем, как маршрутизатор использует заданные пути.

- Счетчик «tick» – это время действия пакета. Другими словами, если пакет не доставлен до того, как «tick» обратится в ноль, то пакет отбрасывается.
- Счетчик «hop count» указывает общее количество переходов через маршрутизаторы, которое пакет может проделать, прежде чем быть отвергнутым.
- Параметр «float» показывает, что маршрут (хотя он и статический) может быть перезаписан динамически вычисленным маршрутом.

Примечание

Тик (tick) – это общепринятое представление времени в мире маршрутизации. Тики (несложно догадаться, что они получили название от тиканья часов) не связаны

непосредственно с какой-то единицей времени. Они представляют собой цикл процессора маршрутизатора или (точнее) среднее количество времени, затрачиваемое процессором маршрутизатора на выполнение набора команд.

Разобравшись в параметрах команды `ipx route`, вы можете установить статический маршрут. Ниже приведены команды, определяющие два статических маршрута для нашего маршрутизатора:

```
Router(config)#ipx route a1234b ffffff b1574c.00d0.58a8.e150
Router(config)#ipx route b1574c ffffff a1234b.00d0.58a8.e150
```

Теперь маршрутизатор из нашего IPX-сценария полностью работоспособен и готов маршрутизировать пакеты между сетями, изображенными на рис. 12.6. Но вы можете встретиться и с такой IPX-средой, как на рис. 12.7.

Обратите внимание на то, что две сети IPX, подключенные к интерфейсу `ethernet 0`, используют разные типы инкапсуляции. Можно настроить маршрутизатор Cisco так, чтобы несколько методов инкапсу-

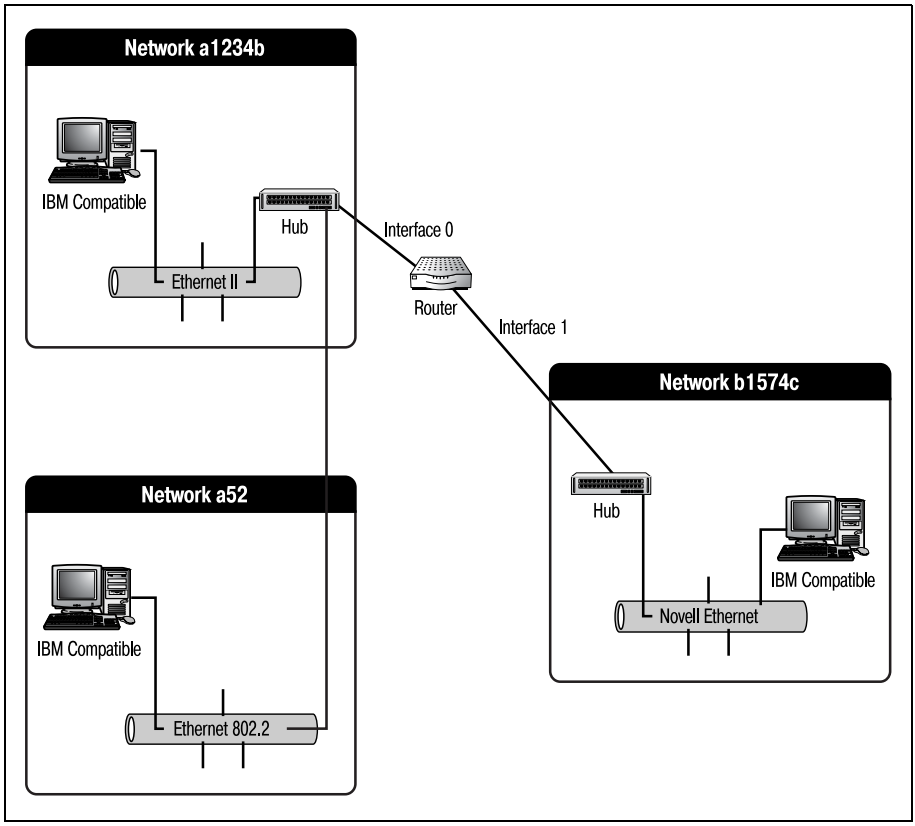

Рис. 12.7. *Среда, состоящая из трех IPX-сетей*

ляции могли использоваться на одном и том же интерфейсе (при этом различные типы инкапсуляции задаются для разных номеров сетей). Например, чтобы сконфигурировать вторую сеть для интерфейса ethernet 0, используйте такие команды:

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ipx routing
Router(config)#^Z
Router#
1d00h: %SYS-5-CONFIG_I: Configured from console by console
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface ethernet 0
Router(config-if)#ipx network a1234b encapsulation arpa
Router(config-if)#ipx network a52 encapsulation sap secondary
Router(config-if)#no shutdown
Router(config-if)#interface ethernet 1
Router(config-if)#ipx network b1574c encapsulation novell-ether
Router(config-if)#no shutdown
Router(config-if)#^Z
Router#
1d00h: %SYS-5-CONFIG_I: Configured from console by console
Router#
```

Ключевое слово secondary позволят указать более чем один IPX-адрес для интерфейса. Если бы мы решили просмотреть конфигурацию IPX для этого маршрутизатора (с помощью команды show ipx interface), то увидели бы, что оба интерфейса настроены на соответствующие им сети и ethernet 0 сконфигурирован для работы с двумя сетями и двумя типами инкапсуляции:

```
Router#show ipx interface
Ethernet0 is administratively up, line protocol is up
  IPX address is A52.00d0.58a8.e150, SAP [up]
  Delay of this IPX network, in ticks is 1 throughput 0 link delay 0
  IPXWAN processing not enabled on this interface.
  Secondary address is A1234B.00d0.58a8.e150, ARPA [up]
  Delay of this IPX network, in ticks is 1
  IPX SAP update interval is 60 seconds
  IPX type 20 propagation packet forwarding is disabled
  Incoming access list is not set
  Outgoing access list is not set
  IPX helper access list is not set
  SAP GNS processing enabled, delay 0 ms, output filter list is not set
  SAP Input filter list is not set
  SAP Output filter list is not set
  SAP Router filter list is not set
  Input filter list is not set
  Output filter list is not set
```

```
  Router filter list is not set
  Netbios Input host access list is not set
  Netbios Input bytes access list is not set
  Netbios Output host access list is not set
  Netbios Output bytes access list is not set
  Updates each 60 seconds aging multiples RIP: 3 SAP: 3
  SAP interpacket delay is 55 ms, maximum size is 480 bytes
  RIP interpacket delay is 55 ms, maximum size is 432 bytes
  RIP response delay is not set
  IPX accounting is disabled
  IPX fast switching is configured (enabled)
  RIP packets received 0, RIP packets sent 0
  SAP packets received 0, SAP packets sent 0
Ethernet1 is administratively up, line protocol is up
  IPX address is B1574C.00d0.58a8.e151, NOVELL-ETHER [up]
  Delay of this IPX network, in ticks is 1 throughput 0 link delay 0
  IPXWAN processing not enabled on this interface.
  IPX SAP update interval is 60 seconds
  IPX type 20 propagation packet forwarding is disabled
  Incoming access list is not set
  Outgoing access list is not set
  IPX helper access list is not set
  SAP GNS processing enabled, delay 0 ms, output filter list is not set
  SAP Input filter list is not set
  SAP Output filter list is not set
  SAP Router filter list is not set
  Input filter list is not set
  Output filter list is not set
  Router filter list is not set
  Netbios Input host access list is not set
  Netbios Input bytes access list is not set
  Netbios Output host access list is not set
  Netbios Output bytes access list is not set
  Updates each 60 seconds aging multiples RIP: 3 SAP: 3
  SAP interpacket delay is 55 ms, maximum size is 480 bytes
  RIP interpacket delay is 55 ms, maximum size is 432 bytes
  RIP response delay is not set
  IPX accounting is disabled
  IPX fast switching is configured (enabled)
  RIP packets received 0, RIP packets sent 0
  SAP packets received 0, SAP packets sent 0
```

Вы получили представление о командах, необходимых для настройки маршрутизатора Cisco на работу с IPX в нескольких разных (достаточно простых) сценариях. Эта глава завершает часть книги, отведенную на маршрутизируемые протоколы локальных сетей. Оставшиеся главы познакомят вас с протоколами глобальных сетей и протоколами маршрутизации.

Резюме

- IPX – это протокол сетевого уровня, похожий на IP.
- Адреса протокола IPX представляются в шестнадцатеричном формате.
- Адрес узла IPX-адреса состоит из MAC-адреса устройства.
- Для конфигурирования IPX на маршрутизаторе Cisco используется команда `ipx network`.
- Тип инкапсуляции должен оставаться постоянным для всей сети и соответствовать типу этой сети.
- Один интерфейс Cisco может быть настроен для работы с несколькими сетевыми адресами и типами инкапсуляции.

Вопросы и ответы

Вопрос Почему для IPX существуют разные типы инкапсуляции?

Ответ IPX долгое время был патентованным протоколом. Поэтому когда компания Novell изменяла заголовочную информацию или версии протокола, это касалось только ее собственных устройств. После распространения технологии понадобилось различать типы сетей.

Вопрос Может ли ключевое слово `secondary` использоваться в других протоколах?

Ответ Да, многие маршрутизируемые протоколы, такие как IP и IPX, могут использовать ключевое слово `secondary`. Это делает среду маршрутизации более гибкой.

Тест

Вопросы

1. Какой тип инкапсуляции наиболее распространен сегодня в сетях Ethernet?
2. Что определяет параметр `<float>`?
3. Сколько типов инкапсуляции может быть определено для одной сети?
4. Сколько шестнадцатеричных разрядов в сетевом адресе IPX?

Ответы

1. Ethernet_II (`arpa`).
2. Он позволяет записать динамический маршрут поверх данного статического маршрута.
3. Один. Во всей сети может быть задан только один тип инкапсуляции. (Но если у вас несколько сетей, то и типов инкапсуляции может быть несколько.)
4. Восемь.

Упражнение

1. В примере представлены команды, используемые для конфигурирования маршрутизатора, соединяющего две IPX-сети. За перечнем команд следует вывод команды `show ipx interface`. Если считать, что методы инкапсуляции корректны, то почему маршрутизатор не работает?

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ipx routing
Router(config)#^Z
Router#
1d00h: %SYS-5-CONFIG_I: Configured from console by console
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface ethernet 0
Router(config-if)#ipx network ab encapsulation arpa
Router(config-if)#interface ethernet 1
Router(config-if)#ipx network bc encapsulation arpa
Router(config-if)#^Z
Router#
1d00h: %SYS-5-CONFIG_I: Configured from console by console
Router#
```

Команда `show ipx interface` выводит такую информацию:

```
Router#show ipx interface
Ethernet0 is administratively down, line protocol is down
  IPX address is Ab.00d0.58a8.e150, SAP [up]
  Delay of this IPX network, in ticks is 1 throughput 0 link delay 0
  IPXWAN processing not enabled on this interface.
    IPX SAP update interval is 60 seconds
  IPX type 20 propagation packet forwarding is disabled
  Incoming access list is not set
  Outgoing access list is not set
  IPX helper access list is not set
  SAP GNS processing enabled, delay 0 ms, output filter list is not set
  SAP Input filter list is not set
  SAP Output filter list is not set
  SAP Router filter list is not set
  Input filter list is not set
  Output filter list is not set
  Router filter list is not set
  Netbios Input host access list is not set
  Netbios Input bytes access list is not set
  Netbios Output host access list is not set
  Netbios Output bytes access list is not set
  Updates each 60 seconds aging multiples RIP: 3 SAP: 3
  SAP interpacket delay is 55 ms, maximum size is 480 bytes
  RIP interpacket delay is 55 ms, maximum size is 432 bytes
```

```
    RIP response delay is not set
    IPX accounting is disabled
    IPX fast switching is configured (enabled)
    RIP packets received 0, RIP packets sent 0
    SAP packets received 0, SAP packets sent 0
  Ethernet1 is administratively down, line protocol is down
    IPX address is BC.00d0.58a8.e151, NOVELL-ETHER [up]
    Delay of this IPX network, in ticks is 1 throughput 0 link delay 0
    IPXWAN processing not enabled on this interface.
    IPX SAP update interval is 60 seconds
    IPX type 20 propagation packet forwarding is disabled
    Incoming access list is not set
    Outgoing access list is not set
    IPX helper access list is not set
    SAP GNS processing enabled, delay 0 ms, output filter list is not set
    SAP Input filter list is not set
    SAP Output filter list is not set
    SAP Router filter list is not set
    Input filter list is not set
    Output filter list is not set
    Router filter list is not set
    Netbios Input host access list is not set
    Netbios Input bytes access list is not set
    Netbios Output host access list is not set
    Netbios Output bytes access list is not set
    Updates each 60 seconds aging multiples RIP: 3 SAP: 3
    SAP interpacket delay is 55 ms, maximum size is 480 bytes
    RIP interpacket delay is 55 ms, maximum size is 432 bytes
    RIP response delay is not set
    IPX accounting is disabled
    IPX fast switching is configured (enabled)
    RIP packets received 0, RIP packets sent 0
    SAP packets received 0, SAP packets sent 0
```

Ответ

Не были включены интерфейсы. Для их активации требуется применить команду `no shutdown`**.**

13

Введение в протоколы глобальных сетей

В этой главе мы займемся изучением протоколов глобальных сетей (WAN-протоколов). Обсуждаемые до этого момента сети были небольшими локальными сетями (LAN). В рассмотренных сценариях использовалась единая среда, связанная одним маршрутизируемым протоколом. Но сегодня, в эпоху межкорпоративных коммуникаций, все реже встречаются маленькие локальные сети.

WAN-протоколы – это специальные протоколы, передающие данные по каналам глобальных сетей. Физические каналы, соединяющие две или более среды, которые находятся на большом расстоянии друг от друга, требуют особого внимания, когда речь заходит о передаче информации. Такие каналы используют протоколы, специально разработанные для выполнения одной задачи: обеспечения коммуникаций в глобальных сетях.

При рассмотрении протоколов WAN будут затронуты такие темы:

- ISDN
- X.25
- Frame Relay

WAN-протоколы, рассматриваемые в этой главе, являются хорошей иллюстрацией технологий, которые вы можете встретить в реальной жизни. Все они (от наименее мощного ISDN и до имеющего самые обширные возможности Frame Relay) имеют свой набор требований к маршрутизации. Но прежде чем заняться тщательным изучением современных протоколов, необходимо поговорить об аппаратном обеспечении маршрутизаторов Cisco, предназначенном для работы с WAN-протоколами.

Некоторые маршрутизаторы Cisco поставляются без элементов, необходимых для поддержки глобальных коммуникаций. Таким устройствам необходим отдельный адаптер WAN-интерфейса (WIC). Имеет ли маршрутизатор оборудование для работы с WAN (или может ли оно быть установлено дополнительно), зависит от серии (и даже внутри одной серии не все устройства оборудованы одинаково).

Например, в серии 1600 некоторые модели снабжены интерфейсами WAN, в то время как другие имеют только слот для WIC. Какие именно WIC могут быть установлены в слоты, также зависит от модели маршрутизатора. В табл. 13.1 представлена информация о том, какие маршрутизаторы Cisco серии 1600 оснащены интерфейсами WAN и использование каких WIC они допускают.

Таблица 13.1. Сравнение WAN-интерфейсов

Модель	WAN-интерфейс	Доступные адаптеры WIC					
		Последовательный	T1 CSU/DSU	56 Кбит/с CSU/DSU	ISDN ST с набором номера	ISDNU	Выделенный ISDN ST
1601	Последовательный	X	X	X	X	X	
1602	56 Кбит/с CSU/DSU	X	X	X	X	X	
1603	ISDN ST с набором номера	X	X	X			X
1604	ISDN ST с набором номера/ISDN-телефон	X	X	X			X
1605	N/A	X	X	X	X	X	

Как видно из таблицы, некоторые маршрутизаторы серии 1600 имеют встроенные WAN-интерфейсы (и допускают установку дополнительных адаптеров в слот WIC), а некоторые (например, 1605) могут использовать только адаптеры WIC.

В оставшейся части главы мы будем изучать эти интерфейсы и протоколы, которые по ним работают. Объединение небольших независимых сетей в большую корпоративную сеть является важной составляющей при изучении маршрутизации Cisco. Одним из ключей к созданию больших и очень больших сетей является освоение WAN-протоколов.

Первым WAN-протоколом, который мы рассмотрим, будет ISDN (Integrated Services Digital Network – цифровая сеть с интегрированными службами). Хотя сегодня ISDN уже не занимает на рынке таких позиций, как несколько лет назад, эта технология все еще используется. Поговорив об ISDN, мы перейдем к технологиям, лежащим в основе других WAN-протоколов, таких как Frame Relay и X.25.

ISDN

ISDN (Integrated Services Digital Network) – это первый из широкополосных продуктов потребительского класса, который получил широкое распространение. Обеспечивая цифровую передачу данных по линиям связи общего доступа, сеть ISDN очень быстро стала популярной у населения (и так было до тех пор, пока самыми быстрыми устройствами связи оставались модемы 56 Кбит/с).

Прежде чем стать популярной технологией для домашнего пользования, ISDN процветала на уровне предприятий.

Одним из основных достоинств ISDN была цена. ISDN предлагает недорогое решение для компаний, которым необходимо соединение между площадками, обладающее высокой пропускной способностью, но которым не нужна 100% работоспособность (или большая пропускная способность) каналов T1. Каналы ISDN могут работать в режиме набора номера с оплатой за использованное время. Другими словами, вы платите только за то, что фактически использовали. Эта особенность сделала ISDN популярным выбором для каналов WAN, которые используются периодически и не требуют постоянной готовности.

Популярности ISDN способствовало и то, что сеть работает по существующим телефонным линиям, то есть клиенты ISDN могут пользоваться имеющейся телефонной инфраструктурой. Домашние пользователи могут получать телефонные звонки одновременно с использованием сервисов ISDN на той же линии.

В наши дни уже осталось мало предприятий, применяющих соединения ISDN, но многие пользователи SOHO (small office/home office – малый офис/домашний офис) все еще верны этому недорогому широкополосному доступу по существующим телефонным кабелям. Домашним офисам особенно удобна такая форма оплаты за фактическое использование, так как сеть, как правило, не используется круглосуточно. Когда персональный компьютер выключается, домашний пользователь перестает оплачивать сервисы WAN.

Очень распространено применение ISDN для связи домашнего офиса с центральным, то есть для настоящей глобальной сети. Сегодня у вас больше всего шансов встретить ISDN именно в таком качестве. К тому же, именно в таком ISDN-подключении чаще всего используется маршрутизатор Cisco.

Технология ISDN

ISDN – это цифровая линия передачи данных, состоящая из нескольких каналов. Наиболее распространенным типом служб ISDN является BRI (Basic Rate Interface – базовый интерфейс). Большинство маршрутизаторов Cisco может работать с BRI ISDN. Поддержка ISDN предоставляется в функциональном пакете Basic IP многих редакций Cisco IOS.

Примечание

Прежде чем пытаться настроить ISDN на вашем маршрутизаторе, убедитесь в наличии соответствующего оборудования.

Базовый интерфейс ISDN BRI состоит из трех каналов: двух опорных B-каналов (bearer – носитель) и одного D-канала (data – данные). Два B-канала – это цифровые каналы, которые используются для передачи данных, в то время как D-канал используется для передачи управляющих сигналов. Комбинация таких каналов образует линию, способную передавать 144 Кбит/с. Пропускная способность линии BRI ISDN отражена в табл. 13.2.

Таблица 13.2. Пропускная способность BRI-каналов

Тип канала	Количество каналов в линии BRI	Скорость передачи данных
B	2	64 Кбит/с (каждый)
D	1	16 Кбит/с

Так как ISDN – это службы с оплатой за использование, то большая часть ISDN-соединений осуществляется через наборное устройство. Аналогично тому, как это происходит на стандартной телефонной линии, ISDN-модемы должны позвонить в центральный офис и инициировать сервис. ISDN-интерфейсы (как встроенные, так и адаптеры WIC) большинства маршрутизаторов Cisco поддерживают функции ISDN-модема.

Прежде чем конфигурировать службы ISDN на маршрутизаторе Cisco, необходимо получить от ISDN-провайдера некоторую ключевую информацию, необходимую для настройки маршрутизатора. В процессе конфигурирования маршрутизатор попросит вас ввести тип коммутатора, метод инкапсуляции и идентификатор SPID (Service Profile ID – идентификатор профиля службы); о том, для чего нужны эти данные, будет рассказано в следующих разделах. Зная, какая информация необходима для установления ISDN-соединения, рассмотрим этапы настройки маршрутизатора Cisco на работу с ISDN.

Под «ISDN» может подразумеваться не только служба соединения каналов WAN, но и набор протоколов, используемых для перемещения данных по этим каналам. ISDN – это динамический набор протоколов, обеспечивающий функциональные потребности различных уровней модели OSI. Хотя специфические функции каждого из протоколов ISDN выходят за рамки нашего обсуждения, но о типах протоколов, входящих в ISDN, мы поговорим в следующем разделе.

Терминология ISDN

Как и у многих других протоколов, у ISDN есть свои специфические термины и аббревиатуры, которые используются в литературе по ISDN и в описании свойств интерфейсов Cisco. В табл. 13.3 и 13.4 приведены основные понятия терминологии ISDN.

Пояснение терминов, использующихся в документации по ISDN для обозначения различных видов оборудования сети ISDN, представлено в табл. 13.3. Такое оборудование может присутствовать как на стороне сервера, так и на стороне клиента.

Таблица 13.3. Типы оборудования ISDN

Обозначение	Термин ISDN	Оборудование ISDN
TE1	Терминальное оборудование типа 1	Это устройство содержит внутри себя все компоненты, необходимые для работы с ISDN. Другими словами, такому маршрутизатору не нужен отдельный модем. Большинство маршрутизаторов Cisco, поддерживающих ISDN, относятся к этому типу
TE2	Терминальное оборудование типа 2	Такое оборудование использует для подключения к сети ISDN терминальный адаптер и модем
TA	Терминальный адаптер	Терминальный адаптер – это интерфейс ISDN, не обладающий функциональностью модема. То есть маршрутизатору, оснащенному терминальным адаптером, необходим отдельный ISDN-модем для установления ISDN-соединения
NT1	Оконечное оборудование сети типа 1	Оборудование NT1 используется для сопряжения сигнала ISDN с оконечным устройством, например мультиплексором
NT2	Оконечное оборудование сети типа 2	Такой тип устройств используется для коммутации или передачи ISDN-сигнала оконечному оборудованию типа 1

Сокращения и термины, обозначающие типы оборудования ISDN, используются повсеместно, в том числе и в литературе по Cisco. Хотя большая часть маршрутизаторов Cisco оснащена по типу TE1, вы должны знать, что бывают ситуации, в которых необходимо использовать интерфейсы TE2.

Еще один ряд терминов и сокращений относится к протоколам, входящим в стек ISDN. Стек ISDN образован протоколами, работающими на первых четырех уровнях модели OSI. То есть протоколы, составляющие ISDN, совместно работают на транспортном, сетевом, канальном и физическом уровнях. Для удобства обращения протоколы разбиты на категории, которые представлены в табл. 13.4.

Таблица 13.4. Категории протоколов ISDN

Термин ISDN	Категория протокола
Q	Эти протоколы используются для передачи управляющих сигналов. Обычно они работают по D-каналу BRI
E	Эта часть протоколов ISDN используется для передачи данных по существующим телефонным сетям. Они работают как стандартные сетевые протоколы (во многом аналогично IP и IPX)
I	Эта часть протоколов используется только для определения ряда концепций технологии ISDN. Они также определяют термины и службы ISDN-соединений

Один из основных протоколов класса Q – это LAPD (Link Access Protocol [for Channel] D – протокол доступа к линии [для канала] D). Он работает аналогично рассмотренным выше протоколам, инкапсулируя данные, отправляемые по ISDN. В отличие от протоколов класса E, обслуживающих сетевой уровень OSI (и в силу этого маршрутизируемых), LAPD работает на канальном уровне.

Одна из главных функций LAPD заключается в поддержке системы адресации ISDN в части взаимодействия с различными устройствами, образующими ISDN-среду. Эти устройства включают в себя и маршрутизаторы Cisco, расположенные у потребителей, и оборудование телефонных сетей общего пользования.

Примечание

В этой книге основное внимание уделяется протоколам сетевого уровня, но вы должны иметь представление о том, какую роль играет LAPD в стеке протоколов ISDN.

Имеется ряд терминов, описывающих опорные точки ISDN-сети. Опорными точками называются элементы ISDN-соединения, связанные с определенными функциями протокола ISDN. В табл. 13.5 перечислены названия и функции опорных точек ISDN.

Таблица 13.5. Опорные точки ISDN

Термин ISDN	Опорные точки сети
R	Соединение между TA и любым другим оборудованием
T	Соединение между NT1 и NT2
S	Соединение между конечной системой и NT2
U	Соединение между NT1 и сетью общего пользования

Теперь, когда вы познакомились с терминологией, применяемой в ISDN, давайте обсудим, как это многообразие понятий работает на практике. В следующем разделе мы рассмотрим принципы функционирования ISDN.

Функционирование ISDN

Протокол ISDN предназначен для работы с WAN-сетях. Вследствие этого забота о поддержке ISDN-соединений (опорных точек типа U) ложится на провайдера. Тем не менее, понимание принципов работы ISDN помогает определить, какая информация требуется маршрутизатору при создании ISDN-соединения.

Для потребителей опорная точка типа U представляется в виде прозрачной интеграции географически удаленных объектов. На рис. 13.1 показано ISDN-соединение с позиции клиента.

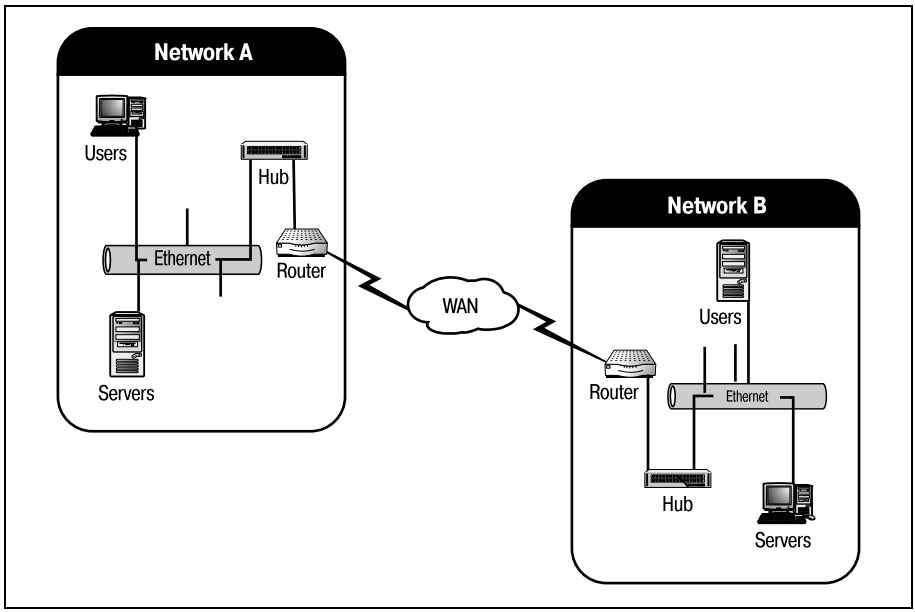

Рис. 13.1. *ISDN-соединение с позиции клиента*

Эта схема корректна и с технической точки зрения. На рис. 13.1 показано, что в ведении инженера по оборудованию Cisco находятся устройства, относящиеся к среде ISDN. Когда вы конфигурируете маршрутизатор (или пару маршрутизаторов), сеть представляется вам именно таким образом. Остальное оборудование, расположенное между этими двумя точками, находится в ведении владельца или оператора сети.

Оператор, ответственный за промежуточное оборудование, поддерживает устройства NT1 и NT2, передающие сигнал ISDN между потребителями. На рис. 13.2 полностью показано оборудование ISDN, соответствующее среде, представленной на рис. 13.1.

Чтобы разобраться во взаимодействии оборудования, изображенного на рис. 13.2, проследим путь данных, передаваемых из сети A в сеть B. Изначально информация находится в сети A в форме IP-пакетов. IP-

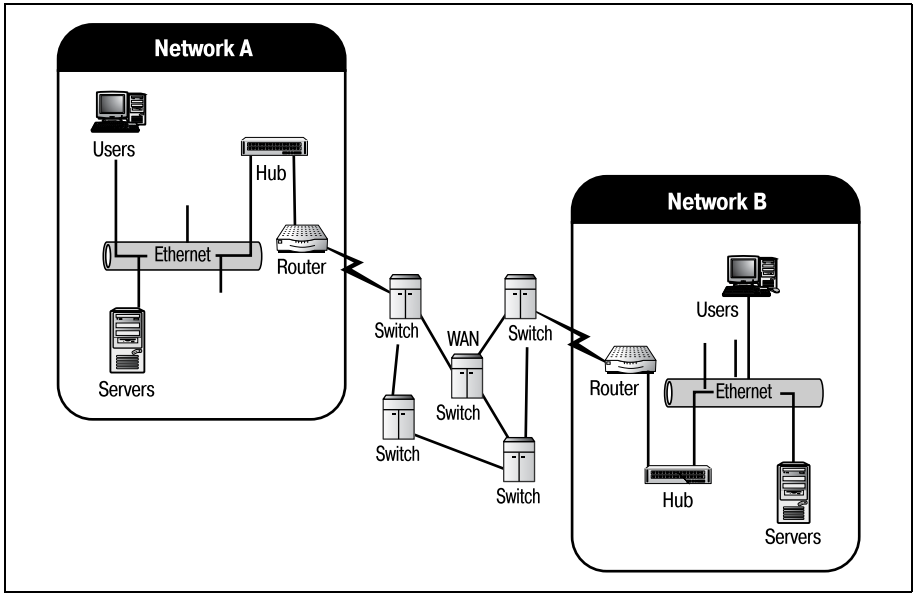

Рис. 13.2. *Полное представление ISDN среды, включающее опорные точки*

пакеты формируются обычным способом (вне зависимости от способа их доставки адресату). IP-адрес назначения (находящийся в сети B) помещается в пакет, и тот отправляется на маршрутизатор A (устройство типа TE1).

Маршрутизатор A сравнивает адрес назначения IP-пакета с имеющейся таблицей маршрутов. В результате сравнения выясняется, что сеть B доступна только через WAN-интерфейс (настроенный для работы по ISDN как опорная точка типа U). Маршрутизатор подготавливает IP-пакет для передачи по ISDN, упаковывая их в «кадры», то есть выполняя инкапсуляцию в соответствии с типом коммутатора (switch), используемого оператором сети. При проектировании протокола ISDN в него была заложена возможность работы с коммутируемыми сетями общего пользования.

Примечание

Термин «коммутируемые сети общего пользования» (PSN, public switching networks) относится к сетям телефонных компаний. Многие из этих сетей появились задолго до ISDN и не могли быть модернизированы по разработанному позднее единому стандарту. В результате возникла необходимость в разработке различных протоколов ISDN (E, Q и т. д.), способных работать в разнообразных существующих сетях. Другими словами, каждый протокол в рамках ISDN может быть сконфигурирован для работы с разными типами опорных точек и коммутаторов.

После кадрирования (инкапсуляции) пакета ISDN-интерфейс устройства TE1 пытается установить соединение с сетью оператора. Интер-

фейс ISDN набирает номер сети оператора, идентифицирует себя и посылает ISDN-кадры в телефонную сеть. Теперь телефонная сеть отвечает за доставку кадров в сеть B (основываясь на ISDN-адресе пункта назначения).

Когда кадры достигают устройства TE1 сети B, служебные поля кадров удаляются и данные снова принимают форму IP-пакетов. Теперь эти IP-пакеты могут маршрутизироваться к адресату. Маршрутизаторы Cisco могут быть сконфигурированы для преобразования IP-пакетов (или любых других протоколов) в кадры, пригодные к передаче по ISDN.

В следующем разделе этой главы объясняется процесс конфигурирования маршрутизатора Cisco для работы с ISDN. Таким образом, вы можете превратить ваш маршрутизатор в полнофункциональное устройство TE1 или TE2, в зависимости от конфигурации имеющегося у вас оборудования.

Конфигурирование ISDN

Теперь, когда вы поняли, как работает ISDN, мы можем обсудить, как нам сконфигурировать маршрутизатор Cisco для успешной работы в ISDN-среде. Для конфигурирования маршрутизатора в качестве полноценного устройства TE1 или TE2 вам понадобится информация, приведенная в табл. 13.6, включающая сведения об идентификаторе SPID и типе коммутатора.

До сих пор для конфигурирования протоколов на маршрутизаторе Cisco требовалась только общая информация, например протокольный адрес, назначенный интерфейсу, и тип инкапсуляции в случае использования IPX.

Примечание

Протоколы WAN, будучи разновидностью маршрутизируемых протоколов, требуют указания большинства той же информации, что и остальные маршрутизируемые протоколы. Однако в силу того, что WAN-протоколы используют соединения между разнородными сетями, не имеющими представления друг о друге (например, ЛВС и телефонная сеть), для успешного установления соединения им необходима более детальная информация.

Конфигурирование ISDN более сложно, и для успешной работы требуется указание большего количества параметров, чем для других маршрутизируемых протоколов. Но по сути дела, ISDN требует указания информации того же типа, что и другие протоколы. В табл. 13.6 представлены данные, необходимые для настройки ISDN.

Примечание

Часть информации, описанной в табл. 13.6, может быть получена у вашего провайдера ISDN.

Таблица 13.6. Информация, необходимая для конфигурирования ISDN

Требуемые данные	Назначение
Тип коммутатора ISDN	Сообщает маршрутизатору, коммутаторы какого типа используются в сети оператора. Всего может быть свыше 15 типов коммутаторов
SPID	Идентификатор SPID используется для задания типа сервиса, доступного в определенном соединении. Каждое ISDN-соединение в сети может иметь до двух идентификаторов SPID
Метод инкапсуляции	Используется совместно с Frame Relay или X.25
Протокольный адрес	Поскольку по ISDN-соединениям передаются данные маршрутизируемых протоколов, им необходим соответствующий адрес

Давайте начнем настройку маршрутизатора Cisco со входа в режим глобального конфигурирования. В этом режиме вы можете указать тип коммутатора. Задание типа коммутатора в глобальном режиме означает, что это значение распространяется на все ISDN-интерфейсы, которые могут быть установлены в маршрутизаторе. В табл. 13.7 показаны типы коммутаторов, применяемые в Северной Америке.

Примечание

Типы коммутаторов, начинающиеся со слова «primary», могут работать только с каналом PRI (Primary Rate Interface – первичный интерфейс) ISDN, который состоит из 23-х[1] B-каналов и одного 64 Кбит/с D-канала.

Таблица 13.7. Типы коммутаторов, применяемые в Северной Америке

Тип коммутатора	Описание
basic-5ess	Коммутаторы BRI AT&T
basic-dms100	Коммутаторы BRI DMS-100
basic-ni1	Национальные коммутаторы ISDN-1
primary-4ess	Коммутаторы AT&T 4ESS (только PRI ISDN)
primary-5ess	Коммутаторы AT&T 5ESS (только PRI ISDN)
primary-dms100	Коммутаторы NT DMS-100 (только PRI ISDN)

Находясь в глобальном режиме, установите тип коммутатора ISDN. Вы можете узнать тип вашего коммутатора у своего ISDN-провайдера. Выполните следующие команды для входа в режим глобального конфигурирования:

```
Router#configure terminal
Router(config)#isdn switch-type basic-ni1
```

[1] Верно для США. В Европе (и в России) 30 B-каналов. – *Примеч. науч. ред.*

Установив тип коммутатора, вы можете указать специфичную для данного интерфейса информацию, которая включает протокольный адрес, SPID и метод инкапсуляции. Но сначала установим значение идентификатора SPID. Имейте в виду, что может быть один, два или ни одного идентификатора – это зависит от типа сервиса, предоставляемого вашему узлу. Уточните значение SPID у своего оператора. Для указания значения идентификатора используйте команды:

```
Router(config)#interface bri 0
Router(config-if)#isdn spid 1 123456789
Router(config-if)#isdn spid 2 987654321
```

Теперь установите метод инкапсуляции. Вы можете выбрать PPP (Point-to-Point Protocol – протокол «точка-точка») либо HDLC (High-level Data Link Control – высокоуровневый протокол управления каналом). Это опять-таки следует выяснить у своего провайдера (чаще применяется PPP). Для установки PPP в качестве метода инкапсуляции выполните:

```
Router(config-if)#encapsulation ppp
```

Наконец, надо присвоить интерфейсу, передающему ISDN-данные, адрес для маршрутизируемого протокола:

```
Router(config-if)#ip address 153.4.16.1 255.255.0.0
```

Теперь интерфейс сконфигурирован для работы с ISDN. Как и в случае с другими, рассмотренными ранее маршрутизируемыми протоколами, необходимо задать маршруты, направляющие IP- (или IPX-) пакеты в ISDN-канал. Для таких протоколов, как IP и IPX, используйте команду route для определения статических маршрутов. Для ISDN используйте команду dialer совместно со списком доступа, чтобы воспользоваться этими «статическими маршрутами». Рассмотрим некоторые шаги при конфигурировании ISDN-маршрутов.

```
Router(config)#dialer map ip 153.4.10.1 name NETWORK_B 234567891
Router(config)#dialer-group 1
Router(config)#dialer-list 1 list 99
Router(config)#access-list 99 permit 153.4.16.0 255.255.0.0 153.4.10.0
255.255.0.0
```

Хотя на первый взгляд такая последовательность команд может выглядеть обескураживающе, при ближайшем рассмотрении она оказывается вполне логичной и понятной. Первая строка (dialer map ip 153.4.10.1 name NETWORK_B 234567891) сообщает, что отображаемый маршрут должен быть установлен для любых ISDN-кадров, направляемых по IP-адресу 153.4.10.1 (вероятно, не принадлежащему локальной сети). Затем адрес 153.4.10.1 сопоставляется сети с именем NETWORK_B, имеющей ISDN-адрес 234567891.

Вторая строка создает группу интерфейсов вызова по номеру и присваивает ей имя 1. Эта группа служит для хранения карты маршрутов и обращения к ней. Третья строка (`dialer-list 1 list 99`) устанавливает связь между указанной группой (добавленной в список набора) и списком доступа IP.

Примечание

Один список набора может содержать несколько групп интерфейсов вызова по номеру.

Последняя строка в приведенной последовательности команд (`access-list 99 permit 153.4.16.0 255.255.0.0 153.4.10.0 255.255.0.0`) фактически создает список доступа IP, позволяющий маршрутизатору принимать пакеты из сети 153.4.16.0 (локальной) и отправлять их в сеть 153.4.10.0 (удаленную).

Иначе говоря, эти четыре команды делают следующее:

Разрешают маршрутизатору принимать IP-пакеты из сети 153.4.16.0 и отправлять их в сеть 153.4.10.0 через ISDN-соединение 234567891 (известное также под именем NETWORK_B).

Протокол ISDN используется наравне с другими, более прогрессивными протоколами WAN. Наиболее популярны из них X.25 и Frame Relay. Сначала мы рассмотрим протокол X.25, так как именно он лег в основу Frame Relay.

X.25

Протокол X.25 используется с середины 70-х годов. Как и большинство технологий, предназначенных для глобальных сетей, X.25 сначала появился в сетях общего пользования, используемых телефонными компаниями. Технология, послужившая основой для X.25, на десятилетие опередила большинство других, применявшихся в глобальных сетях. Другие протоколы WAN, как, например, Frame Relay, основаны на идеях, заложенных в X.25.

Этот раздел познакомит вас с X.25 в таком стиле, который облегчит понимание описанной далее технологии Frame Relay. X.25 – это большая тема, которая не может быть полностью рассмотрена в книге по маршрутизации для начинающих. Кроме того, в современных средах маршрутизации преобладает технология Frame Relay, которую мы рассмотрим гораздо более подробно.

Технологии виртуальных каналов и кадрирования пакетов впервые появились в X.25, а затем проникли и в другие протоколы WAN. Позднее обе они были включены в Frame Relay. Кадрирование пакетов уже обсуждалось вкратце в предыдущем разделе, теперь обратимся к теме каналов. Виртуальные каналы определяют путь, по которому

движутся данные, перемещаясь от одного устройства к другому. Два устройства могут образовать между собой виртуальный канал с целью исключить повреждение данных, передаваемых между ними. Виртуальные каналы, применяемые в X.25, могут быть двух типов: постоянные и коммутируемые.

Постоянные виртуальные каналы

Маршрутизаторы по определению спроектированы так, чтобы находить наилучший из имеющихся путь для передачи данных от устройства к устройству. При использовании такой среды передачи, как телефонная сеть общего пользования, это не самый лучший способ пересылки данных. ПК, отправляя данные, посылает их короткими пакетами. Пакеты попадают в сеть по одному. Принимающее устройство не требует, чтобы пакеты приходили в хронологическом порядке, поэтому они могут путешествовать по любым доступным маршрутам, даже если при этом нарушится порядок их прибытия.

Представьте себе, что такой же сценарий использовался бы при телефонном разговоре. Каждое сказанное вами слово будет уходить к получателю по новому маршруту. Порядок большинства слов нарушится, и разговор будет испорчен. Кроме этого, из-за того что каждое слово перемещается по своему пути, множество разговоров будут передаваться одновременно. Окажется практически невозможным определить, какие слова из этого потока адресованы вам.

По этой причине, когда совершается телефонный звонок, коммутаторы, обслуживающие телефонную сеть, создают на время разговора постоянный канал между абонентами. Во время соединения каждое слово, вздох и шум проделают одинаковый путь, что гарантирует правильный порядок и непрерывность всего сообщения. Еще более важно то, что другие разговоры не смогут использовать те же линии для установления соединения. Такое соединение называется *каналом* (*circuit*). Протоколы глобальных сетей, работающие в телефонных сетях общего пользования, научились использовать преимущества этого способа.

Постоянный виртуальный канал (PVC, Permanent Virtual Circuit) определяет фиксированный маршрут, по которому движутся данные, перемещаясь с места на место. Этот маршрут никогда не изменяется. Когда бы одно устройство не устанавливало сеанс связи с другим (на другом конце глобальной сети), всегда используется один и тот же путь, или канал. Постоянные виртуальные соединения можно рассматривать как нечто подобное статическим маршрутам. Аналогично тому как статические маршруты устанавливаются на маршрутизаторе Cisco для постоянного использования одного и того же пути между устройствами, постоянные виртуальные каналы образуют фиксированный путь между двумя объектами. На рис. 13.3 показан постоянный виртуальный канал между двумя сетями.

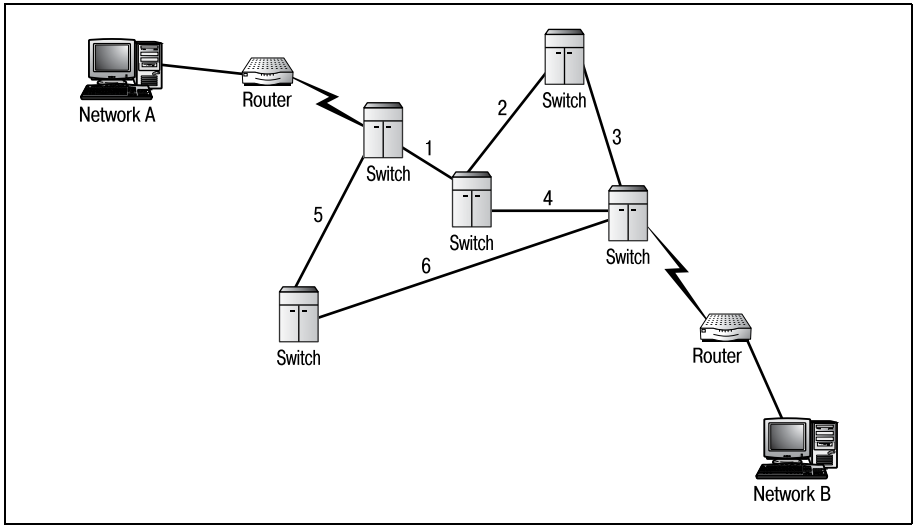

Рис. 13.3. *Постоянный виртуальный канал X.25. Любая информация, отправленная из сети A в сеть B, пройдет по пути «1:2:3»*

У постоянных виртуальных каналов имеется один существенный недостаток. Если одно из соединений будет нарушено (например, соединение 2 на рис. 13.3), то весь канал связи между сетями A и B окажется неработоспособным. Там, где маршрутизатор просто выбрал бы новый путь к адресату (обходя испорченный участок сети), постоянный виртуальный канал не сможет отклониться от предопределенного маршрута. Единственная точка уязвимости превращает глобальную сеть в бесполезный клубок кабелей. Избежать этого позволяют коммутируемые виртуальные каналы (SVC, Switched Virtual Circuits).

Коммутируемые виртуальные каналы

Имеется лишь одно различие между постоянными и коммутируемыми виртуальными каналами. В то время как постоянный канал неизменен, коммутируемый канал может отличаться для каждого сеанса связи. То есть при установлении сеанса связи между двумя устройствами будет создан канал на время этого сеанса. По завершении сеанса канал «разрывается». В следующем сеансе может быть создан уже другой канал.

Этот метод имеет ограниченную надежность в силу использования единственного канала для соединения устройств, но позволяет исключить уязвимость, связанную с единственной точкой отказа. Это увеличивает привлекательность коммутируемых виртуальных каналов по сравнению с постоянными.

Когда разрабатывался протокол Frame Relay, многие технологии и передовые решения, присутствующие в X.25, такие как коммутируемые виртуальные каналы, были адаптированы к новым, более высоким

требованиям современных сетей. Давайте перейдем к обсуждению концепций, лежащих в основе Frame Relay.

Frame Relay

Технология Frame Relay была разработана в конце 80-х в качестве более быстрой (усовершенствованной) версии X.25 для сетей ISDN. Компания Cisco была одним из пионеров в разработке, реализации и использовании Frame Relay. Благодаря этому практически все маршрутизаторы Cisco способны работать с WAN-каналами, использующими Frame Relay.

Frame Relay – это протокол для глобальных сетей, работающий на канальном уровне модели OSI. Это его главное отличие от X.25, работающего одновременно на двух уровнях – сетевом (маршрутизация) и канальном (коммутация). В целях оптимизации поддержка сетевого уровня была удалена из протокола. Благодаря этому коммутаторы получили возможность заниматься своей работой, а маршрутизаторы – своей.

Примечание

Аналогично тому, как маршрутизаторы работают исключительно на сетевом уровне модели OSI, коммутаторы работают на канальном уровне. Таким образом, Frame Relay является протоколом с коммутацией пакетов. Маршрутизаторы Cisco сами не занимаются передачей данных Frame Relay. Они, как и в случае с большинством WAN-протоколов, отправляют предварительно отформатированные данные коммутатору, чтобы тот переслал их другому маршрутизатору. Когда информация достигает маршрутизатора в сети назначения, данные преобразуются обратно в пригодный для маршрутизации формат.

Технология Frame Relay

Процесс конфигурирования маршрутизатора Cisco для соединения Frame Relay требует использования команд и технологий, которые мы до сих пор еще не использовали. Этот небольшой раздел познакомит вас с терминологией и технологией, которые потребуются при настройке маршрутизатора Cisco для работы с соединениями Frame Relay.

Оборудование, необходимое для установления соединения Frame Relay, делится на две категории: DTE (Data Terminal Equipment – терминальное оборудование) и DCE (Data Circuit Terminating Equipment – оконечное оборудование канала данных). Устройства DCE – это коммутаторы, образующие узлы коммутации пакетов (Packet Switch Node, PSN). Оборудование DCE никогда не встречается в пользовательских сетях Frame Relay, в отличие от устройств DTE, которые устанавливаются у пользователей WAN-сетей. Маршрутизаторы Cisco представляют собой устройства типа DTE. В отличие от ISDN, большинство

маршрутизаторов Cisco не имеют интерфейсов Frame Relay, поэтому для установления такого соединения требуется дополнительное оборудование – мультиплексор.

Мультиплексор преобразует канал T1 (или другой скоростной канал, применяемый в телефонии) в несколько отдельных каналов. Часть этих каналов может быть отдана под передачу данных Frame Relay, а часть – под другие нужды. Данные из каналов Frame Relay пересылаются (в последовательном виде) маршрутизатору Cisco. Маршрутизатор должен быть оборудован последовательным интерфейсом для того, чтобы с помощью мультиплексора установить соединение Frame Relay.

Примечание

Широкополосные каналы, такие как T1, в действительности состоят из меньших каналов, которые можно рассматривать как телефонные линии. T1 содержит 24 отдельных канала, которые могут использоваться как для телефонии, так и для передачи данных. Один канал T1 может передавать одновременно данные и голос. Мультиплексоры применяются для разделения потока T1 на отдельные каналы.

Оборудование мультиплексирования обычно предоставляется провайдерами каналов T1 или Frame Relay.

Еще один термин, часто употребляемый при конфигурировании Frame Relay, – это идентификатор канального уровня DLCI (data link connection identifier). Он представляет собой номер, присваиваемый соединению Frame Relay. Этот номер является уникальным и используется маршрутизаторами на обоих концах распределенной сети для установления соединения. Идентификатор канального уровня Frame Relay аналогичен идентификатору профиля обслуживания SPID в ISDN. Настраивая маршрутизатор Cisco для использования Frame Relay, вы должны знать идентификатор DLCI, присвоенный вашему соединению.

Маршрутизатор Cisco может получить идентификатор DLCI в передаваемом Frame Relay обновлении, известном также под названием обновления LMI (Local Management Interface – локальный интерфейс управления). Такой способ позволяет периодически изменять значение идентификатора DLCI без вмешательства администратора. Например, если провайдер использует SVC (коммутируемые виртуальные каналы), то DLCI может изменяться при каждом установлении соединения. Использование обновлений LMI для передачи DLCI позволяет маршрутизатору получать идентификаторы соединений. Единственный случай, когда вам необходимо знание идентификатора DLCI, – это конфигурирование статических карт маршрутов Frame Relay.

Примечание

Что касается каналов, Frame Relay может работать как с постоянными виртуальными каналами, так и коммутируемыми – выбор типа соединения обычно остается за ним.

В некоторых случаях вам может понадобиться информация, связанная с интерфейсом LMI. В LMI есть несколько параметров, которые могут быть установлены на маршрутизаторе, однако если вы хотите работать с LMI, необходимо, чтобы он поддерживался. Часто используемой возможностью LMI является оповещение о состоянии виртуального канала.

Если постоянный виртуальный канал PVC перестал функционировать, интерфейс LMI посылает маршрутизирующему оборудованию уведомление об этом. Подобно тому как это было описано в разделе, посвященном X.25, постоянные виртуальные каналы имеют уязвимость, связанную с наличием единственной точки отказа. Если хотя бы одно соединение из входящих в канал PVC откажет, весь канал окажется бесполезным. С помощью виртуального канала LMI узел PSN может известить маршрутизатор об этом отказе. Давайте воспользуемся полученной информацией для базового конфигурирования Frame Relay на маршрутизаторе Cisco.

Конфигурирование Frame Relay

После того как внешнее оборудование подключено к последовательному интерфейсу маршрутизатора, можно приступать к конфигурированию IOS. Frame Relay – один из новых протоколов, не инициализируемых в режиме глобального конфигурирования Cisco IOS. Так как большинство функций связано с внешним оборудованием, большая часть конфигурирования выполняется на уровне интерфейса.

Первым делом необходимо сообщить интерфейсу, какой метод инкапсуляции применяется в данной сети Frame Relay. Компания Cisco разработала собственный метод инкапсуляции, который может быть использован только на ее оборудовании. Если вам известно (от провайдера сети), что сеть построена на оборудовании Cisco, то можете оставить метод инкапсуляции по умолчанию.

Если же на узле коммутации установлено оборудование другого производителя либо у вас нет о нем сведений, то установите метод инкапсуляции IETF (Internet Engineering Task Force – проблемная группа проектирования Интернета). Этот метод широко применяется при межплатформенном взаимодействии и поддерживается всеми производителями оборудования. В общем, прежде чем настраивать метод инкапсуляции, проконсультируйтесь со своим провайдером Frame Relay.

Конфигурирование маршрутизатора Cisco для использования Frame Relay немного отличается от его конфигурирования для других протоколов. Можете взять приведенный здесь пример за основу, когда будете настраивать соединение Frame Relay.

```
Router#configure terminal
Router(config)#interface serial 0
Router(config-if)#encapsulation frame-relay ietf
Router(config-if)#no shutdown
```

В связи с тем, что маршрутизатор Cisco взаимодействует с мультиплексором через единственный (последовательный) интерфейс, может оказаться, что один последовательный порт должен работать с несколькими каналами Frame Relay. Нет правила, запрещающего каждому каналу, входящему в T1, образовывать отдельный канал Frame Relay. Чтобы справиться с этой проблемой, последовательный интерфейс Cisco поддерживает субинтерфейсы.

Примечание

Субинтерфейс представляет собой часть целого интерфейса. Один физический интерфейс может быть поделен на несколько логических субинтерфейсов, каждый из которых имеет собственные адрес и конфигурацию.

На одном последовательном интерфейсе вы можете сконфигурировать произвольное количество субинтерфейсов. Если ваша конфигурация предусматривает использование нескольких каналов Frame Relay, используйте такие команды для настройки субинтерфейсов:

```
Router#configure terminal
Router(config)#interface serial 0.1 point-to-point
Router(config-if)#encapsulation frame-relay ietf
Router(config-if)#no shutdown
```

Номер субинтерфейса отделяется от номера интерфейса точкой. Указывая `interface serial 0.1`, мы конфигурируем первый субинтерфейс последовательного интерфейса 0. (Третий субинтерфейс последовательного интерфейса 2 получит обозначение 2.3 и т. д.) Ключевое слово `point-to-point` показывает, что создается канал типа «точка-точка». Этот режим используется всегда, если только ваш провайдер не требует иного. Остальные шаги должны быть повторены для всех интерфейсов.

Теперь настало время настроить на маршрутизаторе идентификатор DLCI (предоставляемый Frame Relay). Хотя большинство шагов могут быть выполнены в произвольном порядке, DLCI понадобится при создании карты маршрутов Frame Relay, поэтому сейчас следует задать его значение.

Одновременно с DLCI вы должны указать адрес интерфейса в используемом маршрутизируемом протоколе. Этот шаг аналогичен тому, который вы выполняли при конфигурировании ISDN в начале этой главы. Адрес интерфейса для протокола (IP или IPX) будет использоваться при маршрутизации данных через этот интерфейс. Помните, что протоколы распределенных сетей, такие как Frame Relay, соединяют сети. Это означает, что сети, соединяемые Frame Relay, нуждаются в способе идентификации друг друга. Использование IP- или IPX-адресов делает распределенную сеть прозрачной по отношению к взаимодействию сетей. Используйте такие команды, чтобы связать адресуемый интерфейс маршрутизатора с каналом Frame Relay:

```
Router(config-if)#frame-relay interface-dlci 56
Router(config-if)#ip address 198.156.82.1 255.255.255.0
```

Когда все необходимые адреса настроены, вы можете сконфигурировать статическое отображение адресов (mapping). Как вы увидите, формат команды похож на уже использовавшийся ранее `dialer-map` для ISDN. Команда для статического отображения Frame Relay выглядит так: `frame-relay map`. Эта команда связывает адресуемую по IP сеть с определенным каналом Frame Relay. Обратите внимание на синтаксис этой команды в следующем примере:

```
Router(config)#frame-relay map ip 198.156.81.0 56
```

На обычном языке эта команда звучит так: «маршрутизация всех данных для IP-сети 198.156.81.0 выполняется через канал Frame Relay, имеющий идентификатор (DLCI) 56». Вообще, эта команда принимает такие параметры:

```
#frame-relay map <protocol> <protocol address> <dlci number>
```

На этом базовое конфигурирование Frame Relay заканчивается. Прежде чем перейти к следующей главе, выполните упражнения, приведенные в конце этой главы. Они помогут вам закрепить рассмотренный выше материал.

Начиная со следующей главы и до конца книги основной темой будут протоколы маршрутизации. Эти протоколы лежат в основе функционирования маршрутизаторов Cisco, и оставшиеся главы посвящены их детальному обсуждению.

Резюме

- Сервисы ISDN делятся на две категории, наиболее распространенная из которых – BRI.
- Сервис BRI состоит из двух несущих каналов (B) и одного канала для сигналов управления (D).
- ISDN – это набор протоколов, реализующий четыре нижних уровня модели OSI.
- Протокол X.25 использует метод виртуальных каналов.
- Frame Relay – это усовершенствованный вариант X.25 для существующих сетей ISDN.
- Поддержка третьего уровня OSI, имеющаяся в X.25, отсутствует в Frame Relay.

Вопросы и ответы

Вопрос Если большинство протоколов WAN работает на канальном (коммутируемом) уровне, как они могут обрабатываться маршрутизатором Cisco?

Ответ Помните, что маршрутизатор Cisco не маршрутизирует данные в формате Frame Relay или ISDN; он инкапсулирует маршрутизируемые данные в формат соответствующего протокола и отправляет их коммутатору. Маршрутизатор не интересуется тем, как данные коммутируются во время доставки с места на место, – он просто отправляет информацию в нужном формате узлу коммутации пакетов.

Тест

Вопросы

1. Какова максимальная пропускная способность канала BRI ISDN?
2. Каково назначение канала D BRI?
3. Что такое ISDN TE1?
4. На каком уровне (уровнях) модели OSI работает X.25?
5. Что такое идентификатор, назначаемый каналу Frame Relay?

Ответы

1. 144 Кбит/с: 64 Кбит/с для каждого из двух B-каналов и 16 Кбит/с для D-канала.
2. Канал D используется для передачи управляющих сигналов.
3. Применительно к ISDN TE1 означает Type-1 terminal equipment (терминальное оборудование первого типа).
4. Сетевом, канальном и физическом.
5. DLCI (Data Link Connection Identifier – идентификатор канального уровня).

Упражнения

1. В приведенной последовательности пропущена одна строка. Добавьте необходимую команду.

```
1>    Router#configure terminal
2>    Router(config)#isdn switch-type basic-nil
3>    Router(config)#interface bri 0
4>    Router(config-if)#isdn spid 1 56342
Router(config-if)#isdn spid 2 98654
Router(config-if)#no shutdown
7>    Router(config-if)#encapsulation ppp
8>    Router(config-if)#ip address 198.37.134.1 255.255.255.0
9>    Router(config)#dialer map ip 198.37.135.0 name NETWORK_B 42156987
10>   Router(config)#dialer-group 1
```

```
11>    Router(config)#access-list 108 permit 198.37.134.0 255.255.255.0
198.37.135.0 255.255.255.0
```

Решение

Между строками 10 и 11 вставьте такую строку:

```
Router(config)#dialer-list 1 list 99
```

2. Сконфигурируйте субинтерфейс 14 (IP-адрес 198.52.62.1) интерфейса Frame Relay так, чтобы IP-сеть 198.52.63.0 маршрутизировалась через канал DLIC 130.

Решение

```
Router#configure terminal
Router(config)#interface serial 0.14 point-to-point
Router(config-if)#encapsulation frame-relay ietf
Router(config-if)#no shutdown
Router(config-if)#frame-relay interface-dlci 130
Router(config-if)#ip address 198.52.62.1 255.255.255.0
Router(config)#frame-relay map ip 198.52.63.0 130 cisco
```

14

Введение в протоколы маршрутизации

Данная глава служит введением в протоколы маршрутизации. До этого момента мы имели дело с двумя различными типами протоколов: протоколами глобальных сетей и маршрутизируемыми протоколами, ни один из которых в действительности не используется маршрутизаторами. Точнее говоря, эти протоколы – то, с чем работают маршрутизаторы, просто перемещая упакованные в них данные из одного места в другое. А протоколы маршрутизации – это те самые инструменты, с помощью которых маршрутизаторы выполняют свою задачу.

Маршрутизаторы используют протоколы маршрутизации для выполнения своей повседневной работы по транспортированию данных между сетями. Протоколы глобальных сетей и маршрутизируемые протоколы (такие как IP) служат лишь «материалом» для работы протоколов маршрутизации. Оставшаяся часть книги по мере углубления в процедуры конфигурирования маршрутизаторов и протоколов маршрутизации будет содержать все больше технических деталей.

В этой главе вы познакомитесь с основами «рабочих лошадок» маршрутизации – протоколов маршрутизации. Задача таких протоколов, как RIP, IGRP, OSPF и BGP, заключается в нахождении наилучшего пути, отправке по нему заключенных в маршрутизируемые протоколы данных и проверке того, что данные достигли пункта назначения. В этой главе рассматриваются следующие темы:

- Алгоритмы маршрутизации
- Маршрутизация протоколов
- Динамические обновления

Давайте изучим работу протоколов маршрутизации в два этапа. Первая часть работы протокола состоит в определении наилучшего пути для отправки данных. Маршрутизатор выполняет сложные расчеты для точного определения способа перемещения данных в сеть назначения. Ответственность за эту часть работы лежит на алгоритме маршрутизации.

Вторая часть работы протокола маршрутизации заключается в том, чтобы обеспечить все маршрутизаторы сети одинаковыми «картинами» сетевого окружения. Это значит, что все маршрутизаторы должны получать одну и ту же информацию о наличии или отсутствии сетевых соединений.

Далее в этой главе мы подробно обсудим обе эти составляющие протоколов маршрутизации. Начнем с изучения алгоритмов маршрутизации.

Алгоритмы маршрутизации

Независимо от типа алгоритм маршрутизации представляет собой математическую формулу, которую протокол маршрутизации использует для определения степени предпочтительности тех или иных путей между устройствами и сетями назначения. Маршрутизатор хранит этот алгоритм в памяти и обращается к нему, когда необходимо принять решение о маршрутизации. В протоколах маршрутизации могут использоваться два основных типа алгоритмов: вектор расстояния и состояние канала. Хотя эти алгоритмы используются во всех протоколах маршрутизации, их реализации могут различаться. Существуют, однако, общие термины, в которых описываются назначение и функционирование алгоритмов маршрутизации.

Примечание

Несмотря на то что реализация алгоритмов вектора расстояния и состояния канала может отличаться в разных протоколах, в каждом из протоколов алгоритм реализован одинаково независимо от платформы. Например, оба протокола маршрутизации RIP и IGRP используют алгоритм вектора расстояния. Реализация этого алгоритма может в них отличаться, но RIP как протокол реализован одинаково вне зависимости от производителя и модели маршрутизатора.

Протокол маршрутизации собирает определенную информацию о сетях и маршрутизаторах в своем окружении. Эта информация хранится в таблице маршрутов в памяти маршрутизатора. Данные из таблицы используются алгоритмом маршрутизации. Результат вычислений, проделанных в соответствии с алгоритмом, используется для определения наилучшего пути по определенному сценарию. В табл. 14.1 приведен пример таблицы маршрутов (для гипотетической маршрутизируемой среды).

Таблица 14.1. Пример таблицы маршрутов

Маршрут	Метрика
От маршрутизатора A к маршрутизатору B	2
От маршрутизатора B к маршрутизатору C	3
От маршрутизатора A к маршрутизатору C	6
От маршрутизатора C к маршрутизатору D	5

Примечание

Таблица маршрутов 14.1 приведена в упрощенном виде, чтобы можно было обсудить алгоритмы маршрутизации, не отвлекаясь на особенности отдельных протоколов.

Согласно нашему упрощенному алгоритму, «наилучшим» путем к месту назначения будет тот, который имеет наименьшее значение метрики. Когда маршрутизатор A получает пакет, адресованный маршрутизатору C, в таблице маршрутов он может найти два подходящих пути. В первом случае пакет может быть отправлен от маршрутизатора A непосредственно к C. Во втором – пакет посылается сначала от A к B, а затем к C. Для определения того, какой из способов лучше, используется алгоритм маршрутизации.

Глядя на значения метрик в табл. 14.1, сопоставленных каждому из возможных соединений, мы видим, что пути от маршрутизатора A к B и от B к C имеют суммарную метрику 5, в то время как прямой путь к маршрутизатору C имеет метрику 6. Алгоритм выбирает путь A–B–C и отправляет информацию по этому «наилучшему» маршруту.

Этот упрощенный пример показывает, как действует алгоритм маршрутизации, принимая решение о выборе маршрута. Протокол определяет, какие именно данные содержатся в таблице маршрутов и как они используются алгоритмом. Протоколы состояния канала и вектора расстояния могут использовать разные метрики для выработки решения. В последующих разделах мы обсудим различия между этими двумя типами алгоритмов.

Вектор расстояния

Алгоритмы вектора расстояния похожи на упрощенный алгоритм, использованный в нашем примере (табл. 14.1). Такие алгоритмы используют метрики (стоимости) с целью определения наилучшего пути к месту назначения. Путь с наименьшей общей стоимостью выбирается в качестве «наилучшего» пути.

При использовании в сети алгоритма вектора расстояния каждый маршрутизатор получает свои значения «стоимостей». Эти стоимости могут быть выбраны совершенно произвольно. Администраторы могут назначать любые значения, руководствуясь своими собственными со-

ображениями. Например, если стоимость равна пяти, число 5 не имеет никакого значения для внешнего наблюдателя, в то время как с точки зрения администратора оно может означать меру надежности канала.

Значения стоимостей могут вычисляться динамически, например в зависимости от длительности задержек в канале по сравнению с другими каналами. Все стоимости (назначенные явно и прочие) собраны в таблице маршрутов. Оттуда значения стоимостей извлекаются алгоритмом для расчета наиболее подходящего пути в данном сетевом окружении.

Для того чтобы маршрутизаторы могли заполнять свои таблицы маршрутов, им необходим механизм обмена имеющимися у них данными. Этот механизм называется обновлением маршрутов. В процессе обмена данными между протоколами вектора расстояния таблицы маршрутов полностью или частично пересылаются от маршрутизатора к маршрутизатору. В результате каждый из маршрутизаторов получает информацию, имеющуюся в таблицах остальных маршрутизаторов. Это позволяет каждому из них получить более полное представление о сетевом окружении и обоснованно выбрать лучший маршрут.

Примечание

Все протоколы маршрутизации, независимо от используемых алгоритмов, в той или иной форме выполняют обновление маршрутов. Хотя состав информации, ее количество и частота обновлений могут отличаться у разных протоколов, цель одна: обменяться с соседними маршрутизаторами информацией о маршрутах.

Примерами использования алгоритма вектора расстояния могут служить протоколы RIP (Routing Information Protocol – протокол маршрутной информации) и BGP (Border Gateway Protocol – протокол граничного шлюза), чаще всего применяемые в сетях, построенных на оборудовании Cisco. Другие распространенные протоколы, в частности OSPF (Open Shortest Path First – первоочередное открытие кратчайших маршрутов), являются протоколами с объявлением о состоянии канала. Эти протоколы работают немного иначе, чем их аналоги, использующие алгоритм вектора расстояния.

Состояние канала

Протоколы состояния канала работают по той же базовой схеме, что и алгоритмы вектора расстояния, в том смысле, что и те и другие выбирают путь с наименьшей стоимостью. Однако протоколы состояния канала действуют более локально. В то время как маршрутизатор, использующий протокол вектора расстояния, вычисляет полный путь пакета до места назначения, протокол состояния канала выбирает наилучший путь среди каналов, находящихся в непосредственной близости.

Например, протокол вектора расстояния может вычислить, что наилучшим путем от маршрутизатора A к маршрутизатору E будет A–C–D–E,

и отправит пакет именно по этому пути. В отличие от него, протокол состояния канала определит, что наилучший путь от A к E проходит через C, и отправит пакет ему, чтобы маршрутизатор C сам определил следующий шаг.

Такой метод больше подходит для больших систем, подверженных частым изменениям. В подобных вычислительных средах таблицы маршрутов могут быть очень большими. Поэтому таблица каждого маршрутизатора содержит лишь небольшую часть схемы сети. Когда какое-либо соединение становится недоступным (или меняет свое состояние), маршрутизатор посылает в сеть сообщение, извещающее об этом изменении.

Протоколы состояния канала и вектора расстояния выполняют маршрутизацию совсем по-разному. Обсуждая их в оставшихся главах этой книги, мы будем уточнять, как каждый из них проявляет себя в той или иной ситуации.

Динамическое обновление

Первая часть работы протокола маршрутизации – определить, каким должен быть наилучший путь для передачи данных. Вторая часть работы любого протокола заключается в рассылке информации об изменениях в маршрутах всем устройствам, участвующим в маршрутизации.

В большинстве протоколов маршрутизации в том или ином виде реализовано динамическое обновление. Другими словами, в эти протоколы встроен механизм, позволяющий маршрутизаторам обмениваться информацией в пределах выбранной области или группы. Хотя отдельные параметры, такие как состав данных и преодолеваемое ими количество переходов, могут меняться от протокола к протоколу, ряд общих правил справедлив для всех обновлений. В этом списке перечислены некоторые положения, общие для всех маршрутизаторов и протоколов маршрутизации, использующих динамическое обновление:

- Обновление информации в таблице маршрутов для алгоритма маршрутизации, используемого протоколом.
- Обеспечение каждого маршрутизатора корректной информацией о состоянии сети.
- Упрощение сетевых операций за счет предоставления маршрутизаторам уменьшенных таблиц.

Давайте разберемся, как эти правила используются для успешного завершения операций при обновлении. Когда мы перейдем к обсуждению конкретных протоколов, то рассмотрим эти правила более подробно.

Большинство маршрутизаторов собирают только ту информацию, которая непосредственно относится к их интерфейсам. Другими словами, маршрутизатор получает из первых рук сведения только о тех ка-

налах, которые напрямую присоединены к его интерфейсам. На рис. 14.1 показана связанная сетевая среда.

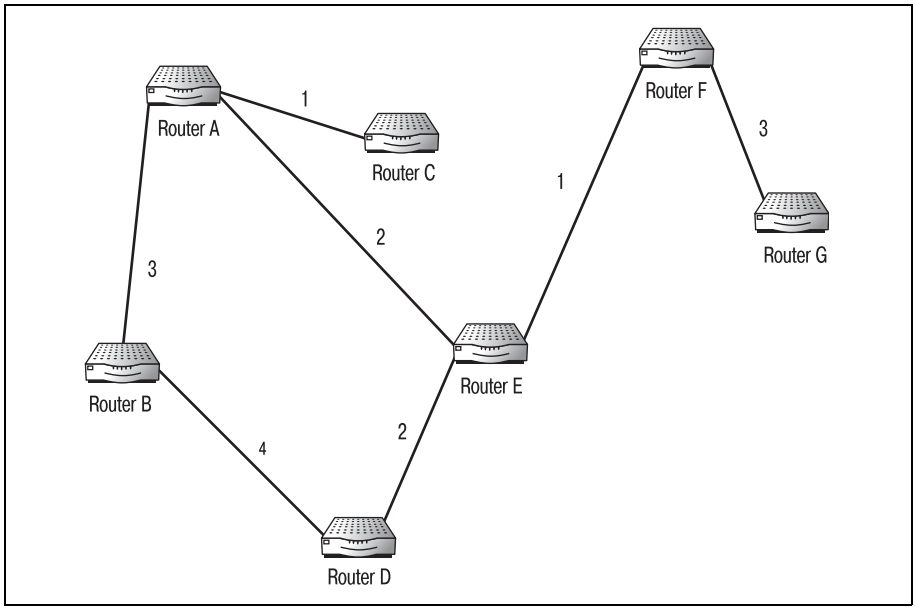

***Рис. 14.1.** Связанная сетевая среда*

Среда, изображенная на рис. 14.2, включает в себя семь маршрутизаторов с (произвольно) назначенными метриками. Так может выглядеть сеть типичной конфигурации. В табл. 14.2–14.8 приведены таблицы маршрутов для каждого из маршрутизаторов в том виде, в каком они существуют до начала каких-либо обновлений.

Таблица 14.2. Таблица маршрутов маршрутизатора A

Путь	Метрика
A–C	1
A–B	3
A–E	2

Таблица 14.3. Таблица маршрутов маршрутизатора B

Путь	Метрика
B–A	3
B–D	4

Таблица 14.4. Таблица маршрутов маршрутизатора C

Путь	Метрика
C–A	1

Таблица 14.5. Таблица маршрутов маршрутизатора D

Путь	Метрика
D–B	4
D–E	2

Таблица 14.6. Таблица маршрутов маршрутизатора E

Путь	Метрика
E–A	2
E–D	2
E–F	1

Таблица 14.7. Таблица маршрутов маршрутизатора F

Путь	Метрика
F–E	1
F–G	3

Таблица 14.8. Таблица маршрутов маршрутизатора G

Путь	Метрика
G–F	3

Из этих таблиц видно, что не каждый маршрутизатор знает пути ко всем остальным. На самом деле каждый из них видит только своих непосредственных соседей. Если маршрутизатор A получает пакет, предназначенный маршрутизатору F, он не может определить, какой из маршрутов предпочтительнее. (Если бы маршрутизатор A использовал протокол вектора расстояния, то ему был бы известен полный путь к маршрутизатору F. Если же маршрутизатор A использует протокол состояния канала, то у него не будет никакого способа узнать, существуют ли еще работающие каналы связи кроме его собственных.) Каждый маршрутизатор должен сообщить остальным, как выглядит его сетевое окружение.

После полномасштабного обновления, когда каждый из маршрутизаторов проинформирует своих соседей о содержимом своей таблицы маршрутов, новая общая таблица маршрутов будет иметь вид, приведенный в табл. 14.9.

После того как обновление закончено, маршрутизатору A гораздо проще отправить пакет маршрутизатору F. Изучив свою таблицу, маршрутизатор A увидит, что маршрутизатор F находится за маршрутизатором E, который доступен как непосредственно, так и по маршруту A–B–D–E.

Таблица 14.9. Обновленная таблица маршрутов маршрутизатора A

Путь	Метрика
A–B	3
B–D	4
A–C	1
A–E	2
D–E	2
E–F	1
F–G	3

Таблицы маршрутов, подобные только что рассмотренной, могут образовываться двумя путями. Их заполнение может инициироваться событиями либо таймером через определенный интервал. Некоторые протоколы запускают процесс обновления в определенное время, например каждые 30 секунд. В этом случае, как только интервал истекает, каждый маршрутизатор посылает обновление, содержащее информацию из его таблицы. Это обновление может содержать важные сведения об изменениях в сети, а может быть и пустым.

Другие протоколы запускают процесс обновления по некоторому событию. Например, если отключается канал между маршрутизаторами A и E, то рассылается обновление, немедленно сообщающее остальным маршрутизаторам, что они не должны полагаться на это соединение.

В обоих случаях обновления приводят к одному результату: конвергенции. Конвергенция в сети достигается, когда все маршрутизаторы используют в своей работе одну и ту же картину сети. Быстрая конвергенция сети является важным требованием и существенно влияет на рыночные перспективы протокола.

Конвергенция

В том, что касается маршрутизации, конвергенция – это волшебное слово. Конвергенцией называют такое состояние, в котором все маршрутизаторы некоторой среды используют в работе одинаковое представление сети. Например, при отключении какого-то из каналов начинается обновление маршрутных таблиц. С момента отключения канала и до завершения обновления всех таблиц маршрутизаторы находятся вне состояния конвергенции. Любой пакет, пересылаемый по сети, не достигшей конвергенции, рискует быть потерянным или отправленным не по адресу.

Состояние конвергенции труднее достигается в больших сетях со сложными маршрутами. Поэтому так важен правильный выбор типа протокола, подходящего для определенной среды. В очень больших сетях протоколы состояния канала достигают конвергенции намного быстрее, чем протоколы вектора расстояния. Однако в небольших се-

тях протоколы вектора расстояния выигрывают в скорости доставки информации.

Резюме

- Работа маршрутизаторов основана на применении протоколов маршрутизации.
- Протоколы маршрутизации используют алгоритмы вычисления наилучшего пути от одной сети к другой.
- Протоколы, использующие состояние канала (link state) и вектор расстояния (distance vector), представляют собой два основных типа протоколов маршрутизации.
- Каждый из маршрутизаторов поддерживает лишь свою часть таблицы маршрутов.
- Для поддержания целостной картины всей сети маршрутизаторы используют обновления.
- Конвергенцией называется такое состояние сети, когда все маршрутизаторы используют в работе одно и то же представление сетевого окружения.

Вопросы и ответы

Вопрос Зачем нужны различные типы протоколов маршрутизации? Почему бы не создать такой протокол, который сможет работать в больших сетях без потери скорости?

Ответ Среда маршрутизации по природе своей очень изменчива. Вследствие этого невозможно предусмотреть все возможные конфигурации и условия работы. Выбор протокола, наиболее подходящего для конкретной сети, – это сложная, но важная задача, которую приходится решать практически каждому сетевому инженеру.

Тест

Вопросы

1. Протоколы какого типа лучше работают в больших маршрутизируемых сетях?
2. Каким термином описывается маршрутизатор с точки зрения среды маршрутизации?
3. Какие значения произвольно назначаются каналам, связанным с маршрутизатором?

Ответы

1. Состояние канала (Link state).
2. Переход (Hop).
3. Метрика (или стоимость).

16

Использование IGRP и EIGRP

В первые годы существования компьютерных сетей, когда они еще находились на стадии экспериментирования, RIP был королем протоколов маршрутизации. Однако по мере роста сетей возникли и новые требования к устройствам маршрутизации. Сети, достигнув установленного RIP предела в 15 переходов, требовали отмены этого ограничения, а также более быстрой и надежной конвергенции. Свою роль сыграло и начавшееся взаимодействие сетей предприятий (а при использовании RIP это было невозможно).

На горизонте уже просматривался Интернет, и для удовлетворения растущих потребностей корпоративной инфраструктуры требовались новые протоколы маршрутизации. Так как технология маршрутизации тогда лишь зарождалась, новые протоколы должны были быть столь же понятны и легки в настройке и сопровождении, как RIP. Но технология, лежавшая в основе RIP, не могла справиться с обработкой растущего трафика.

С появлением Интернета потребность в протоколе маршрутизации, более совершенном, чем RIP, стала еще более очевидной. В середине 80-х компания Cisco начала разработку нового протокола маршрутизации IGRP (Interior Gateway Routing Protocol – протокол маршрутизации внутреннего шлюза). Этот протокол расширял возможности технологии, положенной в основу RIP, переходя на следующий уровень работы с сетью – использование шлюза. Вскоре после выхода в свет IGRP появился его наследник, EIGRP (Enhanced Interior Gateway Routing Protocol – усовершенствованный протокол маршрутизации внутреннего шлюза). Вместе эти два протокола стали ключевым фактором в междоменной маршрутизации.

В этой главе вы познакомитесь с реализацией IGRP и EIGRP, выполненной Cisco. Мы рассмотрим технологии, лежащие в основе этих протоколов, и команды для их конфигурирования на маршрутизаторах Cisco. В этой главе обсуждаются следующие темы:

- Сравнение IGRP и EIGRP с RIP
- Технология IGRP
- Конфигурирование IGRP
- Технология EIGRP
- Конфигурирование EIGRP

Протоколы IGRP и EIGRP имеют определенную специализацию в маршрутизируемой среде. Если их предшественник RIP предназначался для работы в единой ограниченной сети, то IGRP и EIGRP используются при объединении больших сетей, известных как автономные системы. Ключом к изучению функционирования IGRP и EIGRP является понимание их отличия от RIP.

Оба рассматриваемых в этой главе протокола внутреннего шлюза являются производными от RIP, однако сходство этих трех протоколов не всегда очевидно. На первый взгляд может показаться, что единственное их сходство заключается в использовании алгоритма вектора расстояния Беллмана-Форда (Bellman-Ford). Однако проведя небольшое исследование, мы сможем обнаружить и другие общие элементы у этих протоколов.

Изучение сходства и различий всех трех протоколов поможет вам лучше понять, как работает каждый из них. В целом, эти протоколы не полностью взаимозаменяемы, каждый из них имеет собственную специализацию в области маршрутизации. Там, где работает один, не всегда может быть использован другой. Именно так обстоит дело с RIP, IGRP и EIGRP.

IGRP и EIGRP в сравнении с RIP

Из предыдущей главы вы должны помнить, что RIP – это протокол внутреннего шлюза (IGP – interior gateway protocol). Поэтому назначение RIP состоит в перемещении данных в пределах одного сетевого окружения. То есть лежащая в его основе технология не рассчитана на то, чтобы заниматься адресацией и другими задачами маршрутизации данных между множественными средами.

RIP проектировался так, чтобы быть простым, быстрым и функциональным, поэтому авторы были вынуждены пойти на некоторые компромиссы. Они решили уменьшить количество дополнительных параметров и возможностей для протокола. В результате получился достаточно простой, быстрый и удобный в сопровождении протокол. Но в нем не хватало многих свойств, которые администраторы впоследствии стали ожидать от протоколов маршрутизации.

Практически невозможно создать протокол, который бы удовлетворял всем потребностям всех пользователей. Время шло, и администраторам нужны были протоколы, которые бы развивались вместе с ними и их сетями. Cisco (вместе с несколькими ведущими организациями) оперативно начала работу над IGRP. Этот протокол должен был принять эстафету у RIP.

IGRP все еще сохраняет принадлежность к классу IGP; но он работает совсем в другом масштабе, чем RIP, организуя прозрачное соединение между устройствами, принадлежащими разным средам. Такие протоколы внутреннего шлюза, как RIP, IGRP и EIGRP, работают в сетевых средах, не имея возможности перемещать данные между ними. Для обеспечения маршрутизации в больших многосетевых средах необходимо, чтобы рука об руку работали протоколы внутренних и внешних шлюзов. IGRP и позже EIGRP внесли в словарь сетевых инженеров и специалистов по маршрутизации несколько новых терминов. Большая среда общего назначения стала называться автономной системой (AC). AC можно рассматривать как набор сетей или сред с общим элементом маршрутизации. Такие среды для обеспечения удобства адресации группировались в отдельные категории. К автономным системам можно обращаться как к самостоятельным объектам, адресуя данные целой сети.

Еще одним термином, который следовало ввести в словарь маршрутизации, стал *шлюз*. Шлюзы – это устройства, которые направляют информацию, поступающую из некоторой сети (с определенным адресом), к устройствам, принадлежащим другой среде, не входящей в исходную сеть. Другими словами, шлюзы – это обычные маршрутизаторы, которые находятся на границе сети и направляют данные в эту сеть и из нее. Сетевая среда со шлюзом изображена на рис. 16.1.

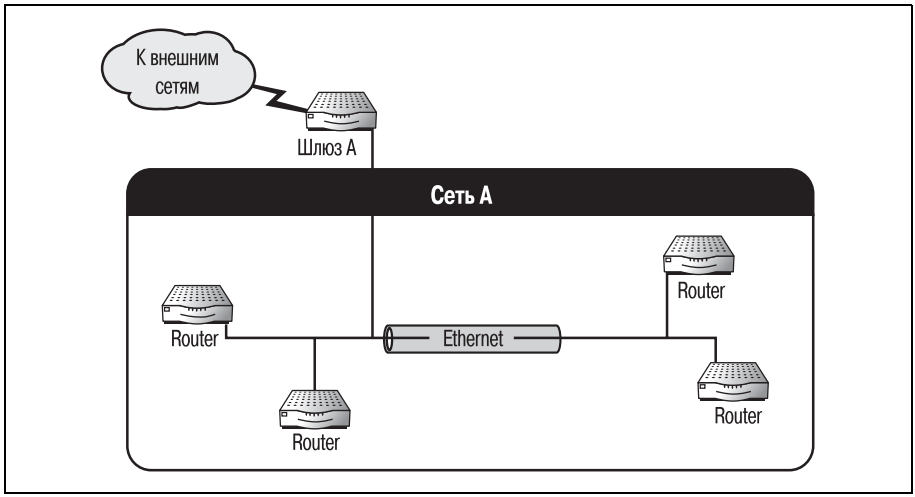

Рис. 16.1. *Пример шлюза*

Из следующих глав вы узнаете, что, будучи концептуальными устройствами, шлюзы чрезвычайно важны при выборе протоколов маршрутизации. Использование шлюзов (или взаимодействие с ними) определяет работу многих протоколов маршрутизации. Поэтому лучше будет поговорить о том, что же такое шлюз.

Сегодня наибольшее распространение имеют те шлюзы, которые связывают компании с Интернетом. По существу, Интернет – это огромная маршрутизируемая среда. Поэтому любой маршрутизатор, осуществляющий связь локальной среды с Интернетом, является шлюзом. Сделаем шаг назад, чтобы отойти от глобального определения шлюза и рассмотреть его на более простом уровне.

Упрощенная многосетевая среда состоит из двух сетей (в каждой из которых есть некоторое количество маршрутизаторов) и двух шлюзов. (Две сети могут находиться в одном здании или на разных концах страны, в любом случае шлюзы, находящиеся с каждой стороны среды, будут соединять их.) Маршрутизаторы каждой сети определяют пути для данных внутри сети, повторяя процедуру, изложенную в предыдущих главах. Когда маршрутизаторы одной из сетей обнаруживают пакет, для которого у них нет адресной информации, он пересылается шлюзу. Шлюз исследует пакет и пересылает его соответствующему шлюзу другой сети. Затем пакет передается маршрутизаторам, находящимся внутри второй сети, и процесс продолжается.

Среды, соединенные при помощи шлюзов, можно разделить на две составляющие: внешнюю и внутреннюю. Та часть среды, которая содержит сеть (и маршрутизаторы, находящиеся внутри сети), считается внутренней, а часть, соединяющая шлюзы, – внешней. Для каждой из компонент, как внешней, так и внутренней, существуют свои протоколы маршрутизации. Такие протоколы внутреннего шлюза, как RIP, IGRP и EIGRP, используются для маршрутизации информации во внутренних частях среды, а протоколы внешнего шлюза (EGP, Exterior Gateway Protocol) – это магистральные протоколы, соединяющие один шлюз с другим.

Усовершенствованный (Enhanced) IGRP, или EIGRP, стал дальнейшим развитием IGRP. В то время как в IGRP улучшения коснулись таких базовых характеристик, как количество разрешенных переходов и набор метрик, в EIGRP значительно улучшены вычислительные алгоритмы и уменьшено время конвергенции. В следующих разделах обсуждается технология, лежащая в основе протоколов IGRP и EIGRP, и рассматриваются команды для их конфигурирования на маршрутизаторах Cisco.

Технология IGRP

Протокол маршрутизации внутреннего шлюза IGRP был создан на основе технологии, ранее использованной в протоколе RIP. Большая

часть базовых компонентов сохранилась, но их возможности существенно возросли. В этом разделе рассматриваются те элементы, которые подверглись усовершенствованию в процессе проектирования IGRP.

Основные элементы RIP, усовершенствованные в IGRP, – это метрики, используемые при расчете расстояний, допустимое количество переходов и технология выравнивания нагрузки для маршрутов с разными метриками (unequal-cost load balancing). Разработчики Cisco, работавшие над спецификациями IGRP, имели преимущество, которого не было у авторов RIP: они проектировали протокол для определенных устройств. В то время как RIP был адаптирован для работы на маршрутизаторах, IGRP создавался непосредственно для них. Поэтому отдельные компоненты были модифицированы так, чтобы они могли использовать возможности новой аппаратной платформы.

Небольшие изменения коснулись также способов пересылки и обработки обновлений для таблиц маршрутов. Ниже мы рассмотрим, как изменились обновления таблиц в IGRP по сравнению с RIP.

Метрики IGRP

Протокол IGRP использует метрики в алгоритмах расчета наилучшего пути от одной сети к другой. Как вы помните, RIP использует только одну метрику: счетчик переходов. В протоколе IGRP счетчик переходов заменен более информативными и точными переменными. Мы рассмотрим каждую из метрик с точки зрения расчета маршрутов в этом протоколе. В IGRP используются следующие метрики:

- Межсетевая задержка
- Пропускная способность
- Нагрузка
- Надежность

Алгоритм вектора расстояния Беллмана-Форда, используемый RIP, применяется и в IGRP, изменились лишь параметры, используемые при расчете. Хотя счетчик переходов еще используется в тех случаях, когда вся сеть построена на едином носителе (вследствие чего уравниваются значения остальных метрик), были добавлены и новые метрики. Протокол IGRP учитывает задержку, пропускную способность, нагрузку и надежность при определении наилучшего пути для отправки данных.

Межсетевая задержка характеризует время, за которое пакет, отправленный одним маршрутизатором, достигает пункта назначения (другого маршрутизатора). Значение времени задержки может быть выражено числом в диапазоне от 1 до 16 777 216 (в десятках микросекунд). Задержка, по сравнению с другими метриками, представляет собой наиболее отвлеченное понятие. Может существовать множество причин, по которым задержка в некотором канале может быть больше, чем в

другом. Одна из возможных причин – значительный трафик. Величина задержки определяется целым рядом факторов, и возможность ее регулирования позволяет существенно влиять на производительность сети.

Пропускная способность – это фактическая скорость передачи данных в канале, находящаяся в диапазоне от 1200 бит/с до 10 Гбит/с. Понятие пропускной способности не столь абстрактно, как межсетевая задержка. Большинство администраторов имеют четкое представление о пропускной способности своих каналов. Если какой-либо из каналов имеет невысокую пропускную способность, то его шансы быть выбранным в качестве наилучшего маршрута невелики.

Нагрузка канала определяется значением в диапазоне от 1 до 255, выражающим степень интенсивности его использования. Если канал сильно загружен, значение этой метрики увеличивается, повышая таким образом стоимость данного маршрута. Несколько иное значение имеет параметр балансировки нагрузки, который управляет распределением адресованных некоторому узлу пакетов между несколькими доступными путями (в соответствии с их стоимостями) так, чтобы ни один из них не простаивал и не испытывал перегрузки.

Последняя метрика – надежность – может принимать значение от 0 до 255. Надежность канала характеризует то, как часто теряются передаваемые по нему пакеты. Значение метрики, равное 255, говорит о том, что канал надежен на 100% и никогда не теряет пакеты.

Все метрики, имеющиеся в IGRP, используются совместно и позволяют более точно провести вычисления наилучшего пути для передачи информации от одной сети к другой. Каждому каналу маршрутизатора, работающего с IGRP, присваиваются такие метрики. Следовательно, протокол IGRP нуждается в месте для хранения расширенного набора метрик, используемых в расчете маршрутов. Поэтому таблица маршрутов этого протокола была расширена по сравнению с имевшейся в RIP. Таблица маршрутов IGRP имеет дополнительные поля для записи значений задержки, пропускной способности, нагрузки и надежности.

Увеличение разрешенного количества переходов

Разработчики IGRP сделали еще одно усовершенствование в базовой технологии RIP: они увеличили предельное количество переходов. С целью уменьшения количества непредусмотренных петель маршрутизации и сохранения максимального объема памяти для вычислений создатели RIP предусмотрели ограничение в 15 переходов. К моменту начала работы Cisco над протоколом IGRP было уже очевидно, что такое ограничение совершенно недостаточно для поддержки растущих сетей.

Максимальное количество переходов было увеличено в IGRP с 15 до 255. Это существенно расширило возможности его применения для

маршрутизации. Протокол IGRP легко справлялся с маршрутизацией в сетях, слишком больших для RIP.

Примечание

Не путайте переход (hop) с узлом (node). Термином «переход» обозначается маршрутизатор (либо компьютер с установленной программой маршрутизации), в то время как «узлом» называют любое устройство, присутствующее в сети.

Выравнивание нагрузки

Последним и наиболее значимым усовершенствованием технологии RIP стала способность IGRP распределять нагрузку между каналами с различной стоимостью. То есть протокол IGRP может быть сконфигурирован таким образом, чтобы посылать данные по нескольким каналам одновременно в соответствии с их текущей загруженностью.

Большинство протоколов маршрутизации при передаче пакетов автоматически переходят с одного канала на другой с тем же значением метрики. В отличие от них, IGRP способен распределять нагрузку между каналами, имеющими разные значения метрик. Рисунок 16.2 иллюстрирует ситуацию, в которой может потребоваться балансировка нагрузки в каналах с разными стоимостями.

На рис. 16.2 маршрутизатор A имеет два возможных пути к маршрутизатору E. Предпочтение всегда будет отдаваться пути A–B–E благодаря тому, что он имеет лучшее суммарное значение метрик, чем путь A–D–F–E. Однако это может создать проблемы. По сути дела, в сети оказываются два пути, один из которых перегружен, а второй никогда не используется.

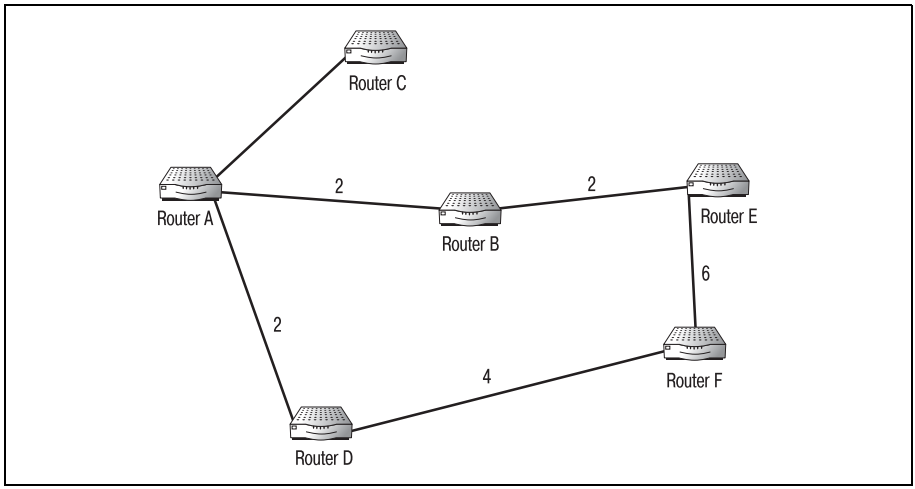

Рис. 16.2. *Пример выравнивания нагрузки для маршрутов с разными стоимостями*

Балансировка нагрузки предлагает решение в тех ситуациях, когда путь оказывается перегруженным только из-за того, что имеет наилучшее значение метрики. При конфигурировании протокола IGRP ему можно сказать: «Оба пути, A–B–E и A–D–F–E, ведут к маршрутизатору E и должны быть использованы поочередно при отправке данных от маршрутизатора A к маршрутизатору E».

В процессе конфигурирования балансировки нагрузки один из каналов назначается кратным другому. То есть путь A должен использоваться в *x* раз чаще, чем путь B. В этом случае каждая группа из *x* пакетов распределяется между несколькими каналами с учетом баланса нагрузки.

Примечание

Протокол IGRP допускает использование до четырех каналов в одном сбалансированном маршруте.

Обновления IGRP

Обновления таблицы маршрутов IGRP содержат в основном ту же информацию, что и в случае RIP. То есть за исключением новых полей, предназначенных для метрик IGRP, информация в обновлениях осталась той же. Маршрутизаторы IGRP, как и RIP-маршрутизаторы, периодически посылают локальные обновления своих таблиц маршрутов всем непосредственно связанным с ними соседям. Но расписание отправки обновлений в IGRP существенно отличается от применяемого в RIP. Помимо этого, добавлено обновление нового типа: обновление с мгновенной рассылкой (flash update).

Обновления IGRP отправляются каждые 90 секунд. Это значительное увеличение по сравнению с 30-секундным интервалом в RIP. Причина этого проста: IGRP может управлять значительно более крупными сетями, чем RIP, и поэтому должен предоставить маршрутизаторам больше времени на выполнение корректировок с тем, чтобы следующее обновление не началось раньше, чем закончится предыдущее. Вследствие увеличения паузы между обновлениями время отмены неиспользуемых маршрутов также увеличилось.

Если обновление, предназначенное некоторому маршрутизатору, не получено им в течение 270 секунд, данный маршрут помечается как неактивный. Неактивные маршруты не используются, но сохраняются в таблице маршрутов для последующего использования. Если обновление не получено в течение 630 секунд, маршрут удаляется из таблицы.

Обновления маршрутов IGRP имеют больший размер (из-за дополнительных полей) и занимают больше времени (из-за большего количества переходов). По этой причине разработчики IGRP решили увеличить интервалы времени между ними.

Новый тип обновлений IGRP, отсутствующий в RIP, – это обновления с мгновенной рассылкой. Обновления маршрутов отправляются в IGRP каждые 90 секунд, однако некоторые из метрик этого протокола требуют обновления в реальном масштабе времени. Поэтому был добавлен механизм, позволяющий протоколу извещать соседей в случае значительных изменений метрик.

При изменении метрик IGRP может инициировать мгновенное обновление, в процессе которого новые значения метрик некоторого маршрута пересылаются соседям. Мгновенные обновления позволяют поддерживать таблицы маршрутов IGRP в актуальном состоянии, не перегружая сеть полными обновлениями.

В связи с обновлениями IGRP следует упомянуть те элементы, которые не претерпели изменений при переходе от RIP к IGRP. Способы исключения петель, имевшиеся в RIP, такие как расщепление горизонта, таймеры временного удерживания изменений и отмена маршрута, используются также и в IGRP.

Конфигурирование IGRP

Разработчики протокола IGRP придерживались того правила, что протокол, оставаясь простым в настройке, должен предоставить администратору полный контроль над маршрутизацией. Процесс конфигурирования IGRP очень прост, и в то же время Cisco предоставляет большой выбор дополнительных параметров для его тонкой настройки. Такие возможности, как балансировка нагрузки (параметры которой будут рассмотрены ниже), делают IGRP очень гибким протоколом маршрутизации.

Как и в случае с RIP (и со всеми прочими протоколами маршрутизации), вы должны конфигурировать IGRP в режиме конфигурирования маршрутизатора Cisco IOS. Первый параметр, который IOS предложит вам ввести при входе в режим глобального конфигурирования, это номер автономной системы (ASN – Autonomous System Number), используемый для ее идентификации. Допустимо любое значение в диапазоне от 1 до 65 535. При отсутствии специальных требований можете использовать любое подходящее на ваш взгляд значение. Синтаксис команды, с которой начинается конфигурирование IGRP, таков:

```
#router igrp <asn>
```

Примечание

Некоторые протоколы, в частности BGP, используют значения ASN, выдаваемые центральным органом, аналогично тому, как выдаются IP-адреса. Вам следует всегда сверяться с официальным реестром, прежде чем назначать ASN.

После того как вы назначили номер автономной системы, следует указать адрес сети. Адрес сети – это корректный IP-адрес сети конфигури-

руемого маршрутизатора. Этот адрес используется во всех обновлениях таблицы маршрутов. Команда network имеет формат

```
#network <ip address>
```

Значения ASN и адреса сети – это все необходимые параметры при конфигурировании IGRP. Следующая последовательность команд иллюстрирует процесс активации протокола IGRP на маршрутизаторе Cisco:

```
Router#configure terminal
Router(config)#router igrp 210
Router(config-router)#network 198.10.0.0
```

Этими командами маршрутизатор успешно конфигурируется для работы в сети с протоколом IGRP. Существует, однако, ряд дополнительных параметров, которые могут быть использованы для настройки производительности IGRP. Эти дополнительные параметры отвечают за выравнивание нагрузки, установку таймеров обновления и отключение таких функций, как расщепление горизонта.

Выравнивание нагрузки в каналах с разной стоимостью

В предыдущем разделе говорилось, что выравнивание нагрузки для маршрутов с разными метриками – это процесс, в котором маршруты, имеющие разные метрики и проходящие через одинаковую пару источник-приемник, могут брать на себя часть нагрузки наиболее загруженного пути, чтобы один канал не оказывался «завален» трафиком из-за того, что у него лучшее значение метрики.

Чтобы использовать путь в качестве маршрута для выравнивания нагрузки, выполните команду variance. Вместо того чтобы указывать, какой именно путь должен использоваться для выравнивания нагрузки, вы с помощью команды variance определяете «диапазон», в который должны попадать потенциальные маршруты. Маршрутизатор Cisco использует любой путь, попавший в «вариацию», как путь для выравнивания нагрузки. Команда variance имеет следующий формат:

```
#variance <multiplier>
```

Вариация – это то, на что умножается метрика «наилучшего пути». То есть каждый путь, который рассматривается в качестве маршрута для выравнивания нагрузки, должен иметь значение суммарной метрики, не превышающее произведения вариации и метрики лучшего пути. Например, если лучший локальный путь имеет метрику 6, а вариация маршрутизатора установлена равной 2, то любой путь, который может применяться для выравнивания нагрузки, должен иметь метрику, не превышающую 12 (в том же направлении).

Еще одно правило, которое должно выполняться для того, чтобы путь рассматривался как возможный маршрут для выравнивания нагруз-

ки, заключается в том, что следующий переход от локального маршрутизатора должен иметь меньшее значение метрики. Другими словами, если маршрутизатор A имеет в направлении D метрику 4, то маршрутизатор B должен иметь метрику 3 (в том же направлении), чтобы считаться маршрутом выравнивания нагрузки.

Хотя условия создания и использования маршрутов выравнивания нагрузки могут показаться слишком жесткими, они помогают поддерживать целостность алгоритма, который делает IGRP универсальным протоколом. Единственное, что нужно сделать, чтобы настроить маршрутизатор Cisco на использование выравнивания нагрузки для маршрутов с различными стоимостями, – выполнить команду `variance`:

```
Router#configure terminal
Router(config)#router igrp 51
Router(config-router)#variance 3
Router(config-router)#^Z
```

Как видите, использовать команду `variance` несложно. Она просто задает значение, которое Cisco IOS может использовать для определения маршрутов, подходящих для выравнивания нагрузки. Благодаря тому что указывается не конкретный маршрут, а вариация, маршрутизатор может динамически адаптироваться к изменениям среды. Если маршрут, использовавшийся для выравнивания нагрузки, внезапно стал недоступен, он просто выпадает из интервала, заданного вариацией, и больше не используется. И наоборот, если создан новый маршрут, попадающий в интервал вариации, он сразу же используется для разделения трафика.

Изменение таймеров обновления

Для инициации и отслеживания некоторых событий, связанных с обновлением маршрутов, IGRP использует четыре таймера. Это таймер обновления маршрута, тайм-аут маршрута, таймер временного удерживания изменений и таймер удаления маршрута. Каждый таймер имеет значение по умолчанию. Но для того чтобы изменить некоторую среду маршрутизации, можно установить новые значения таймеров. Значения, в которые таймеры IGRP установлены по умолчанию, приведены в табл. 16.1.

Для изменения значений, в которые таймеры IGRP установлены по умолчанию в Cisco IOS, используйте команду `timers basic`. Вы можете изменить таймер, чтобы чаще получать обновления или быстрее удалять неисправные маршруты из таблицы маршрутов. Общая цель изменения какого-либо таймера IGRP заключается в достижении более быстрой и надежной конвергенции. Синтаксис изменения таймера IGRP таков (все значения времен указываются в секундах):

```
#timers basic <update timer> <timeout timer> <hold-down timer> <removal timer>
```

Таблица 16.1. Значения таймеров IGRP по умолчанию

Таймер	Значение по умолчанию
Обновление маршрута	90 секунд
Тайм-аут	270 секунд
Удаление маршрута	630 секунд
Временное удерживание изменений	Утроенное значение таймера обновления плюс 10 секунд

Вы можете изменить значения таймеров, чтобы попытаться достичь более быстрой конвергенции всей сети. Будьте осторожны при внесении изменений, ведь они могут привести и к тому, что обновления станут более длительными и менее точными. Выполните следующие команды:

```
Router#configure terminal
Router(config)#router igrp 51
Router(config-router)#timers basic 75 200 76 550
Router(config-router)#^Z
```

Таймер обновления маршрутизатора Cisco изменен с 90 секунд на 75; тайм-аут – с 270 на 200; таймер временного удерживания изменений установлен в 76; и таймер удаления маршрута изменен с 630 на 550.

Разрешение/запрещение расщепления горизонта и временного удерживания изменений

Удобным свойством IGRP является возможность разрешать и отменять различные функции, используемые маршрутизатором для предотвращения петель маршрутизации. В небольших средах, где очень мало динамических изменений, такие функции могут и не понадобиться. Может случиться и так, что вам будет необходима какая-то одна функция, но не все остальные.

Cisco имеет встроенную возможность разрешения и отмены расщепления горизонта и таймеров удерживания изменений. Их необходимо настраивать в разных режимах конфигурирования Cisco IOS. Начнем с временного удерживания изменений.

Маршрутизатор применяет таймер временного удерживания изменений для того, чтобы помечать подозрительные пути. Маршрут может быть удержан, если маршрутизатор только что узнал о нем или если он внезапно оказался неисправным. В любом случае маршрутизатор не будет использовать такой маршрут в течение того времени, на которое маршрут будет удержан. Когда это время закончится, маршрутизатор сможет пересылать пакеты по маршруту как обычно.

Примечание

Когда маршрутизатор «узнает» что-либо о маршруте, то черпает эту информацию из обновления. Другими словами, маршрутизатор не имеет прямых сведений о маршруте; он узнает о нем от других маршрутизаторов.

Чтобы отменить таймер временного удерживания изменений, используйте ключевое слово `no` в режиме конфигурирования маршрутизатора:

```
Router#configure terminal
Router(config)#router igrp 51
Router(config-router)#no metric holddown
Router(config-router)#^Z
```

Примечание

Использование команды `no metric holddown` маршрутизаторами одной среды должно быть стандартизовано. То есть все маршрутизаторы одной среды, работающие по IGRP, должны одновременно или разрешать, или же запрещать таймеры временного удерживания изменений.

Расщепление горизонта обрабатывается в Cisco IOS несколько иначе. Правило расщепления горизонта запрещает маршрутизаторам отправлять обновления тем маршрутизаторам, от которых они получены. Например, маршрутизатор A получает обновление от маршрутизатора B. В этом обновлении маршрутизатор B сообщает A, что один из каналов B не работает. Если расщепление горизонта действует, маршрутизатор A не может отправить B собственное обновление. Это уменьшает риск того, что маршрутизатор A узнает (от третьего маршрутизатора) неверную информацию о канале маршрутизатора B и передаст ее B.

Разрешение или запрет на применения правила расщепления горизонта повлияет на отправку обновлений вашим маршрутизатором только в том случае, если на маршрутизаторе несколько RIP-интерфейсов. (Если же на маршрутизаторе всего один RIP-интерфейс, то отключение расщепления ни к чему не приведет, разве что, вероятно, несколько разгрузит процессор.)

Разрешение и запрет расщепления горизонта устанавливается на уровне интерфейса. То есть Cisco предоставляет возможность использовать это свойство для отдельных интерфейсов. Чтобы запретить расщепление горизонта, необходимо указать конкретный интерфейс в режиме конфигурирования интерфейсов:

```
Router#configure terminal
Router(config)#interface ethernet 1
Router(config-int)#no ip splithorizon
Router(config-int)#^Z
```

Мы рассмотрели принципы и технологию работы IGRP. Но в середине 90-х годов компания Cisco начала работу над улучшенной версией про-

токола маршрутизации. В новой версии IGRP, получившей уместное в данном случае название EIGRP (Enhanced Interior Gateway Routing Protocol – усовершенствованный IGRP), переработаны и улучшены некоторые возможности IGRP.

Технология EIGRP

Создавая преемника IGRP, Cisco внесла в протокол ряд изменений. Первое, над чем решили поработать проектировщики, – это сокращение времени конвергенции. Но они понимали, что имеющаяся технология обновлений, разработанная для RIP, уже исчерпала все свои возможности. Для EIGRP была создана новая модификация алгоритма вектора расстояния IGRP.

DUAL (Diffusing Update Algorithm – алгоритм диффузного обновления) позволяет достичь чрезвычайно быстрой конвергенции за счет интенсивного использования процессора. Когда маршрутизатор узнает о новом пути, DUAL вычисляет путь, для того чтобы определить (с достаточной степенью уверенности), можно ли считать путь «свободным от петель». Начиная с этого момента обрабатываются и применяются только обновления, относящиеся к изменениям среды.

Объем вычислений, производимых DUAL, очень велик, поэтому маршрутизаторы EIGRP не выполняют их при каждом обновлении. Благодаря тому что маршрутизаторы IGRP не вычисляют заново каждый маршрут при каждом обновлении, значительно сокращается время конвергенции.

Примечание

Вычисление маршрута при помощи алгоритма DUAL производится только тогда, когда используемый маршрут изменяется (что очень редко случается в оптимальной среде). Это единственная задача EIGRP, требующая интенсивной работы процессора. Поэтому в обычной повседневной работе процессор маршрутизатора используется очень рационально.

Еще одним важным улучшением IGRP, сделанным в процессе разработки EIGRP, стало добавление пакетов hello. Многие протоколы маршрутизации используют для выявления соседей специальные пакеты, называемые пакетами hello. Обычно hello-пакеты не зависят от протокола и используются для контроля над динамическими средами маршрутизации.

Маршрутизаторы EIGRP отправляют своим прямым соседям hello-пакеты (или сообщения hello) через установленные промежутки времени. Смысл таких сообщений в том, чтобы определить корректность локальной таблицы маршрутов. Это легко выполнить, так как hello-пакеты являются специальной разновидностью широковещательных пакетов.

Примечание

Широковещательные сообщения – это пакеты, которые не предназначены какому-то одному адресату. Они обрабатываются каждым устройством, которого они достигают.

Локальный маршрутизатор периодически отправляет широковещательный пакет. Каждый маршрутизатор, непосредственно связанный с локальным, отвечает на hello-пакет подтверждением. Локальный маршрутизатор сравнивает свою таблицу маршрутов с перечнем подтверждений. Если маршрутизатор обнаруживает какие-либо отличия, создается обновление с мгновенной рассылкой.

Конфигурирование EIGRP

Команды Cisco IOS, применяемые для конфигурирования EIGRP, практически идентичны командам, использованным для IGRP. Это логично: ни одна из управляемых командами настроек не изменилась при переходе от одного протокола к другому. Большая часть изменений была сделана «под капотом». Хотя некоторые необязательные параметры и изменились, но основные команды, использованные для разрешения работы IGRP, сохраняются и в EIGRP.

Ниже приведены команды, которые необходимо выполнить для того, чтобы разрешить использование EIGRP на маршрутизаторе Cisco:

```
Router#configure terminal
Router(config)#router eigrp 578
Router(config-router)#network 10.0.0.0
Router(config-router)#^Z
```

Маршрутизаторам EIGRP также необходимо указание сетевого адреса и номера автономной системы. Эти параметры используются как IGRP, так и EIGRP. Иначе говоря, для разрешения работы IGRP и EIGRP на маршрутизаторе Cisco используется одна и та же базовая командная структура.

Необязательный параметр, который можно установить в EIGRP, указывает, какую пропускную способность EIGRP может использовать для определенной линии. По умолчанию EIGRP использует 50% доступной пропускной способности каждой линии маршрута. Cisco предоставляет возможность изменить процентное соотношение на уровне интерфейса.

Если вы знаете, что какой-то канал имеет очень низкую пропускную способность (так что потеря еще половины означала бы невозможность использования канала), то можете скорректировать значение используемой пропускной способности в режиме конфигурирования интерфейсов, применив такую команду:

```
#ip bandwidth-percent eigrp <percentage>
```

В режиме конфигурирования определенного интерфейса выполните команду `ip bandwidth-percentage`, в которой укажите ту величину пропускной способности канала, которая может использоваться. После этого маршрутизатор разрешит EIGRP использовать пропускную способность только в заданном объеме.

```
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-int)#ip bandwidth-percent eigrp 25
Router(config-int)#^Z
```

Из этого очень краткого описания команд EIGRP видно, что основные различия между EIGRP и IGRP лежат не на поверхности. Большая часть команд и принципов, применяемых для конфигурирования одного протокола, подходят и для другого.

Резюме

- IGRP был разработан для расширения возможностей RIP.
- Количество переходов в IGRP ограничено 255.
- Среды маршрутизации IGRP и EIGRP называются автономными системами.
- Шлюзы служат границами автономных систем.
- IGRP и EIGRP осуществляют выравнивание нагрузки для каналов с различными метриками.
- Вместо того чтобы обрабатывать все обновления при каждом поступлении, маршрутизаторы IGRP и EIGRP обрабатывают только обновления, противоречащие локальной таблице маршрутов.
- EIGRP использует hello-пакеты для проверки существования соседних маршрутизаторов.

Вопросы и ответы

Вопрос Зачем был разработан EIGRP, если IGRP уже был улучшением RIP?

Ответ Ко времени создания EIGRP технология уже сделала большой шаг вперед и появилась возможность проектирования более быстрой и надежной версии IGRP. В EIGRP настолько сокращено время достижения конвергенции сети, что протокол был быстро принят компаниями, которые ранее использовали IGRP.

Тест

Вопросы

1. Какие два номера необходимо указать при конфигурировании как IGRP, так и EIGRP?
2. Что такое DUAL?
3. Истинно или ложно такое утверждение: чтобы маршрут мог использоваться для выравнивания нагрузки, его метрика должна быть равна произведению локальной метрики и вариации?
4. Какие три метода используются IGRP для решения проблемы зацикливания маршрутов?

Ответы

1. ASN (номер автономной системы) и адрес сети.
2. Алгоритм, используемый EIGRP для вычисления маршрутов, свободных от зацикливания.
3. Ложно. Метрика пути должна быть меньше произведения локальной метрики и вариации.
4. Расщепление горизонта, таймеры временного удерживания изменений и отмена маршрутов.

Упражнение

1. Сконфигурируйте маршрутизатор автономной системы 20 (сеть 198.32.98.0) для работы с IGRP. Необходимо запретить расщепление горизонта для интерфейса Ethernet 0 и изменить значение таймера обновления, установив его в 25 секунд.

Решение

```
Router#configure terminal
Router(config)#router igrp 20
Router(config-router)#network 198.32.98.0
Router(config-router)#timers basic 25 270 90 630
Router(config-int)#^Z
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-int)#no ip splithorizon
Router(config-int)#^Z
```

III

Более сложные аспекты маршрутизации Cisco

15

Конфигурирование RIP

К этому моменту нами рассмотрены все темы, понимание которых необходимо для изучения маршрутизации Cisco. Мы говорили об оборудовании Cisco, о Cisco IOS, простых маршрутизируемых протоколах и WAN-протоколах. Создана база знаний, которая необходима любому будущему специалисту по маршрутизации.

Итак, основные вопросы обсуждены, и пришло время перейти к более специализированным вещам. Некоторые темы и технологии нуждаются в более тщательном рассмотрении, и одна из таких технологий – это протоколы маршрутизации. В этой главе вы познакомитесь с первым из протоколов маршрутизации – RIP. Глава состоит из следующих разделов:

- Общее представление о RIP
- Алгоритм маршрутизации RIP
- Работа RIP
- Конфигурирование RIP
- Поддержка RIP

Общее представление о протоколе RIP

Протоколы маршрутизации – это основа самого процесса маршрутизации. Протокол маршрутизации отвечает за успешную передачу данных из одного места сети в другое по наилучшему пути. Поэтому инженеры обычно тратят на проектирование и реализацию протоколов маршрутизации больше времени, чем на любые другие работы, связанные с текущим обслуживанием, но это, в общем, понятно, ведь именно на протоколах маршрутизации лежит огромная ответственность.

RIP (Routing Information Protocol – протокол маршрутной информации) был одним из первых протоколов маршрутизации, используемых в крупномасштабных средах. RIP был впервые выпущен в 1982 г. для Unix-сред. Если же проследить происхождение технологии, лежащей в основе RIP, то можно обнаружить протокол Xerox под названием PUP (PARC Universal Protocol) GWINFO. RIP как протокол маршрутизации относится к разновидности IGP (Interior Gateway Protocol, протокол внутреннего шлюза). Другими словами, RIP создан для работы внутри одной однородной среды.

Протоколы маршрутизации можно разделить на IGP (interior gateway protocols – протоколы внутреннего шлюза) и EGP (exterior gateway protocols – протоколы внешнего шлюза). К IGP относятся протоколы, которые перемещают данные в пределах одной определенной среды. IGP-протоколы обычно не обладают возможностью адресации, обеспечивающей связь между различными средами.

EGP – это протоколы, объединяющие несколько несвязанных сетевых сред. Слабо связанные сети, как те, что образуют Интернет, связаны внешними шлюзовыми протоколами.

Хотя одна среда может состоять из нескольких сетей, RIP не может быть использован для соединения нескольких сред. Однако определение изолированной (single) однородной (многосетевой) среды достаточно расплывчато и зависит от конкретного используемого протокола. Хорошим практическим способом распознавания того, подходит ли изолированная среда маршрутизации для работы RIP, является проверка на наличие WAN-соединения. Обычно в такой среде отсутствует WAN-соединение, использующее сеть общего доступа, такую как PSN (public switched network, коммутируемая сеть общего пользования). Если ваша среда охватывает два или более WAN-соединения, то она может не подходить для RIP.

Так как RIP проектировался тогда, когда среды маршрутизации были не такими большими, как сегодня, он не может работать в более сложных сетях, которые достаточно часто встречаются на предприятиях. Но как протокол, предназначенный для сред небольшого и среднего размеров, RIP остается чрезвычайно популярным и в наши дни.

Одна из причин продолжающейся популярности RIP возникла абсолютно случайно. Когда RIP внедрялся, его не позиционировали как протокол маршрутизации для небольших и средних сетей. Он задумывался для поддержания наиболее сложных из современных сетевых технологий. Но технологии продолжали совершенствоваться, сети становились все больше, а RIP остался позади. Однако с ростом сетей инженеры обнаружили, что очень мало какие протоколы работали в небольших средах лучше, чем RIP.

Технология, лежащая в основе RIP

RIP был разработан в начале 80-х годов компанией Xerox – одной из передовых компаний в области сетевых технологий. Изначально про-

токол создавался для работы на ПК, Unix-серверах и других вычислительных устройствах. При помощи ранних версий RIP такие устройства могли быть объединены в единую сеть, наподобие сети с топологией типа «шина».

Примечание

Сеть с шинной топологией – это простая форма сети, состоящая из компьютеров, подключенных к одной магистрали. Редко используемые сейчас, сети с шинной топологией были популярны в условиях небольшого количества устройств и ограниченного пространства или ресурсов.

Так как RIP создавался для компьютеров, а не для маршрутизаторов, он спроектирован так, чтобы имелась возможность разделения процессорного времени. Другими словами, процессор, который обрабатывает запросы от протокола, должен также обрабатывать запросы от других программ. Это значит, что RIP должен был быть хорошо оптимизирован, чтобы не занимать все доступное процессорное время.

Кроме того, компьютеры, на которых работал RIP, имели очень ограниченные ресурсы (память и дисковое пространство). Чтобы обойти эти ограничения, RIP должен был быть очень простым. RIP требует гораздо меньше памяти, чем современные протоколы маршрутизации. Это быстрый протокол, не требующий больших издержек.

Первые проектировщики RIP также попытались встроить в него некоторые функции обеспечения безопасности, чтобы защитить сети от возможных проблем маршрутизации. Одной из таких распространенных проблем является петля маршрутизации. Петля маршрутизации возникает тогда, когда устройства соединены так, что протоколы, запутавшись, продолжают бесконечно ходить по кругу. Простая петля маршрутизации изображена на рис. 15.1.

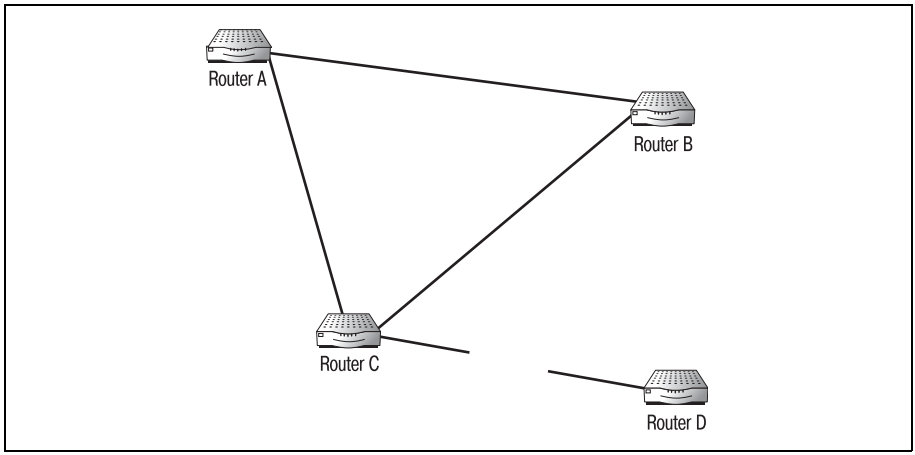

Рис. 15.1. *Простая петля маршрутизации*

Как показано на рис. 15.1, если устройство A посылает пакет устройству D, то пакет попадает в петлю маршрутизации. Канал связи с D стал недоступен, но каждое устройство думает, что D доступно с остальных устройств. Пакет перемещается от A к B, затем к C в поисках адресата. Устройство C (не имеющее доступа к D) пересылает пакет на следующий маршрутизатор (устройство A). Цикл продолжается бесконечно.

Чтобы исправить положение, первые разработчики RIP вставили в протокол ограничитель: максимальное количество переходов, при превышении которого пакет помечается как недоставленный и отбрасывается. Это свойство RIP называется «ограничение количества переходов». Оно входит в число четырех технологических усовершенствований маршрутизации, появившихся в RIP. Перечислим эти четыре новые технологии:

- Ограничение количества переходов
- Отмена маршрута
- Расщепление горизонта
- Временное удерживание изменений

Каждый из элементов в той или иной форме используется в большинстве нынешних протоколов маршрутизации. Специалистам необходимо разобраться в них, так как эти технологии могут значительно повлиять на весь процесс маршрутизации. Освоение этих широко используемых принципов поможет вам и в дальнейшем при поиске и устранении ошибок. Функциональность протокола и способы работы с ним во многом определяются перечисленными функциями, поэтому мы не пожалеем времени и поговорим о каждой из них. Но прежде чем приступить к этому разговору, необходимо познакомиться с таблицей маршрутов RIP, от которой зависят все вышеназванные функции.

Таблица маршрутов RIP

Протокол маршрутизации порождает таблицу маршрутов, в которой хранится вся информация, необходимая маршрутизатору Cisco для перемещения данных. Обычно такая таблица находится в ОЗУ маршрутизатора (а не во флэш-памяти). Благодаря этому маршрутизатор может быстро получить доступ к таблице и (при необходимости) сделать изменения.

Примечание

Помните, что ОЗУ маршрутизатора Cisco является энергозависимым, то есть его содержимое теряется при выключении устройства. Такие файлы, как `running-config`, также хранятся в ОЗУ. Если маршрутизатор Cisco выключается (вне зависимости от протокола маршрутизации), таблица маршрутов теряется.

В отличие от файла `running-config`, таблица маршрутов является динамическим файлом, который нельзя скопировать во флэш-память для того, чтобы сохранить.

Содержимое таблицы маршрутов зависит от используемого протокола маршрутизации. Типичная таблица маршрутов маршрутизатора Cisco, работающего с RIP, представлена в табл. 15.1 (таблица принадлежит маршрутизатору D):

Таблица 15.1. Пример таблицы маршрутов

Сеть	Следующий переход	Метрика	Таймер	Флаги
153.19.88.0	Маршрутизатор A	2	30–180–240	
198.63.35.0	Маршрутизатор B	6	30–180–240	
153.19.89.0	Маршрутизатор C	1	30–180–240	

Примечание

Столбец «Флаги» в табл. 15.1 необязателен. Как правило, в поле «флаг» не записано никакой информации. Данные, которые могут быть помещены в это поле, зависят от протокола, так как для каждого протокола можно определить какие-то свои параметры.

Таблица хранит информацию, необходимую для принятия решений о маршрутизации. Ее поля описаны в приведенном ниже списке:

- *Сеть*: адрес сети назначения.
- *Следующий переход*: адрес маршрутизатора, который является следующим звеном при перемещении к адресату.
- *Стоимость*: также называется метрикой, представляет количество переходов после «следующего», необходимых для достижения адресата.
- *Таймер*: поле на самом деле представляет три разных таймера, используемых RIP. Таймер обновления маршрутов (routing update timer) указывает интервал между обновлениями. Обычно RIP отсылает копию своей таблицы маршрутов каждые 30 секунд. Второй таймер – это тайм-аут для маршрута (route timeout). Если от какой-то конкретной сети обновление маршрутов не получено в течение 180 секунд, то маршрут помечается как недостижимый. Последний таймер – таймер удаления маршрута (route removal timer). Таймер удаления маршрута удаляет из таблицы маршрутов любой маршрут, который не обновлялся в течение последних 240 секунд.
- *Флаги*: в поле хранятся различные необязательные данные RIP (используется нечасто).

Из таблицы маршрутов, приведенной в табл. 15.1, можно узнать, что маршрутизатор D находится в двух переходах от сети 153.19.88.0 через маршрутизатор A, в 6 переходах от сети 198.63.35.0 через маршрутизатор B и в одном переходе от сети 153.19.89.0 через маршрутизатор C. Когда маршрутизатор D получает пакет, адресованный сети 153.19.88.0, он использует RIP для просмотра таблицы маршрутов и

приходит к выводу о том, что пакет следует отправить маршрутизатору A.

Переходы – это важное понятие в маршрутизации. Каждый маршрутизатор, через который пакет данных должен пройти, прежде чем достичь своего адресата, называется переходом. Количество переходов между источником и адресатом существенно влияет на следующие элементы маршрутизации:

- ***Истечение TTL (Time to Live – время жизни)***
- ***Искажение сигнала. (Как в старой детской игре «испорченный телефон»: чем дольше передается сообщение, тем больше вероятность его искажения.)***
- ***Ограничение размера сети. Многие протоколы маршрутизации накладывают ограничение на размер сети (в переходах).***

Маршрутизаторы Cisco, использующие RIP, отправляют соседним устройствам копии своих таблиц маршрутов каждый раз по истечении таймера обновления маршрутов. Так маршрутизаторы остаются в курсе изменений топологии сети.

Примечание

Сосед маршрутизатора – это любое соединенное с ним устройство. Другими словами, два непосредственно связанных друг с другом маршрутизатора называются соседями. Если же маршрутизаторы разделены третьим маршрутизатором (или другим устройством, обеспечивающим соединение), они не считаются соседями.

Смысл обновлений (корректировок) маршрутов в том, чтобы дать каждому маршрутизатору возможность информировать соседей о текущем количестве переходов от одного устройства к другому. Но здесь может возникнуть одна проблема: маршрутизаторы могут предоставлять некорректную информацию о сетях, о которых у них нет достоверных сведений. Лекарство от многих часто встречающихся проблем маршрутизации может быть найдено в четырех дополнительных свойствах RIP: ограничении количества переходов, корректировке отмены маршрута, расщеплении горизонта и таймерах временного удерживания изменений.

Ограничение количества переходов

Больше всего полемики вызывает такое свойство RIP, как ограничение количества переходов (hop count limit). При проектировании протокола для всех RIP-сред было установлено ограничение в 15 переходов. Это ограничение стало палкой о двух концах. Хотя ограничение количества переходов – это эффективное средство борьбы с петлями маршрутизации, но уж слишком строго ограничивается размер среды, в которой может использоваться протокол.

Число переходов, через которые прошел пакет данных, можно увидеть в заголовке маршрутизируемого протокола. Например, IP-пакеты содержат поле Time to Live (TTL), о котором рассказывалось в главе 5 «Пе-

ремещение данных маршрутизаторами». У пакетов, которые отправляются с ПК по RIP, отсчет TTL начинается с 15. Каждый раз, когда пакет проходит через маршрутизатор, значение поля TTL уменьшается на 1. Когда пакет осуществляет свой пятнадцатый переход, TTL достигает нуля и устройство отбрасывает пакет как недоставленный. Если возникает петля маршрутизации, она сможет длиться не более 15 переходов от маршрутизатора к маршрутизатору, так как затем пакет будет отброшен, и маршрутизатор займется выполнением других задач. Аналогично, если в заголовке IPX-пакета присутствует поле «управление пересылкой» (transfer control), то его значение при каждом переходе увеличивается на 1, и, когда оно достигает 16, пакет отбрасывается.

Каждый раз, когда пакет попадает на маршрутизатор RIP, от него отделяется заголовок маршрутизируемого протокола. Маршрутизатор определяет наилучший путь следования данных, и пакет опять получает заголовок, но уже скорректированный RIP. Когда к пакету вновь присоединяется заголовок, то изменяется значение поля, следящего за количеством сделанных переходов (такое как поле TTL (Time to Live) в пакете IP). Это означает, что пакет прошел еще через один маршрутизатор. Поэтому когда пакет попадает на маршрутизатор, то, исследуя заголовок, маршрутизатор может сразу же определить, достиг ли пакет предельного количества переходов. Если это так, то пакет может быть отброшен.

В те годы, когда разрабатывался RIP, его создатели вряд ли могли представить себе отдельную среду, содержащую более 15 переходов. Многие наиболее крупные сети того времени не имели и 10 сетевых устройств. Путь, длина которого превосходила бы 15 переходов, казался чем-то маловероятным.

По мере развития технологии маршрутизации и укрупнения сетей становилось понятно, что среды маршрутизации очень скоро выйдут за пределы 15 RIP-переходов. Было бы очень просто отказаться от старого протокола маршрутизации RIP и перейти на современную версию. Но вместо того чтобы перепроектировать RIP так, чтобы он отвечал требованиям сегодняшних сетей, было решено создать на основе его архитектуры (но без ограничений) новые протоколы. Такие протоколы маршрутизации, как OSPF, были разработаны для удовлетворения потребностей предприятий, которым не подходил RIP.

RIP не был перепроектирован с целью удаления ограничения количества переходов, потому что такое ограничение является ценным средством борьбы с зацикливанием. Хотя ограничение количества переходов и ограничивает размер вашей среды маршрутизации, но все же оно больше помогает администраторам, чем мешает. Помните, что большая часть петель маршрутизации возникает неумышленно, и вы можете их сразу не заметить. Ограничение количества переходов RIP – это еще одно средство, с помощью которого инженеры могут справиться с такой распространенной проблемой.

Отмена маршрута

В RIP также реализована корректировка, отменяющая маршрут. Отмена маршрута (route poisoning) происходит тогда, когда некоторому пути в таблице маршрутов присваивается значение 16 для количества переходов. После такого присваивания путь становится невидимым или недостижимым. Отмена маршрута может быстро предотвратить появление петли маршрутизации.

Рассмотрим в качестве примера ситуацию, в которой маршрутизатор должен принять решение об отмене маршрута. Если маршрутизатор непрерывно получает обновления маршрутов, в которых метрика (стоимость) некоторого пути возрастает, то маршрутизатор считает, что возникла петля маршрутизации. Тогда маршрутизатор устанавливает в своей таблице маршрутов метрику пути, равную 16, и инициирует обновление маршрутов. Это обновление предупреждает все соседние устройства о том, что путь «испорчен» и не должен использоваться. На рис. 15.2 изображена сеть, в которой может произойти отмена маршрута.

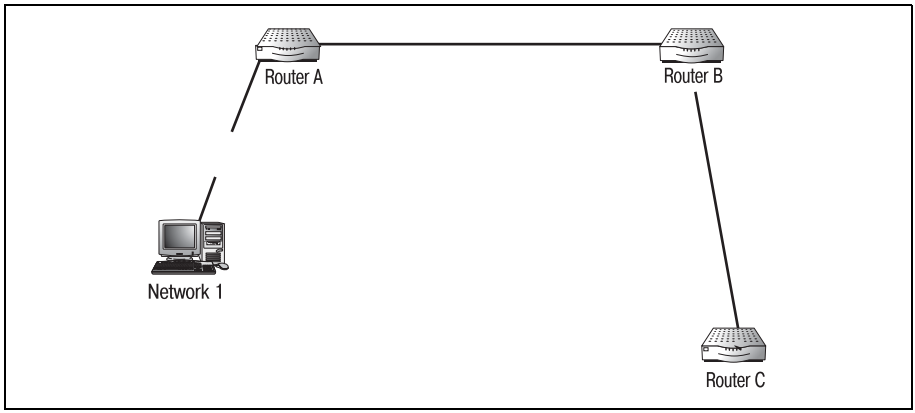

Рис. 15.2. *Испорченный путь*

На рис. 15.2 вы видите, что канал связи между сетью 1 и маршрутизатором A более недоступен. В процессе обновления маршрутизатор B сообщает как маршрутизатору A, так и C, что он может достичь сети 1 за 2 перехода. Маршрутизатор A, осознав, что он сам уже не может достичь сети 1, изменяет в своей таблице маршрутов метрику для сети с 1 на 3: 1 (его собственная метрика) + 2 (метрика, полученная от маршрутизатора B), и отправляет обновление соседям. Маршрутизатор B получает обновление, видит, что A изменил свою метрику с 1 на 3, и корректирует свою таблицу маршрутов значением 4 для метрики. Это классическая петля.

Чтобы устранить проблему, маршрутизатор C (получив корректировку маршрутизации от устройства B) видит, что возникла петля, и сра-

зу же устанавливает метрику пути в 16. Когда маршрутизаторы A и B получают обновление от C, они узнают, что путь испорчен и игнорируют его.

Расщепление горизонта

«Расщепление горизонта» (split horizon) – это еще одно средство, способствующее предотвращению петель маршрутизации. Правило расщепления горизонта состоит в том, что маршрутизатор не отправляет обновление маршрута обратно тому устройству, от которого пришла информация об обновлении. Благодаря исключению таких избыточных обновлений маршрутизаторы не вводятся в заблуждение и не считают действующими те каналы, которые на самом деле не существуют.

На рис. 15.3 приведен пример сети, которой помогло бы применение правила расщепления горизонта.

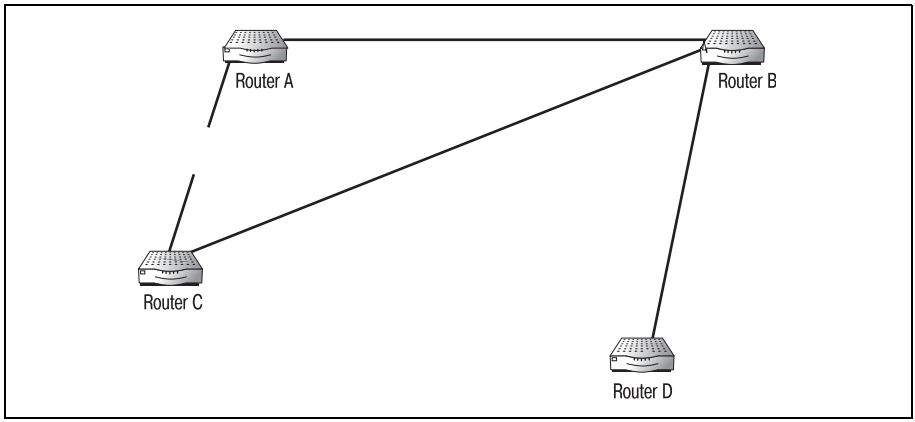

Рис. 15.3. *Расщепление горизонта*

На рис. 15.3 изображены четыре маршрутизатора (A, B, C и D). Для наглядности давайте будем считать, что канал между маршрутизаторами A и C неисправен. Маршрутизатор A корректирует свою таблицу маршрутов, чтобы показать, что канала связи с C больше не существует. Маршрутизатор A отправляет обновление маршрутизатору B, чтобы уведомить его о недоступном канале. Но прежде чем маршрутизатор сможет отправить свое обновление, он получит корректировку от маршрутизатора D, в которой будет сказано, что канал от A к C работает. Тогда маршрутизатор B исправит свою таблицу и уведомит маршрутизатор A, что канал между A и C функционирует (хотя это и не так). Возникает петля маршрутизации между устройствами A и B.

Чтобы справиться с этой проблемой, применим правило расщепления горизонта, которое говорит, что маршрутизатор не может посылать обновление тому маршрутизатору, от которого он получил корректировку, до тех пор пока не обновлены таблицы остальных его соседей. Ис-

пользуя такое правило, маршрутизатор может быть уверен в том, что корректная информация будет отправлена всем заинтересованным сторонам.

Таймеры временного удерживания изменений

Таймеры временного удерживания изменений (hold-down timers) используются вместе с другими возможностями RIP, например с расщеплением горизонта. Такой таймер применяется для того, чтобы задать количество времени, в течение которого какой-то определенный путь не может быть обновлен. В примере с расщеплением горизонта таймер временного удерживания изменений можно было бы использовать, чтобы предотвратить активацию бездействующего маршрута.

Когда маршрутизатор обнаруживает недоступный канал, инициируется обновление таблицы маршрутов. Тогда маршрутизатор устанавливает для этого маршрута таймер временного удерживания изменений. Таймер не допускает обновления таблицы маршрутов данного маршрутизатора потенциально недействительной информацией, полученной от соседнего устройства (которое могло еще не получить корректировку). После того как время удерживания истекает, маршрутизатор снова получает возможность обновлять путь.

Алгоритм маршрутизации RIP

RIP остается простым эффективным протоколом, который соответствует потребностям сетей небольшого и среднего размеров. Хотя за прошедшие годы структура маршрутизируемых сетей и изменилась, RIP просто превратился из «протокола маршрутизации для любых сетей» в «лучший протокол маршрутизации для небольших и средних сетей».

Одним из объяснений успеха RIP, выдержавшего испытание временем, является то, что это протокол вектора расстояния. То есть протокол использует для маршрутизации алгоритм вектора расстояния (известный также как алгоритм Беллмана-Форда). Алгоритм определяет, каким образом протокол принимает решение о выборе маршрутов и куда отправляются корректировки таблицы маршрутов.

Говоря буквально, этот алгоритм вычисляет расстояния до адресата по различным путям (векторам), а затем отправляет данные по кратчайшему пути. Значения (метрики), приписанные этим векторам (которые алгоритм использует для расчета маршрута), могут быть произвольными числами, заданными администраторами маршрутизатора, а могут представлять реальные значения, например время задержки пакета для конкретного канала. (В следующем разделе вы узнаете, как информация о метриках RIP извлекается из заголовка маршрутизируемого протокола.)

Примечание

Заголовок протокола – это часть пакета, содержащая специальную протокольную информацию. Каждый маршрутизируемый или WAN-протокол (но не протокол маршрутизации) добавляет собственную заголовочную информацию к каждому пакету.

Когда маршрутизатор получает RIP-пакет, содержащий информацию об обновлении таблицы маршрутов, алгоритм маршрутизации использует два основных компонента данных, составляющих заголовок пакета, для сравнения с существующей таблицей. Результатом сравнения является тот путь (или пути), по которому пакет может достичь адресата. Если в результате сравнения получено несколько путей, то алгоритм сравнивает метрики всех маршрутов, чтобы выбрать «лучший».

В следующем разделе на примере будет рассмотрен весь процесс (от начала и до конца) маршрутизации пакета с помощью RIP. При этом будут подробно описаны алгоритмы, обновления и заголовки протоколов. Когда мы лучше поймем, как работает RIP, то перейдем к изучению команд и процедур конфигурирования RIP на маршрутизаторе Cisco.

Как работает RIP

RIP работает как любой маршрутизируемый протокол или WAN-протокол. RIP – это общий язык, на котором могут говорить все маршрутизаторы в пределах одной среды, для того чтобы облегчить перемещение данных от одной сети к другой. Когда имеешь дело с таким протоколом маршрутизации, как RIP, важно отдавать себе отчет в том, что все маршрутизаторы рассматриваемой среды должны использовать один протокол маршрутизации.

Примечание

Хотя все маршрутизаторы некоторой среды должны использовать один и тот же протокол маршрутизации, им не обязательно использовать один и тот же маршрутизируемый протокол. Маршрутизаторы могут соединять сети, которые работают с различными маршрутизируемыми протоколами. Но для того чтобы информация пересылалась с маршрутизатора на маршрутизатор, устройства должны совместно использовать протокол маршрутизации.

Пакет данных достигает маршрутизатора, уже претерпев некоторые изменения (отсылаем вас к разговору о маршрутизируемых протоколах: пакет сегментируется и инкапсулируется с данными, представляющими применяемый маршрутизируемый протокол). Маршрутизатор получает данные и начинает читать информацию заголовка протокола. Протокол маршрутизации пока еще не используется.

При обсуждении того, как RIP обращается с пакетами и заголовками пакетов, помните, что будучи протоколом маршрутизации RIP не инкапсулирует данные. Это

значит, что не существует RIP-инкапсулированных пакетов. RIP (как протокол маршрутизации) просто работает с пакетами, подготовленными другими, маршрутизируемыми протоколами, такими как IP.

Сначала маршрутизатор смотрит на поле заголовка «адрес назначения» (поля заголовка маршрутизируемого протокола, в данном случае IP, представлены в табл. 15.2). Затем маршрутизатор сравнивает адрес назначения с записями в своей таблице маршрутов (созданной протоколом маршрутизации). Имейте в виду, что маршрутизатор еще не принял решение о маршрутизации пакета; маршрутизатором еще не использовались никакие протоколы: ни маршрутизации, ни какие-либо еще. Пока что маршрутизатор просто определяет, куда пакет стремится попасть.

После того как маршрутизатор определил, что пакет направлен адресату, который достижим с одного из его интерфейсов, он передает пакет протоколу RIP. RIP более внимательно изучает таблицу маршрутов, чтобы определить, куда именно нужно переправить пакет. Работая с адресом назначения, указанным в исходном протокольном заголовке пакета, RIP сравнивает его значение со списком возможных известных адресатов. Если найдено более одного соответствия, то RIP смотрит на метрики путей. В табл. 15.2 приведен пример заголовка IP-пакета. Из примера видно, что в заголовке присутствует вся необходимая информация. В табл. 15.3 дан пример таблицы маршрутов. Используем эти две таблицы для того, чтобы проследить за процессом маршрутизации данных протоколом RIP.

Таблица 15.2. Заголовок входящего IP-пакета

Поле	Значение
Версия	4
Длина заголовка	6
Тип сервиса	0
Общая длина	16
Идентификатор сегмента	1
Флаги	0
Смещение фрагмента	0
TTL	15
Протокол	17
Контрольная сумма	1024
Адрес источника	153.85.23.15
Адрес назначения	153.85.26.85

Данные заголовка пакета сравниваются с данными таблицы маршрутов устройства. В результате сравнения адреса назначения из заголов-

ка с адресами сетей, перечисленными в таблице маршрутов, найдено два возможных соответствия: маршрутизаторы C и D.

Таблица 15.3. Пример таблицы маршрутов

Сеть	Следующий переход	Метрика	Таймер	Флаги
153.85.23.0	Маршрутизатор B	4	30–180–240	0
153.85.24.0	Локальный	0	30–180–240	0
153.85.25.0	Локальный	0	30–180–240	0
153.85.26.0	Маршрутизатор C	2	30–180–240	0
153.85.26.0	Маршрутизатор D	3	30–180–240	0
203.152.0.0	Маршрутизатор C	2	30–180–240	0

Теперь маршрутизатор должен сделать выбор: по какому маршруту отправить пакет, чтобы он достиг сети 153.85.26.0? В соответствии с таблицей маршрутов маршрутизатор может переслать пакет или маршрутизатору B, или C. Для выбора пути применяется алгоритм маршрутизации RIP. Алгоритм сравнивает метрики путей: 2 для маршрутизатора C и 3 для D, и определяет, что пакет следует переправить маршрутизатору C.

Выбрав наилучший путь для пакета, RIP переключается на передачу данных по этому пути. К пакету вновь присоединяется заголовок IP, но при этом значения некоторых полей изменяются. Поля заново присоединенного IP-заголовка пакета перечислены в табл. 15.4.

Таблица 15.4. Заново присоединенный протокольный заголовок

Поле	Значение
Версия	4
Длина заголовка	6
Тип сервиса	0
Общая длина	16
Идентификатор сегмента	1
Флаги	0
Смещение фрагмента	0
TTL	14
Протокол	17
Контрольная сумма	1024
Адрес источника	153.85.23.15
Адрес назначения	153.85.26.85

Подытожим все вышесказанное. RIP, будучи протоколом маршрутизации, работает с пакетами, отправленными маршрутизируемыми

протоколами. Когда устройство получает пакет маршрутизируемого протокола, он передается в ведение RIP. Исследуя заголовок пакета, RIP выделяет ключевую информацию: адреса источника и назначения и метрики маршрутов. Затем алгоритм маршрутизации RIP рассчитывает наилучший путь передачи пакета, заново присоединяет к пакету заголовок маршрутизируемого протокола и пересылает пакет.

Такая же процедура повторяется на каждом маршрутизаторе, куда приходит пакет. Это значит, что для обеспечения точности маршрутизации таблицы маршрутов RIP на всех этих устройствах должны быть идентичными. В следующем разделе будет рассказано о том, как RIP обновляет таблицы маршрутов и поддерживает целостность сетевой среды.

Обновления маршрутов и RIP

Кроме выполнения обязанностей по перемещению данных по сети, RIP также должен заниматься обновлением таблиц маршрутов тех устройств, на которых он работает. С помощью автоматических корректировок RIP выполняет обновление таблиц маршрутов и достигает успешной конвергенции сети.

Корректировки (обновления) RIP отправляются каждые 30 секунд каждым маршрутизатором сети. Но, в отличие от большей части протоколов состояния канала, маршрутизаторы RIP посылают обновления только соседним устройствам. Эти обновления (в зависимости от инициатора обновления) могут включать в себя целую таблицу или же ее часть.

Примечание

Инициатор обновления – это событие, которое или констатирует обновление маршрутов, или запрашивает такое обновление.

Инициировать обновление маршрутов RIP могут три события: истечение времени корректировки маршрутов (таймер update routing), изменение состояния канала и непосредственный запрос RIP на обновление. Первое событие отражается в поле таблицы маршрутов RIP. По умолчанию таймер корректировки маршрутов устанавливается в 30 секунд. Когда это время проходит, маршрутизатор отправляет копию своей таблицы маршрутов каждому своему непосредственному соседу (устройству, с которым он соединен напрямую). Соседи используют информацию для обновления собственных таблиц маршрутов и порождения своих собственных корректировок.

Второе событие происходит, если канал связи между двумя маршрутизаторами (или между маршрутизатором и сетью) оказывается поврежденным, тогда непосредственно связанный с этим каналом маршрутизатор обновляет свою таблицу и незамедлительно отправляет коррек-

тировку соседям. В случае обновления, вызванного изменением сети, отправляется только та часть таблицы маршрутов, которая подверглась изменению. Соседние маршрутизаторы обновляют соответствующие части своих таблиц и продолжают передавать корректировки.

Наконец, третье событие происходит, когда маршрутизатор Cisco, использующий RIP, запрашивает обновление у определенного маршрутизатора. Обычно это делается по истечении срока временного удерживания изменений. Когда маршрутизатор получает запрос на обновление, запрашивающей стороне отсылается вся таблица маршрутов.

Вне зависимости от того, какое событие вызвало обновление, таблицы маршрутов отправляются в формате заголовочной информации RIP. Только порожденные протоколом маршрутизации пакеты используются для отправки обновлений таблиц маршрутов. Пакет обновления обычно состоит из одного заголовка (без данных). Помните, что обновление маршрутов не использует инкапсуляцию протокола, это средство пересылки данных с одного маршрутизатора на другой. В табл. 15.5 представлен заголовок пакета обновления таблицы маршрутов. Поля, используемые RIP для распространения информации о корректировках таблиц маршрутов, приведены в табл. 15.6.

Таблица 15.5. Поля заголовка сообщения RIP об обновлении

Поле	Назначение
Команда	Указывает, является пакет запросом на обновление или же ответом на запрос
Версия	Указывает номер используемой версии RIP, обычно это версия 2
Ноль	Два байта, заполненных нулями

Таблица 15.6. Поля таблицы маршрутов RIP

Поле	Назначение
Идентификатор семейства адресов	Определяет, какое семейство протоколов используется, обычно IP
Метка маршрута	Заполнено нулями
Адрес	Протокольный адрес маршрута
Ноль	Заполнено нулями
Ноль	Заполнено нулями
Метрика	Метрика, соответствующая адресу

Поля, перечисленные в табл. 15.6, могут входить в заголовок RIP по несколько раз, чтобы представить всю таблицу маршрутов. (Одно обновление ограничивается 25 адресами.) Если пакет сообщает об изменении канала, то он содержит всего одну запись.

RIP продолжает отсылать обновления каждому из соседних маршрутизаторов до тех пор, пока не будет достигнута конвергенция. Конвергенция – это то состояние, в котором все маршрутизаторы одной среды имеют одинаковую информацию о маршрутах. Если сеть не достигла конвергенции, могут возникнуть серьезные проблемы, например петли маршрутизации. Если при использовании RIP на вашем маршрутизаторе Cisco вы обнаружите, что конвергенция недостижима или достигается медленно, можно изменить некоторые параметры, например таймеры.

Пришло время применить изученные нами понятия (относящиеся к RIP) к маршрутизаторам Cisco. Будем двигаться вперед и рассмотрим настройку RIP на маршрутизаторе Cisco.

Конфигурирование RIP

Команды, используемые для конфигурирования RIP на маршрутизаторе Cisco, должны показаться похожими на аналогичные команды для других протоколов. И, как и многие другие, уже рассмотренные в этой книге протоколы, RIP должен быть сначала активирован и лишь затем сконфигурирован. Чтобы начать процесс настройки RIP, необходимо перевести маршрутизатор в режим глобального конфигурирования. В этом режиме вводим команду `router rip` для входа в режим конфигурирования протоколов маршрутизации:

```
Router#configure terminal
Router(config)#router rip
Router(config-router)#
```

Примечание

На протяжении оставшейся части книги команда `router` будет часто использоваться. Она применяется для активирования почти всех протоколов маршрутизации.

Теперь можно задавать параметры, необходимые для работы протокола. Чтобы сделать возможной работу RIP на маршрутизаторе Cisco, определите сеть маршрутизатора. Так маршрутизатор получит отличительную черту, которая будет характеризовать его в сети. Используйте команду `network` для указания адреса сети маршрутизатора:

```
Router#configure terminal
Router(config)#router rip
Router(config-router)#network 198.124.0.0
Router(config-router)#^Z
```

Примечание

Отдельные интерфейсы маршрутизатора также необходимо конфигурировать (следуя примерам, приведенным в первой части) для работы с соответствующим марш-

рутизируемым протоколом. Даже если использование RIP было корректно разрешено, это принесет мало пользы, если интерфейсы не будут настроены.

Эти простые команды разрешают работу RIP на маршрутизаторе Cisco. Так как команды выполняются в режиме глобального конфигурирования, то относятся ко всем интерфейсам маршрутизатора. Пока никакого дополнительного конфигурирования для активации RIP на маршрутизаторе больше не требуется. Но вы должны знать о существовании некоторых необязательных параметров. Эти параметры значительно упрощают администрирование и сопровождение маршрутизатора Cisco, использующего RIP. В следующем разделе будут описаны параметры, задающие таймеры RIP и определяющие адреса соседей.

Сопровождение RIP

Cisco предоставляет специалистам большую свободу в использовании RIP. Для настройки его работы существует множество необязательных параметров. Многие свойства и параметры, связанные с RIP, могут быть установлены вручную, тем самым можно удовлетворить потребности практически любой среды. Как говорилось в предыдущем разделе, необязательные параметры включают в себя:

- Таймеры RIP
- Адреса соседних маршрутизаторов
- Версию RIP, которая совместима с другими реализациями и продуктами других производителей

Изучив синтаксис и назначение данных параметров, вы сможете лучше управлять маршрутизатором и содержать среду маршрутизации «в чистоте и порядке».

Установка таймеров RIP

Ряд необязательных параметров Cisco, доступных для изменения пользователям, относится к таймерам RIP. В процессе маршрутизации RIP использует четыре таймера: таймер корректировки маршрутов, таймер временного удерживания изменений, тайм-аут маршрута и таймер удаления маршрута. Все четыре таймера связаны со способом обновления таблиц маршрутов и тем интервалом, через который маршрутизатор делает эти обновления. Каждый таймер имеет значение по умолчанию, которое может быть заменено на более подходящее для конкретной сети. Для установки таймеров RIP используйте команду `timers basic`. Структура этой команды такова:

```
#timers basic <routing update timer> <route timeout> <hold-down timer> <route
removal>
```

Например, чтобы задать определенные значения времени с помощью команды `timers basic`, вам следует выполнить в режиме конфигурирования протокола маршрутизации такие команды:

```
Router(config)#router rip
Router(config-router)#timers basic 30 180 45 270
Router(config-router)#^Z
```

Примечание

Будьте осторожны при использовании команды `timers basic`. Вы можете не только серьезно изменить производительность сети, но и случайно установить неверное значение (так как команда изменяет все четыре таймера одновременно).

В предыдущем примере для таймера корректировки маршрутов было установлено значение 30 секунд, то есть данный маршрутизатор Cisco отправляет обновления своей таблицы маршрутов каждые 30 секунд. Тайм-аут для маршрута был установлен равным 180 секундам. Это означает, что если обновление от какого-то маршрутизатора не поступило в течение 180 секунд, то такой маршрутизатор будет помечен. Таймер временного удерживания изменений установлен равным 45 секундам. Так как этот таймер определяет время, в течение которого маршрутизатор не может производить никаких корректировок, то он должен быть отрегулирован так, чтобы не совпадать ни с каким существующим таймером корректировки.

Наконец, таймер удаления пути установлен равным 270 секундам. Любой маршрутизатор, обновление от которого не получено в течение этого времени, считается недостижимым и удаляется из таблицы маршрутов. Так как таймеры контролируют обновления, рассылаемые соседям маршрутизатора Cisco, вы можете захотеть вручную проинформировать маршрутизатор о местоположении и адресах его соседей.

Конфигурирование RIP-соседей вручную

Еще одной возможностью при конфигурировании маршрутизатора Cisco является указание соседних маршрутизаторов RIP. Так как RIP устанавливается на маршрутизаторе глобально, по умолчанию он будет активен на всех интерфейсах. Однако Cisco осознает, что в некоторых средах не будет необходимости в работе RIP на каждом интерфейсе. Например, один маршрутизатор может иметь соединения сразу с несколькими сетями, одна из которых использует RIP, а другая – нет. Для такого маршрутизатора отправка обновлений по обоим каналам будет создавать только ненужную нагрузку на сеть.

Маршрутизатор Cisco автоматически опознает своих соседей. Но в среде, где присутствуют маршрутизаторы разных типов, вы можете не захотеть использовать RIP для работы с каждым маршрутизатором, с которым вы взаимодействуете. Или же вы захотите получить более

полный контроль над процессом маршрутизации. В любом из этих случаев используйте команду neighbor.

Команда Cisco IOS neighbor применяется для указания соседних маршрутизаторов, которым RIP посылает обновления таблиц маршрутов. И наоборот, команда passive-interface определяет интерфейсы, через которые RIP *не* посылает обновления. Комбинация этих команд позволяет администраторам более тщательно контролировать процесс работы RIP в конкретной среде.

Чтобы использовать команду neighbor, переведите маршрутизатор в режим конфигурирования протоколов маршрутизации. Синтаксис команды:

```
#neighbor <protocol address>
```

С помощью этой команды настроим маршрутизатор RIP на распознавание маршрутизатора с адресом 198.53.10.1 как соседнего:

```
Router#configure terminal
Router(config)#router rip
Router(config-router)#neighbor 195.53.10.1
Router(config-router)#^Z
```

Команда neighbor может применяться для задания любого количества соседних маршрутизаторов. Чтобы посмотреть, какие маршрутизаторы определены в вашей системе как соседние, используйте команду show running-config.

```
Router#show running-config

!
router rip
 network 153.5.0.0
 neighbor 153.5.86.2
 neighbor 153.5.46.1
!

(output abbreviated)
```

Примечание

Помните, что все изменения в настройках отражаются только в файле running-config. Чтобы сохранить их, вы должны сами скопировать изменения в файл startup-config.

Файл running-config показывает, что два маршрутизатора были сконфигурированы как соседи маршрутизатора Cisco. Чтобы удалить соседа, назначенного вручную, используйте ключевое слово no:

```
Router#configure terminal
Router(config)#router rip
Router(config-router)#no neighbor 153.5.86.2
Router(config-router)#^Z
```

В то время как команда neighbor позволяет указать конкретный маршрутизатор в качестве соседнего устройства, команда passive-interface запрещает отправку обновлений RIP с определенных интерфейсов. Команда passive-interface имеет такой формат:

```
#passive-interface <interface> <interface number>
```

Чтобы запретить отсылку обновлений маршрутов с интерфейса Ethernet 1, используйте приведенную ниже последовательность команд:

```
Router#configure terminal
Router(config)#router rip
Router(config-router)#passive-interface ethernet 1
Router(config-router)#^Z
```

Овладев командами neighbor и passive-interface, пользователь Cisco может полнее контролировать работу RIP на маршрутизаторе. Существует еще одна важная команда конфигурирования, которая позволяет администратору указать, какая версия RIP должна использоваться маршрутизатором. Маршрутизаторы Cisco способны поддерживать как версию 1, так и версию 2 (и поставляются с поддержкой обеих версий). В следующем разделе будет рассказано, как выбрать версию для использования на вашем конкретном маршрутизаторе.

Работа с различными версиями RIP

Использование RIP в среде, в которой присутствуют маршрутизаторы разных производителей, может создать проблемы. Например, маршрутизаторы Cisco по умолчанию используют версию 1 RIP. Но оборудование некоторых производителей может работать только с версией 2 протокола RIP.[1]

Cisco предоставляет набор команд, которые позволяют указать, какая версия протокола RIP должна использоваться глобально или для какого-либо интерфейса. Глобальная команда, version, определяет, какая версия RIP используется на всем маршрутизаторе.

Примечание

Версии 1 и 2 протокола RIP несовместимы. Если вы работаете в Cisco-среде и настраиваете один из маршрутизаторов на использование RIP 1, то этот маршрутизатор не будет работать (остальные маршрутизаторы Cisco по умолчанию будут использовать RIP 2). Прежде чем изменять настройки, убедитесь в том, что все ваши маршрутизаторы используют совместимые версии RIP.

[1] В документации Cisco IOS вплоть до версии 12.2 написано: «По умолчанию принимаются пакеты RIP версии 1 и 2, но отсылаются только пакеты RIP версии 1». – *Примеч. науч. ред.*

Команда `version` часто применяется для того, чтобы сделать маршрутизатор Cisco транслятором для протокола, то есть использовать RIP 1 на одном интерфейсе (для взаимодействия с оборудованием, работающим с RIP 1) и настроить RIP 2 на другом (чтобы построить «мост» к устройствам, использующим RIP 2). Синтаксис глобальной команды `version` – это просто `version <version number>`:

```
Router#configure terminal
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#^Z
```

Гибкость настройки обеспечивается возможностью определения версии RIP на уровне интерфейса. Пользователь может указать, какую версию RIP использовать как для отсылки, так и для приема на каждом отдельном интерфейсе.

Такая гибкость полезна, если вы участвуете в перераспределении маршрутов. Для маршрутизатора, который занимается перераспределением маршрутов, можно решить получать обновления только от маршрутизаторов RIP 1, а передавать эти обновления маршрутизаторам RIP 2. Указание версий RIP для отправки и получения на отдельном интерфейсе чрезвычайно полезно при экспериментировании с различными конфигурациями сети.

Примечание

Команды установки версии RIP для интерфейса доступны только в случае, если на этом интерфейсе работает IP.

Команда определения версии RIP для отправки или приема обновлений интерфейсом выполняется в режиме конфигурирования интерфейсов и имеет такой синтаксис:

```
#ip rip <direction> <version number(s)>
```

Ниже приведен пример команды `version` для интерфейсов, которая указывает, что интерфейс Ethernet 1 будет посылать только обновления RIP 1, в то время как Ethernet 2 будет получать как версию 1, так и 2:

```
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-int)#ip rip send version 1
Router(config-int)#interface ethernet 1
Router(config-int)#ip rip receive version 1 2
Router(config-int)#^Z
```

Рассмотренные нами команды очень полезны любому, кто использует RIP в Cisco-среде. Но эти команды только устанавливают свойства RIP. Часто можно встретить маршрутизатор с уже существующей конфигурацией RIP. Последний раздел посвящен командам, используе-

мым для просмотра статистики уже сконфигурированных маршрутизаторов RIP.

Просмотр статистических данных RIP

Настройка RIP на маршрутизаторе Cisco – это важный шаг на пути изучения механизмов маршрутизации Cisco. Но от конфигурирования RIP будет мало толку, если вы не сможете следить за работой маршрутизатора после того, как его настройка будет завершена. Cisco дает возможность просматривать информацию о ряде ключевых элементов, но самым важным для вас на данный момент является просмотр таблицы маршрутов RIP.

Примечание

Команда `show` (которую мы будем использовать для просмотра таблицы маршрутов RIP) может применяться для вывода сотен различных элементов информации. Все, от использующихся сокетов IP и до установленной версии IOS, может быть показано при помощи команды `show`.

Обеспечение отсылки и получения корректных обновлений маршрутов – это важный компонент сопровождения маршрутизаторов RIP. Вы можете просмотреть содержимое таблицы маршрутов, чтобы определить, активны ли пути или же повреждены, а также используются ли неисправные пути. Просмотр таблицы RIP – это также способ убедиться в том, что маршрутизатор действительно правильно сконфигурирован и работает корректно. Чтобы просмотреть таблицу RIP маршрутизатора, используйте команду `show` следующим образом:

```
Router#show ip route rip
```

Вывод команды будет выглядеть приблизительно так:

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR
       T - traffic engineered route

Gateway of last resort is not set

R 153.16.4.0 [90/1] via 153.16.4.1, 00:00:10 Ethernet1
C 198.56.0.0 is directly connected, Ethernet2
```

Регулярный просмотр таблицы маршрутов RIP может способствовать изменению настроек RIP (при необходимости) для обеспечения ровной работы сети. Например, если канал становится недоступным, но вы видите, что он все еще присутствует в таблице маршрутов, то можно изменить значения таймеров маршрутизации, чтобы избавиться от петли.

Резюме

- RIP – это один из старейших протоколов маршрутизации.
- В борьбе с петлями маршрутизации протоколу RIP помогают такие его свойства, как расщепление горизонта, таймеры временного удерживания изменений, отмена маршрута и ограничение длины маршрута.
- RIP может обслуживать только среды, состоящие из 15 или менее переходов.
- Обновления таблицы маршрутов информируют другие маршрутизаторы RIP-среды о текущем состоянии сети.
- Маршрутизатор RIP отсылает соседям полную копию своей таблицы маршрутов каждые 30 секунд.
- Маршрутизаторы Cisco по умолчанию используют RIP версии 2.[1]

Вопросы и ответы

Вопрос Почему RIP все еще используется несмотря на ограничение длины пути в 15 переходов?

Ответ Так как RIP ограничен 15 переходами в пределах одной среды, он отлично подходит для небольших сетей. В этой главе рассказывалось, что для конфигурирования RIP необходимо всего несколько команд. Поэтому этот протокол хорош для пользователей, которые не могут уделить управлению маршрутизатором много времени.

Тест

Вопросы

1. Как долго маршрутизатор Cisco ждет по умолчанию, прежде чем удалить маршрут (если не получено обновление от соответствующего маршрутизатора)?
2. Какая команда используется для просмотра таблицы маршрутов RIP?
3. Некоторая среда состоит из пяти маршрутизаторов, два из которых соединены с маршрутизатором A. Сколько таблиц маршрутов устройство A будет получать в качестве обновлений каждые 30 секунд?

Ответы

1. 240 секунд.

[1] См. сноску на стр. 314.

2. `show ip route rip`.
3. Две, по одной от каждого из непосредственных соседей.

Упражнение

1. Сконфигурируйте соединение маршрутизатора с сетью 10.0.0.0 для RIP. RIP будет работать только на одном интерфейсе Ethernet 0 (из двух интерфейсов маршрутизатора: Ethernet 0 и Ethernet 1). Кроме того, обновления должны приниматься от других устройств сети, работающих с RIP обеих версий, но маршрутизатор должен отправлять только обновления RIP версии 2. (Протокол IP на маршрутизаторе уже настроен.)

Решение

```
Router#configure terminal
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#passive-interface ethernet 1
Router(config-router)#^Z
Router#configure terminal
Router(config)# interface ethernet 0
Router(config-int)#ip rip send version 2
Router(config-int)#ip rip receive version 1 2
```

17

Конфигурирование OSPF

Протоколы вектора расстояния – это особая форма протоколов маршрутизации, процветающая в специфических средах. Многие инженеры избегают их по одной причине – используемый алгоритм маршрутизации. Протоколы вектора расстояния, такие как RIP, IGRP и EIGRP, имеют сравнительно низкую скорость конвергенции и маршрутизации; в данном случае «сравнительно» означает по сравнению с их ближайшими родственниками, протоколами состояния канала.

В этой главе будет рассмотрен один из протоколов состояния канала – протокол маршрутизации OSPF (Open Shortest Path First – первоочередное открытие кратчайших маршрутов). OSPF – это IGP состояния канала, который может работать приблизительно в таких же средах, что и IGRP или EIGRP. То есть OSPF, подобно RIP, IGRP и EIGRP, используется для маршрутизации данных в пределах, обозначенных пограничными шлюзами. OSPF очень популярен благодаря его природной способности к быстрой конвергенции. В данной главе будет описана технология, лежащая в основе OSPF, а также команды Cisco, используемые для конфигурирования OSPF. В главе представлены следующие разделы:

- Технология OSPF
- Обновления OSPF
- Конфигурирование OSPF

Введение в OSPF

OSPF был разработан в конце 80-х годов группой IETF в качестве замены для RIP (в это же время компания Cisco работала над IGRP.) Вмес-

то того чтобы использовать в качестве основы технологию RIP, IETF решила начать с нуля. Это решение привело к осознанию того, что для внутришлюзовой маршрутизации лучше использовать не протокол вектора расстояния, а протокол состояния канала.

В наши дни OSPF остается одним из наиболее популярных протоколов внутренней маршрутизации для больших сред. В приведенном ниже списке перечислены свойства OSPF, обеспечившие его популярность:

- Протоколы состояния канала имеют более высокую скорость конвергенции.
- Несмотря на то что OSPF относится к IGP, он может посылать маршруты другим автономным системам (и получать маршруты от них) при помощи протокола внешнего шлюза (EGP).
- OSPF может осуществлять маршрутизацию на основе адресов подсетей IP.
- OSPF обеспечивает перераспределение нагрузки для каналов одинаковой стоимости.

OSPF обладает рядом возможностей, отсутствующих в других протоколах маршрутизации. OSPF был создан для удовлетворения тех потребностей маршрутизации, которым не соответствовали протоколы вектора расстояния. При его создании не использовалась никакая из уже существующих платформ. В следующем разделе мы поговорим о технологии, которая сделала OSPF одним из ведущих IGP для больших сред маршрутизации.

Технология OSPF

Первое, чем следует заняться при изучении OSPF, – это терминология, связанная с особенностями работы протокола. По большей части протоколы, описанные ранее, используют один и тот же набор терминов и функций, а вот OSPF, за счет реализации новой технологии, предлагает новые понятия. Первое такое понятие – это *автономная система* (AC).

Примечание

В некоторых источниках автономные системы OSPF называют *доменами*. Многие протоколы вектора состояния работают в доменах, но Cisco использует термин «автономная система». Поэтому в нашей книге домены OSPF будут называться AC.

Пример OSPF-домена, или автономной системы, изображен на рис. 17.1.

С точки зрения OSPF автономная система – это наибольшая маршрутизируемая область. Другими словами, автономная система – это группа сетей, использующих одну схему адресации IP. Одна AC может содержать одну или несколько более мелких сетей. Граница AC определяется местоположением пограничных маршрутизаторов (шлюзов).

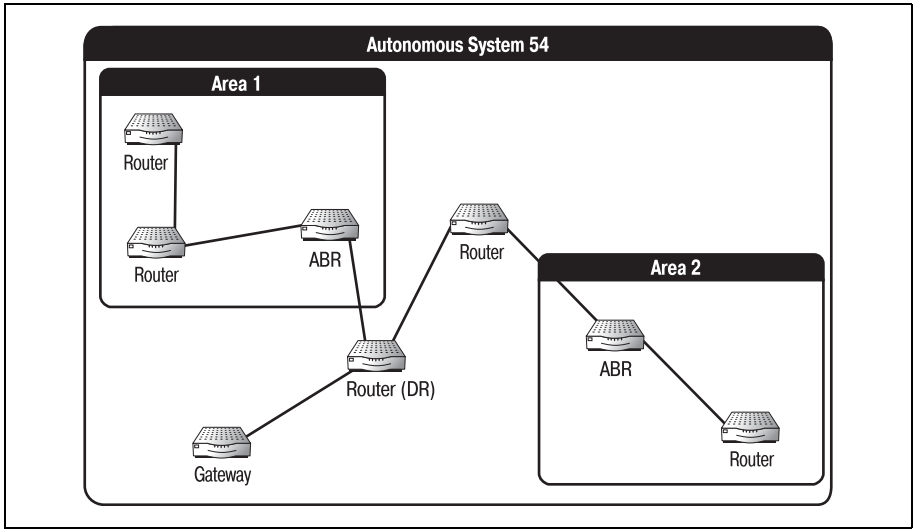

Рис. 17.1. *Автономная система OSPF*

Примечание

OSPF был спроектирован для работы только с IP. Поэтому адресация OSPF-сети – это всегда IP-адресация.

Связь автономной системы с другими объектами, Интернетом или другими автономными системами осуществляется при помощи маршрутизатора пограничного шлюза. OSPF – это протокол внутренней маршрутизации, то есть он может работать только на маршрутизаторах, находящихся внутри одной АС.

Несмотря на то что OSPF работает на маршрутизаторах, являющихся пограничными шлюзами, он не используется для взаимодействия с другими шлюзами. Это задача протоколов внешней маршрутизации, таких как BGP. Шлюзы используют OSPF для сбора маршрутной информации об автономной системе. Затем посредством EGP шлюз распространяет информацию об этих путях среди других автономных систем OSPF.

Одна автономная система OSPF может содержать несколько пограничных маршрутизаторов. Вне зависимости от количества все шлюзы АС занимаются одним и тем же – распределяют пути к внешним сетям, о которых сам OSPF ничего не знает.

Маршруты распространяются по автономной системе с помощью объявлений LSA (link state advertisement – объявление о состоянии канала). Объявление о состоянии канала содержит часть таблицы маршрутов, в которой указаны состояния непосредственно связанных с данным маршрутизатором каналов. LSA рассылаются всем маршрутизаторам, такой процесс называется *затоплением* (*flooding*) (в отличие от

протоколов вектора расстояния, рассылающих обновления только прямым соседям маршрутизатора). Каждый маршрутизатор создает на основе LSA базу данных текущего состояния сети.

В пределах АС может существовать несколько более мелких объектов. Их называют *областями*. Область – это сеть или группа маршрутизаторов, которые совместно используют топологическую базу данных. То есть все маршрутизаторы одной области имеют одинаковое представление об остальных частях АС. У каждой области автономной системы есть идентификатор – номер, используемый для идентификации всех маршрутизаторов данной области.

Примечание

Всем маршрутизаторам области разрешено хранить топологические данные только для данной области. Несмотря на то что они знают, какие адреса находятся вне области (за счет рассылок LSA), они не знают точной топологии этих адресов.

Области OSPF можно разделить на две категории: тупиковые (stub) и не совсем тупиковые области (not-so-stubby areas – NSSA). Как тупиковые, так и не совсем тупиковые области используются только в автономных системах, которые имеют дело с внешними маршрутами. То есть специальная область может использоваться только тогда, когда АС связана с другими АС и получает от них информацию о маршрутах.

Тупиковая область – это область АС, в которой невозможна передача информации о внешних маршрутах. Чтобы ограничить сетевой трафик, администратор может указать, что внешние LSA не должны попадать в некоторые области. Тупиковые области должны быть напрямую соединены с магистралью автономной системы. На рис. 17.2 изображена область, которая могла бы быть тупиковой.

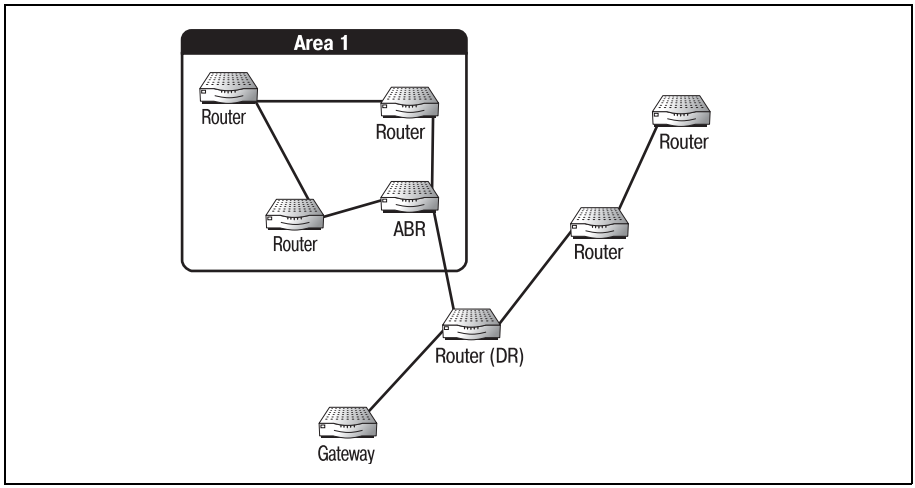

***Рис. 17.2.** Тупиковая область*

Маршрутизаторы, соединяющие области с другими частями среды, называются пограничными маршрутизаторами области (ABR – area border router). Чтобы область была тупиковой, необходимо, чтобы ее ABR был напрямую соединен с магистралью АС (как показано на рис. 17.2). Область 19 на рис. 17.3, например, не могла бы быть тупиковой областью, так как ABR этой области соединен с другой областью, а не с магистралью.

NSSA, как и тупиковая область, не принимает внешние LSA. Но если тупиковые области должны перенаправлять все пакеты (адресованные наружу) магистрали, то не совсем тупиковые области могут использовать маршруты по умолчанию для достижения некоторых адресатов. Так достигается большая гибкость, при этом трафик сообщений LSA также уменьшается.

И наконец, в автономной системе OSPF один маршрутизатор должен быть назначен выделенным маршрутизатором (DR – designated router). Такой маршрутизатор следит за процессом затопления сообщениями LSA и только он хранит полную базу данных маршрутов. То есть DR обладает топологической базой данных всей автономной системы.

Примечание

Когда маршрутизаторы автономной системы выбирают DR, они выбирают и резервный DR. Если у выделенного маршрутизатора возникают проблемы, то на резервном DR всегда есть полная база данных. Это избавляет от необходимости массового затопления автономной системы сообщениями LSA для заполнения базы данных DR.

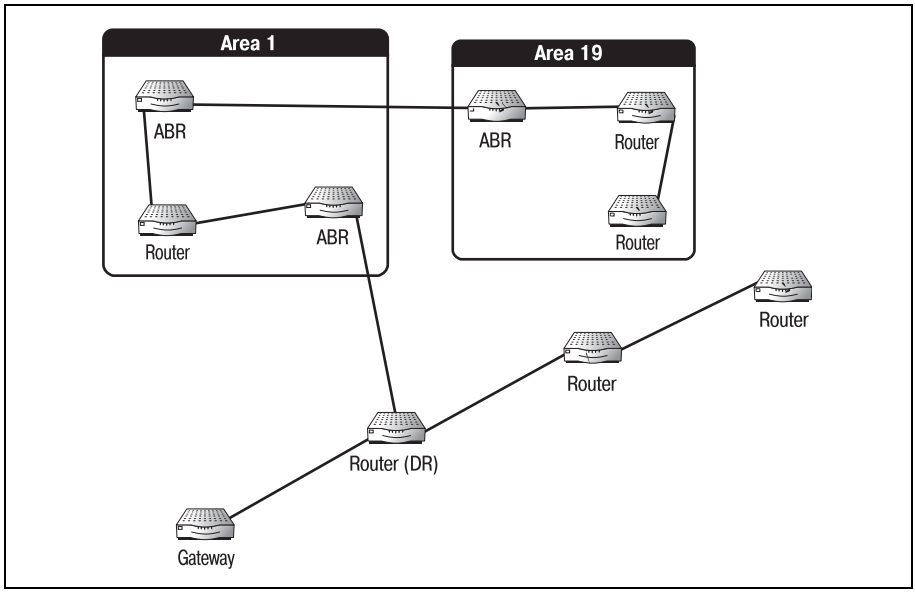

***Рис. 17.3.** Область, которая не может использоваться как тупиковая*

Изучить основные понятия терминологии OSPF очень важно еще и потому, что они используются и другими протоколами. Теперь, когда мы определили несколько используемых в OSPF терминов, давайте перейдем к изучению деталей технологии. В оставшихся разделах главы будут рассмотрены следующие темы:

- Алгоритмы состояния канала
- Тупиковые области
- NSSA
- Перераспределение маршрутов
- Обновления OSPF

Алгоритмы состояния канала

Алгоритм состояния канала базируется на очень простой теории. Алгоритм просто вычисляет кратчайший путь между двумя точками. Используемый OSPF алгоритм состояния канала работает со специальной метрикой, на основе которой и вычисляется кратчайший путь через АС.

Алгоритм – это формула, по которой протокол вычисляет метрики для того, чтобы определить кратчайший маршрут. Чаще других в сетях OSPF используется алгоритм Дейкстры.

Как математическая функция алгоритм Дейкстры используется для параллельного вычисления кратчайших путей ко всем точкам решетки. Имея заданную точку отсчета, алгоритм вычисляет расстояние (используя предопределенные метрики) между исходной точкой и каждой другой точкой сети.

Ниже приведено понятное объяснение работы алгоритма Дейкстры (в настоящий момент метки A, B, C и D произвольны):

1. Для точки отсчета (A) определить соседние маршрутизаторы (B и C).
2. Вычислить расстояние между точкой отсчета и соседними маршрутизаторами.
3. Пометить маршрутизатор, расстояние до которого является наименьшим (C).
4. Определить всех соседей для только что помеченного маршрутизатора и оставшихся соседей исходной точки отсчета (B и D).
5. Перейти к шагу 2.
6. Повторять до тех пор, пока не будут вычислены все маршруты.

   Шаг 1. Определить, какие маршрутизаторы непосредственно связаны с исходным. То есть, зная, какой маршрутизатор является точкой отсчета, алгоритм Дейкстры просматривает таблицу маршрутов и видит, какие устройства напрямую связаны с точкой отсчета.

Шаг 2. Алгоритм смотрит в таблицу маршрутов и находит в ней метрики, назначенные маршрутизаторам, непосредственно связанным с точкой отсчета.

Шаг 3. Алгоритм сравнивает метрики и метки (находящиеся у него в памяти), чтобы найти соседа с наименьшей метрикой.

Шаг 4. Аналогично шагу 1 алгоритм снова ищет прямых соседей. Но на этот раз алгоритм учитывает соседей, которые остались непомеченными на предыдущем шаге, и соседей маршрутизатора, который только что был помечен как имеющий наименьшую метрику.

Шаг 5. Алгоритм Дейкстры возвращается к шагу 2.

Шаг 6. Процесс продолжается до тех пор, пока не будут помечены все маршруты.

Короче говоря, алгоритм вычисляет кратчайший путь между двумя точками и помечает соответствующий путь. Давайте шаг за шагом прогоним алгоритм Дейкстры для сети, изображенной на рис. 17.4.

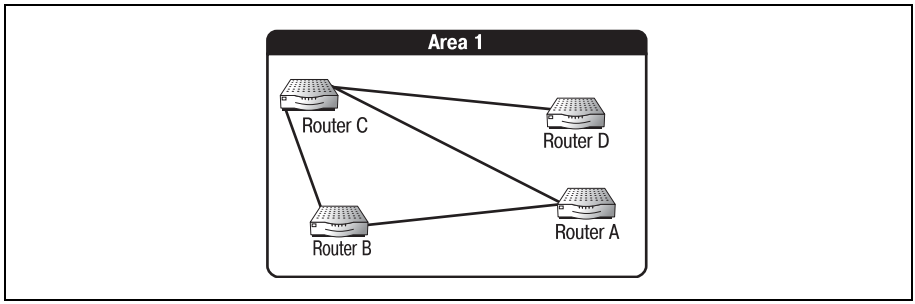

Рис. 17.4. *Соседи маршрутизатора A*

Шаг 1. Устанавливаем соседей точки отсчета – маршрутизатора A.

Шаг 2. Вычисляем расстояния между точкой отсчета и соседними маршрутизаторами. Соседями A являются B и C. Обратите внимание на стоимости каналов, связывающих A с его соседями. Наименьшей стоимостью обладает канал A–C, поэтому путь A–C является кратчайшим.

Шаг 3. Помечаем маршрутизатор, путь до которого оказался кратчайшим (C).

Шаг 4. Определяем оставшихся соседей. Рассматриваем непомеченных соседей A (B) и соседей C (D).

Шаг 5. Заново вычисляем маршруты. Маршруты вычисляются, и процесс повторяется до тех пор, пока не будет охвачена вся сеть. Заметьте, что путь A–B имеет стоимость 3, а C–D – стоимость 2. Но так как алгоритм запущен для точки A, то реальная стоимость C–D равна 3 (A–C + C–D). В этом случае стоимости A–B и A–D совпадают.

Можно предположить, что при увеличении количества маршрутизаторов в сети этот алгоритм становится достаточно сложным. Но в процессе маршрутизации алгоритм Дейкстры тратит несколько сот миллисекунд для расчета всей сети, так что это очень быстрая и эффективная формула.

Процесс одновременного вычисления всех маршрутов в пределах области имеет свои недостатки: он очень загружает процессор. Это означает, что маршрутизаторы, работающие с OSPF, обычно используют больше вычислительной мощности, чем маршрутизаторы, работающие с другими протоколами.

Как и всем алгоритмам маршрутизации, алгоритму состояния канала необходима метрика для определения кратчайшего расстояния между двумя точками. OSPF использует метрику, назначаемую локальным маршрутизатором. Учитывая пропускную способность своих интерфейсов, локальный маршрутизатор выполняет некоторые расчеты для определения метрики конкретного маршрута. Уравнение, используемое OSPF-маршрутизаторами для определения метрики канала, выглядит так:

$$\text{Значение метрики канала} = 10^8/\text{пропускная способность канала}$$

Из формулы следует, что наименьшие метрики имеют каналы с наибольшей пропускной способностью. Например, канал 128 Кбит/с имеет стоимость 781, в то время как стоимость T1 – 64. За счет этого каналам, имеющим наибольшую доступную пропускную способность, отдается предпочтение перед каналами, которые, может быть, были бы не в состоянии обработать трафик.

Но то, что канал обладает наибольшей пропускной способностью, еще не означает, что это наилучший выбор. Например, канал с наибольшей пропускной способностью, ведущий к месту назначения, может быть выделенной линией с оплатой за трафик. Администратор, вероятно, захочет, чтобы OSPF-маршрутизатор использовал эту линию не так часто, как бесплатный канал, ведущий к тому же адресату.

Примечание

Так как метрика, используемая OSPF, автоматически вычисляется маршрутизатором, то этим не приходится заниматься администратору, и он может сосредоточиться на других задачах.

В тех ситуациях, когда вы не хотите, чтобы первой выбиралась линия с наибольшей пропускной способностью, можно вручную назначить ей значение, которое заменит имеющуюся метрику. Вы можете присвоить выделенной линии метрику, превышающую метрику бесплатной линии (обладающей меньшей пропускной способностью), и тем самым обеспечить ее более редкое использование. Такая замена метрики также бывает полезна, когда проектировщику необходимо создать поток

данных через определенную область (или повлиять на такой поток). Используя динамическую природу протокола, можно избегать работы в аварийных областях и интенсивнее использовать наиболее удобные области.

Информация, которая необходима алгоритму для вычисления кратчайшего пути, распространяется посредством рассылки обновлений. Такие обновления отправляются всем маршрутизаторам области; они помогают устройствам создать общую картину окружения в каждый момент времени.

Обновления OSPF

В сетях OSPF обновления периодически рассылаются всем маршрутизаторам, входящим в одну область. В отличие от обновлений протоколов вектора расстояний (в каждом обновлении содержится вся таблица маршрутов), обновление OSPF содержит только ту часть таблицы, которая относится к непосредственным соседям отправляющего ее маршрутизатора.

В OSPF-области возможны пять типов обновлений, каждый из которых имеет особую цель. Типы обновлений перечислены ниже:

- Hello-сообщения
- Описание базы данных
- Запрос состояния каналов
- Обновление состояния каналов
- Подтверждение приема сообщения о состоянии каналов

Для построения наиболее точной картины сети маршрутизаторы OSPF используют комбинацию обновлений. Обновления любого типа посылаются с маршрутизатора на маршрутизатор в виде OSPF-пакетов. Заголовок пакета содержит информацию, относящуюся к типу пакета. Поля заголовка OSPF-пакета представлены на рис. 17.5.

Примечание

На рис. 17.5 вы видите, что в заголовке OSPF-пакета присутствует множество полей. Эта книга для начинающих, поэтому мы не будем подробно рассматривать все поля, поговорим лишь о тех, которые непосредственно влияют на ситуации, рассматриваемые нами в качестве примеров.

Второе поле заголовка пакета указывает тип обновления. По этому полю маршрутизатор, принимающий пакет, может сразу же определить, является ли пакет hello-сообщением или же обновлением другого типа.

Маршрутизаторы OSPF используют сообщения первого типа, hello-сообщения, для определения текущего состояния своих каналов. Маршрутизаторы регулярно рассылают hello-пакеты всем соседям, с которыми они связаны напрямую. Если маршрутизатор получает hello-со-

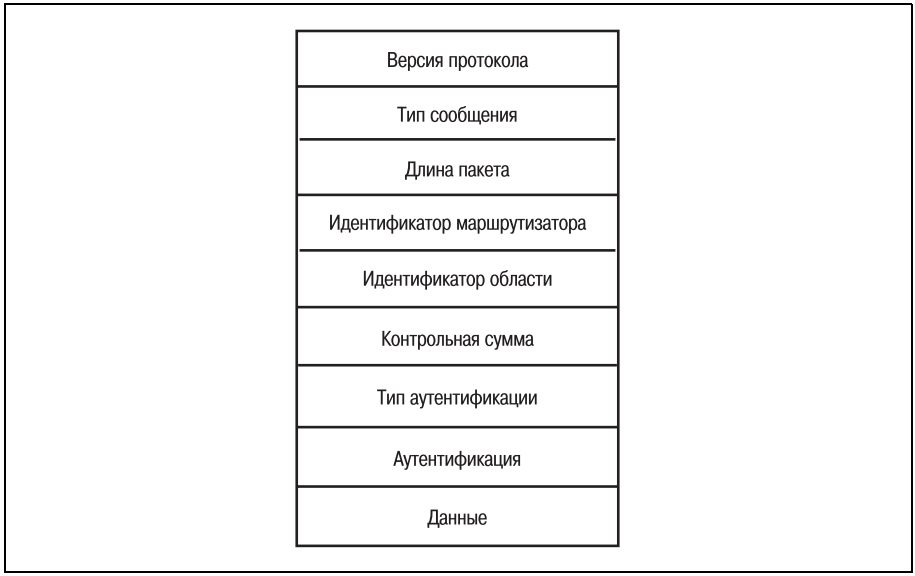

***Рис. 17.5.** OSPF-пакет*

общение от соседа, он может считать, что канал между ними функционирует. Если же в течение некоторого заданного промежутка времени hello-пакет не получен, то канал помечается для удаления.

Второй тип сообщений – это описание базы данных. Описание базы данных – это особый вид обновлений, использующийся двумя маршрутизаторами, находящимися в отношении смежности. Когда два маршрутизатора одной области имеют идентичные топологические базы данных, выделенный маршрутизатор (DR) может пометить их как смежные. Два маршрутизатора, находящихся в отношении смежности, обмениваются описаниями базы данных, чтобы обеспечить синхронизацию имеющейся у них информации.

Третий тип обновлений, запрос состояния каналов, маршрутизаторы используют для запуска процесса затопления сообщениями о состоянии каналов прежде, чем истечет таймер обновлений. Некоторые маршрутизаторы отправляют запросы состояния каналов при включении питания. Запрос вызывает поступление обновлений, которые помогают новому маршрутизатору построить свои таблицы маршрутов.

Четвертый тип обновлений – это обновление состояния каналов. Этот тип используется для затопления сети топологическими данными. Каждый маршрутизатор сети OSPF использует обновление состояния каналов, чтобы иметь наиболее точное представление о среде.

Последний тип обновлений – это подтверждения приема сообщений о состоянии каналов. Все обновления состояния каналов, отправляемые по сети, требуют отсылки отправляющему устройству уведомления о

получении. Маршрутизаторы сравнивают список уведомлений с таблицей маршрутов, чтобы проверить, все ли маршрутизаторы действительно работают. Многие из обновлений рассылаются только в пределах области. Помните, что в одной автономной системе может быть несколько областей. Поэтому, для того чтобы понять, как перемещаются по сети OSPF-пакеты, необходимо знать, как образуются области.

Области OSPF

Автономные системы OSPF разбиваются на области. Области – это небольшие сети внутри АС, имеющие одну топологическую базу данных. Протокольные адреса этих областей не зависят от АС, к которой они принадлежат, и от других областей. В автономной системе OSPF могут быть образованы области трех типов:

- Стандартная область
- Тупиковая область
- Не совсем тупиковая область

Примечание

Существует и четвертый тип области – магистральная. Но магистраль автономной системы OSPF на самом деле не является областью в том же смысле, как три основных типа. Магистраль просто создается из всех маршрутизаторов, не вошедших ни в одну из областей. Назначение магистрали состоит в соединении областей АС друг с другом и с любыми внешними сетями.

Прежде чем перейти к рассмотрению того, что делает область каждого типа и как они влияют на общую производительность сети OSPF, следует хорошо представлять себе смысл образования областей. Область диктует правила, полностью определяющие работу содержащихся в ней маршрутизаторов. Маршрутизаторы области имеют достоверную информацию (полученную из первоисточника) только друг о друге. Хотя бы один маршрутизатор области должен заниматься обеспечением связи области с другими областями или с магистралью (магистраль – это множество маршрутизаторов, оставшихся вне основных областей). Такой маршрутизатор называется пограничным маршрутизатором области (ABR).

Маршрутизаторы области имеют прямой доступ к сведениям только друг о друге. При необходимости отправить информацию в другую область они могут переслать пакеты на ABR, который доставит их за пределы области, при этом пакеты будут снабжены косвенными данными о среде маршрутизации. Используя эти данные, маршрутизаторы области могут правильно адресовать пакеты, направляемые в другие области той же АС.

Если области необходима связь с магистралью автономной системы, но она находится за другой областью, можно сконфигурировать виртуаль-

ный канал. Виртуальный канал – это канал, идущий от ABR к промежуточному ABR и завершающийся в магистральной области. (При создании виртуальных каналов необходимо придерживаться нескольких важных правил, о которых будет рассказано в разделе «Конфигурирование OSPF».) Пример виртуального канала изображен на рис. 17.6.

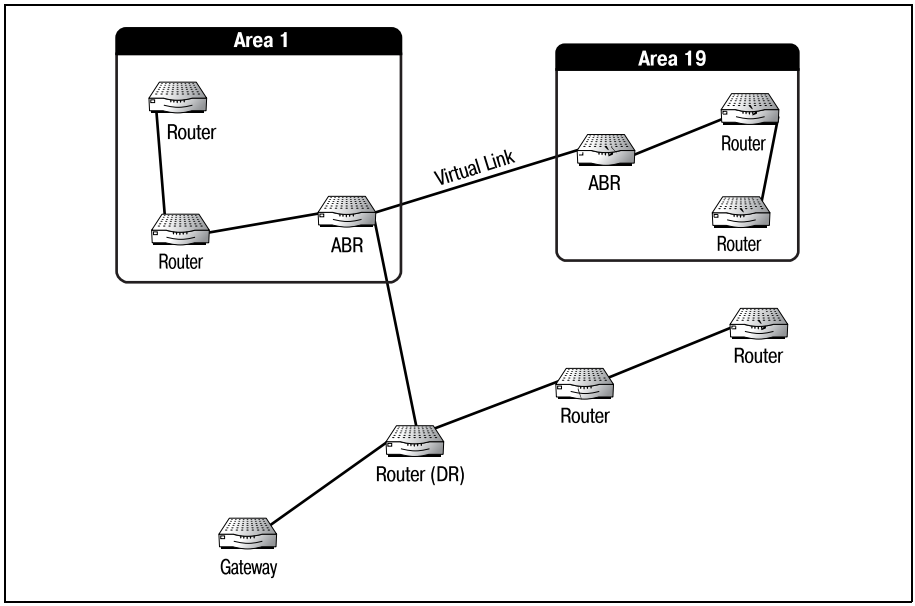

***Рис. 17.6**. Виртуальный канал*

На рис. 17.6 область 19 установила виртуальную связь с магистралью через область 1. Теперь маршрутизаторы области 19 могут отправлять пакеты прямо в магистральную область.

Примечание

При конфигурировании виртуального канала транзитная область не может быть тупиковой.

Маршрутизаторы области получают информацию о внешних маршрутах посредством такого процесса, как объявление маршрутов. Шлюз автономной области, используя EGP, собирает LSA-информацию от внешних автономных систем. Затем шлюз пересылает эту информацию через магистраль на пограничные маршрутизаторы области. ABR ретранслируют информацию о маршрутах всем маршрутизаторам области, которые используют ее для отправки данных за пределы автономной системы.

Рассылка маршрутов может быть обременительной для области из-за дополнительного трафика и множества сообщений LSA. Поэтому, если область непосредственно связана с магистралью AC, вы можете убе-

речь ее от лишнего трафика, связанного с объявлениями маршрутов, сделав область тупиковой.

Тупиковые области

Тупиковая область – это область, которая специально предназначена для того, чтобы не принимать информацию о маршрутах от областей, находящихся вне АС. Создание тупиковых областей может уменьшить трафик и освободить ресурсы, занятые под обработку дополнительных обновлений. Основным критерием при формировании тупиковой области является то, что она должна быть напрямую связана с магистралью автономной системы.

Маршрутизаторы магистрали имеют неявные сведения о внешних для автономной системы маршрутах. Поэтому маршрутизаторы тупиковой области, которым нужно отправить информацию по внешнему маршруту, могут пересылать такие пакеты в магистральную область, что может привести к дополнительной нагрузке на некоторые из наиболее занятых маршрутизаторов АС.

Не совсем тупиковые области (NSSA)

NSSA – это тупиковая область, в которой запрограммированы прямые пути к внешним автономным системам. Как и в тупиковых областях, маршрутизаторы внутри NSSA освобождены от дополнительного трафика по распространению информации о маршрутах. Однако они могут достичь некоторых внешних маршрутов, используя предопределенные маршруты. При конфигурировании маршрутизатора задается информация, необходимая для достижения некоторых внешних по отношению к АС маршрутов.

NSSA не должны быть прямо связаны с магистралью автономной системы. Так как маршрутизаторы обладают информацией, необходимой для правильной адресации пакетов, эти пакеты могут пересылаться через другие области. Но недостаток NSSA в том, что входящие в такую область маршрутизаторы могут достичь не любого внешнего маршрута, а только того, о котором у них имеются заранее введенные сведения.

Примечание

Несмотря на то что область, не соединенная непосредственно с магистралью, может функционировать, не рекомендуется создавать такие области. Многие документы Cisco настойчиво советуют не делать этого.

Чертой, отличающей тупиковые и не совсем тупиковые области от остальных, является их неспособность получать и обрабатывать информацию о маршрутах. Автономные системы передают сведения о своей внутренней структуре другим областям посредством процесса, носящего название «перераспределение маршрутов». Это процесс делает возможной связь между АС.

Перераспределение маршрутов

Перераспределение маршрутов происходит, когда пограничный маршрутизатор распространяет информацию о внешних путях. Это могут быть статические пути, пути, о которых сообщили другие маршрутизаторы, или пути, полученные от других протоколов внутреннего шлюза (таких как RIP). Перераспределение маршрутов автономной области позволяет устройствам из других АС общаться непосредственно с устройствами локальной АС.

Процесс перераспределения маршрутов доверен протоколу EGP (Exterior Gateway Protocol, протокол внешнего шлюза), соединяющему две или более автономные системы. В следующей главе, посвященной одному из таких протоколов, BGP (Border Gateway Protocol, протокол пограничного шлюза), перераспределение маршрутов будет описано более подробно.

В OSPF пользователи могут определить, какие маршруты они хотели бы объявлять внешним источникам и куда хотелось бы их отправлять. Пограничные шлюзы могут объявить внешним источникам все локальные маршруты, их часть или не объявлять ни один из маршрутов. В следующем разделе мы поговорим о конфигурировании маршрутизатора Cisco для работы в автономной области OSPF. В процессе конфигурирования будет затронут вопрос о перераспределении маршрутов.

Конфигурирование OSPF

Настроить маршрутизатор Cisco на работу с OSPF несложно. Для разрешения работы протокола на маршрутизаторе (как и для других протоколов, уже рассмотренных в данной книге) необходимо выполнить несколько простых команд. Изучение параметров, позволяющих протоколу оптимально использовать среду, может потребовать времени. OSPF содержит несколько команд, которые позволяют приспособить протокол для нужд большинства сетей.

По сравнению с сетями IGRP и EIGRP, автономные системы OSPF достаточно сложны и запутаны. Прежде чем вы займетесь конфигурированием маршрутизаторов, настойчиво рекомендуем вам нарисовать схему сети. В зависимости от роли маршрутизатора в АС существует несколько способов конфигурирования. Маршрутизаторы областей конфигурируются не так, как магистральные маршрутизаторы, и каждый из них имеет разные настройки, зависящие от типа используемых областей. Суть в том, что вы должны иметь четкое представление о том, как будет выглядеть среда и где будет находиться конфигурируемый маршрутизатор, прежде чем начать собственно конфигурирование.

При разрешении маршрутизатору работы с OSPF вам будет предложено ввести некоторую информацию. Как и в большей части других, уже

рассмотренных нами протоколов внутреннего шлюза, для обеспечения уникальности запрашивается идентификатор процесса. Идентификатор может быть числом в диапазоне от 1 до 65 535. Он используется в качестве параметра команды `router ospf`.

```
Router#configure terminal
Router(config)#router ospf 349
```

Примечание

На том маршрутизаторе, где вы собираетесь настраивать OSPF, должен работать IP. Если ни один интерфейс не использует IP, вы получите следующее сообщение об ошибке:

```
Router(config)#router ospf 349
OSPF: Could not allocate router id
```

Основой OSPF-идентификатора маршрутизатора служит его IP-адрес.

Маршрутизатор входит в режим конфигурирования. Следующий шаг зависит от того, находится ли конфигурируемый маршрутизатор в области или же на магистрали. Если маршрутизатор будет работать в области, необходимо ввести идентификатор области и IP-схему области.

Для задания этой информации используется команда `network`, которая имеет такой формат:

```
#network <ip address> <wildcard bits> area <area address>
```

Как видно из приведенной выше записи, команда `network` требует введения после IP-адреса подстановочной маски (wildcard bits). Подстановочная маска образуется при инвертировании маски сети. Вместе с IP-адресом сети она указывает диапазон адресов, доступных для области. Например, адрес сети класса C 225.65.34.0 использует подстановочную маску 0.0.0.255 для включения всех IP-адресов сети.

Создадим область с номером 1 и адресной схемой класса B:

```
Router(config-router)#network 198.56.0.0 255.255.0.0 area 1
Router(config-router)#^Z
```

Чтобы настроить маршрутизатор для работы на магистрали, нужно указать область 0. Кроме того, чтобы обеспечить магистральным маршрутизаторам возможность обработки всего потенциального трафика сети, необходимо задать диапазон IP-адресов от 0.0.0.0 до 255.255.255.255 (широковещательный диапазон IP). Следующая команда иллюстрирует задание такого диапазона IP-адресов:

```
Router(config-router)#network 0.0.0.0 255.255.255.255 area 0
Router(config-router)#^Z
```

Примечание

Магистраль сети OSPF – это всегда область 0.

Любой конфигурируемый пограничный маршрутизатор области должен иметь хотя бы один интерфейс, настроенный для работы с магистралью, и один – для области. Например, ABR для области 3 (некоторой области, непосредственно связанной с магистралью) должен иметь один интерфейс с IP-адресом из диапазона, приемлемого для области 3, и один интерфейс, связанный с магистралью. Тем самым обеспечивается корректная связь между всей областью и магистралью. В приведенной ниже последовательности команд сконфигурирован ABR для области 3:

```
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-if)#ip address 198.109.1.1 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#interface ethernet 1
Router(config-if)#ip address 197.85.1.1 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#^Z
Router(config)#router ospf 45
Router(config-router)#network 198.109.0.0 0.0.255.255 area 3
Router(config-router)#network 0.0.0.0 255.255.255.255 area 0
Router(config-router)#^Z
```

Интерфейс ethernet 0 подключается к области 3, а ethernet 1 – к магистрали. Маршрутизатор настроен как ABR, который будет перемещать пакеты между областью и магистралью.

Для конфигурирования тупиковой или не совсем тупиковой области необходимо применить еще несколько команд. Для создания как тупиковой области, так и NSSA используется команда `area`. Команда `area` выполняется на ABR, отделяющем область от магистрали, и имеет следующий формат:

```
#area <area number> <area type> <area specific params>
```

Если вы хотите настроить ABR из предыдущего примера так, чтобы область 3 стала тупиковой, нужно использовать следующие команды:

```
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-if)#ip address 198.109.1.1 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#interface ethernet 1
Router(config-if)#ip address 197.85.1.1 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#^Z
Router(config)#router ospf 45
Router(config-router)#network 198.109.0.0 0.0.255.255 area 3
Router(config-router)#network 0.0.0.0 255.255.255.255 area 0
Router(config-router)#area 3 stub no-summary
Router(config-router)#^Z
```

Давайте посмотрим на некоторые параметры команд, приведенных выше. Параметр `no-summary` указывает, что в тупик не должны отправляться сводки IP-маршрутов. Тем самым максимально ограничивается трафик внутри тупиковой области. Единственный параметр, который может использоваться в NSSA, но не в тупиковых областях, – это `no-redistribution`.

Команда `area` используется также для создания виртуальных каналов между областями и магистралью. Виртуальные каналы позволяют областям общаться непосредственно с магистралью (через транзитную область). ABR одной области виртуально связывается с ABR транзитной области (которая напрямую связана с магистралью). Возникает прозрачная виртуальная связь с магистралью. Синтаксис создания виртуального канала таков:

```
#area <local area ID> virtual-link <neighbor ABR address> <optional
parameters>
```

Используем следующую команду для создания виртуального канала для области 3:

```
Router(config)#router ospf 45
Router(config-router)#area 3 virtual-link 123.1.1.2
Router(config-router)#^Z
```

На некоторых маршрутизаторах может возникнуть необходимость изменить значение метрики по умолчанию для какого-то канала. OSPF автоматически вычисляет метрику маршрута, деля 10^8 на пропускную способность канала. Знайте, что в результате такого вычисления не всегда получается желаемый результат.

Наибольшая пропускная способность, к которой может применяться метрика OSPF, соответствует оптоволоконному (FDDI) каналу. Такой канал всегда имеет метрику 1. Если несколько интерфейсов маршрутизатора подключено к FDDI-каналам или если пропускная способность выше, чем у FDDI, то можно уточнить формулу вычисления метрики. Для изменения значения, используемого для вычисления метрики по умолчанию, используйте команду `auto-cost`, как в приведенном ниже примере:

```
Router(config)#router ospf 45
Router(config-router)#auto-cost reference-bandwidth 4294967
Router(config-router)#^Z
```

Примечание

Значение, указываемое для автоматического вычисления метрики, представляет собой максимальную скорость передачи данных в Мбит/с.[1] Значение должно принадлежать диапазону от 1 до 4 294 967.

[1] Значение по умолчанию равно 100. – *Примеч. науч. ред.*

Хотя существует еще множество необязательных параметров, которые можно использовать для настройки OSPF, в этой главе рассмотрены только те, которые необходимо знать начинающему администратору Cisco. Описанные команды являются ключевыми при разрешении и установке различных возможностей.

Резюме

В этой главе приведены основные термины и команды наиболее популярного протокола маршрутизации, используемого в больших сетях, – OSPF. Протокол OSPF гибко подходит к созданию областей и организации потоков данных. В главе были рассмотрены следующие темы:

- OSPF – это протокол состояния канала.
- Протоколы состояния канала обеспечивают более быструю сходимость, чем протоколы вектора расстояния.
- Среда работы протокола OSPF – это автономная система (AC).
- Автономная система OSPF разбивается на области.
- Области OSPF могут быть тупиковыми и не совсем тупиковыми (NSSA).

Вопросы и ответы

Вопрос Почему протоколы состояния канала обеспечивают более быструю сходимость, чем протоколы вектора расстояния?

Ответ Протоколы состояния канала хранят подробную информацию о маршрутах только для непосредственных соседей маршрутизатора, в то время как протоколы вектора расстояния обмениваются таблицами маршрутов со всеми маршрутизаторами среды. Благодаря тому что количество обновлений протоколов состояния канала ограничено, сходимость достигается быстрее.

Тест

Вопросы

1. Что такое тупиковая область?
2. По какой формуле вычисляется метрика OSPF по умолчанию?
3. Какой параметр необходим для разрешения работы OSPF?

Ответы

1. Область, которая не получает перераспределенные маршруты.
2. Стоимость = 10^8/пропускная способность канала.
3. Идентификатор процесса (необходимо, чтобы на устройстве был настроен IP).

Упражнение

1. Настройте OSPF на пограничном маршрутизаторе области, находящемся между не совсем тупиковой областью (область 1) и магистралью. Схема IP-адресации области – 10.0.0.0, адреса интерфейсов 10.1.1.1 и 10.20.1.1.

Решение

```
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-if)#ip address 10.1.1.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#interface ethernet 1
Router(config-if)#ip address 10.20.1.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#^Z
Router(config)#router ospf 572
Router(config-router)#network 10.0.0.0 0.255.255.255 area 1
Router(config-router)#network 0.0.0.0 255.255.255.255 area 0
Router(config-router)#area 1 nssa
Router(config-router)#^Z
```

18

Введение в BGP

К этому моменту вы уже должны иметь достаточно прочные базовые знания в области Cisco. Поэтому новая глава посвящена более сложной теме – протоколу BGP (Border Gateway Protocol, протокол граничного шлюза). Один из наиболее популярных (и, наверное, самый важный) среди используемых сегодня протоколов маршрутизации, BGP, благодаря своей устойчивости стал основой Интернета. Основное назначение BGP состоит в объявлении существования сети (и ее структуры) другим BGP-маршрутизаторам, в частности маршрутизаторам провайдера интернет-услуг.

В современном мире маршрутизации сведущие специалисты в области конфигурирования и поддержки BGP-сред очень востребованы. Можно без преувеличений сказать, что именно эти специалисты приводят в движение Интернет. Поэтому чем лучше вы разберетесь в том, что такое Интернет и как он работает с маршрутизаторами Cisco и с протоколом BGP, тем лучше вы будете подготовлены к работе инженера в реальном мире.

Интернет – это гигантское объединение разнородных и не похожих друг на друга систем, архитектур и протоколов. Представьте себе большую вечеринку, на которую приезжают гости из разных стран, каждый из которых говорит на своем языке: так и в Интернете работает множество различных типов компьютеров и сетевых систем. И вот на вечеринке звонит телефон. Официант записывает сообщение, ставит в начале имя того, кому оно адресовано, и передает его кому-нибудь из толпы, надеясь, что сообщение достигнет адресата. Если никто из гостей не говорит на нескольких языках, то сообщение вряд ли уйдет далеко. Оно может быть отброшено уже первым же получившим его человеком.

У сообщения было бы больше шансов оказаться доставленным, если бы на вечеринке был переводчик, знающий достаточное количество слов каждого из языков, на которых говорят гости, чтобы понять, на каком из них написано имя адресата. Тогда сообщение, переданное одному из гостей, без труда нашло бы своего хозяина. Даже если человек, у которого сообщение находилось в какой-то момент, не мог распознать имя получателя, он обратился бы к переводчику, и тот помог бы найти гостя, который мог бы знать адресата. В Интернете таким переводчиком служит BGP.

Протокол BGP ломает языковые барьеры, стоящие между разнородными системами Интернета. Этот протокол позволяет системам и сетям распознавать не родственные им системы (и быть распознанными ими). Системы, работающие с BGP, могут объявлять свои маршруты и структуры другим системам вне зависимости от архитектуры сетей, участвующих в передаче данных.

Примечание

Протокол, который известен всем как BGP, – это на самом деле BGP4 (о нем мы и поговорим в данной главе). Эта четвертая реализация протокола очень отличается от всех предыдущих версий. BGP4 был принят интернет-провайдерами и в наше время является наиболее популярной формой BGP, поэтому его часто называют просто BGP.

Как вы видели, большая часть протоколов маршрутизации занимается маршрутизацией данных внутри некоторой сетевой среды. Такие протоколы, как RIP, используются для внутренней маршрутизации данных в сетях и системах. Они позволяют пользователям сети обмениваться данными с другими пользователями, находящимися в той же области. А вот BGP переправляет данные из одной сети в другую (из одной автономной системы (AC) в другую).

AC не имеет врожденных знаний обо всех остальных системах, существующих в мире. Это и понятно, ведь для того чтобы каждая сеть автоматически знала топологию любой другой сети в мире, необходима какая-то сверхмощная технология, а такая технология пока еще не разработана. Так как одна система не знает топологии любой другой системы, то обмен данными между ними может быть затруднителен.

Протокол BGP (BGP4) позволяет системам, не обладающим знаниями о других системах, свободно общаться друг с другом. Но если бы все было так просто, то наша глава уже могла бы заканчиваться. Изучение BGP считается темой повышенного уровня, так как существует множество переменных и параметров, которые и делают BGP тем, чем он является. Так как вы работаете с маршрутизаторами Cisco, вам необходимо уметь различать и понимать все эти тонкости.

BGP фактически состоит из двух протоколов: EBGP (Exterior Border Gateway Protocol, внешний протокол граничного шлюза) и IBGP (Inte-

rior Border Gateway Protocol, внутренний протокол граничного шлюза). EBGP используется для пересылки данных от одной автономной системы к другим (именно так большинство людей представляет себе BGP), в то время как IBGP занимается маршрутизацией данных внутри одной AC.

Прежде чем погрузиться в конфигурирование маршрутизатора Cisco для работы с BGP, необходимо обозначить и обсудить термины и параметры, использующиеся в BGP-средах. В главе будет приведено множество примеров конфигурирования, которые помогут вам представить, как работает BGP.

Автономные системы

Автономные системы (AC) – это основа BGP-сетей. AC является базовой единицей маршрутизации протокола BGP.

Примечание

Автономная система может быть отдельной сетью или группой связанных сетей (как глобальная сеть).

Примечание

Каждый протокол маршрутизации работает со своей стандартной базовой единицей. Например, PNNI использует домены (см. приложение A), а IS-IS – одноранговые группы (см. главу 19). По существу, все эти единицы и группы одинаковы. Поэтому не пугайтесь изменений в терминологии. Для одних и тех же сетевых сред придумано множество названий для обозначения различных групп оборудования.

Протоколы могут по-разному работать с сетями, автономными системами, одноранговыми группами и доменами, но сама единица при этом не меняется.

Чтобы понять автономные системы, давайте начнем с того, что посмотрим, как они образуются физически. Затем мы поговорим о том, как BGP их различает. AC – это связанная группа систем. То есть, как и локальная или глобальная сеть, AC – это сетевая среда. Типичная AC изображена на рис. 18.1.

Сеть, представленная на рис. 18.1, – это небольшая локальная сеть, разделенная на несколько подсетей. Эти подсети находятся в географически удаленных друг от друга областях и обмениваются информацией с главным офисом. Доступ во внешний мир осуществляется только через главную подсеть (1).

Так как маленькие (2 и 3) подсети не имеют отдельного соединения с Интернетом (они получают доступ к нему только через центральный офис), они считаются частью той же AC. То есть все три подсети образуют одну автономную систему.

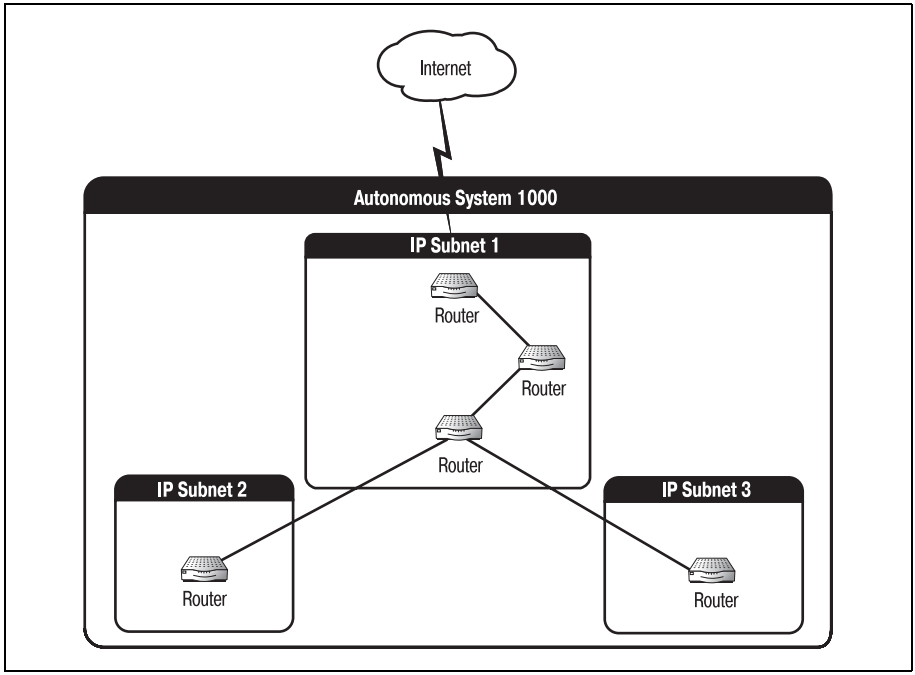

***Рис. 18.1.** Автономная система*

Автономная система может быть еще более элементарной, чем представленная на рис. 18.1. В небольшой сети, такой как POP (point of presence – точка присутствия), может быть всего один или два мощных маршрутизатора (вероятнее всего, маршрутизаторы Cisco масштаба предприятия). На рис. 18.2 изображена маленькая сеть с одним маршрутизатором, представляющая собой АС.

Сеть на рис. 18.2 невелика, но она является настоящей автономной системой. Один маршрутизатор, соединяющий POP-серверы с центральным офисом, делает POP автономной системой.

Как же образуется АС? Сами маршрутизаторы не превращают сеть в автономную систему. Если бы это было так, то практически каждая сеть в стране (LAN, WAN или SOHO) была бы автономной системой, что, конечно же, неправда. Существует пара требований, которые сеть должна выполнять для того, чтобы быть автономной системой.

Первое условие – это соединение с Интернетом. Для большинства компаний это означает наличие канала к интернет-провайдеру. Учитывая, что BGP обеспечивает эффективную связь между системами, необходим соответствующий канал связи. Данное условие логично и элементарно, но оно связано со вторым.

Вторым условием существования автономной системы является наличие у нее номера ASN (autonomous system number – номер автономной

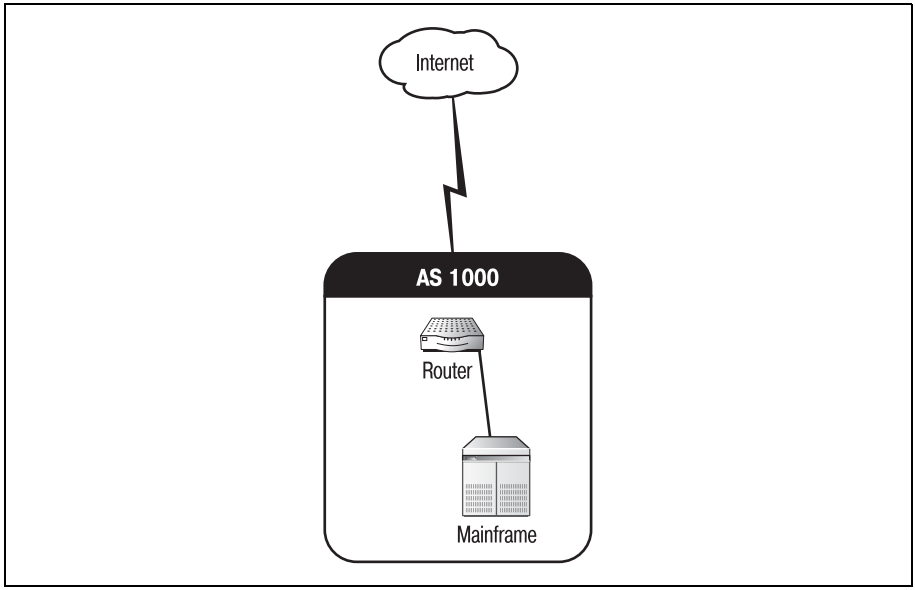

Рис. 18.2. *Автономная система с одним маршрутизатором*

системы). Номер автономной системы для сети – это ее «паспорт» в Интернете (то, что отличает ее от других BGP-систем). ASN принадлежит диапазону от 1 до 65 535.

Когда вы начнете работать с большими средами Cisco и проектировать их, вам может понадобиться получить и присвоить ASN сети BGP. Этот процесс не очень сложен, но заслуживает того, чтобы описать его в следующем разделе.

Получение номера автономной системы

Номер автономной системы, как и сетевой адрес IP, должен быть получен от административного органа (если предполагается, что вам понадобится общаться с внешним миром по EBGP). В большинстве случаев ваш интернет-провайдер присваивает вам ASN, являющийся подмножеством его собственного номера. Очевидно, что раз номера автономных систем принадлежат диапазону 1–65 535, то присвоено может быть лишь ограниченное количество номеров. Поэтому только небольшой диапазон адресов был оставлен для частного использования.

Примечание

Использование и присваивание номеров автономных систем тесно связано с теми же операциями для IP-адресов. Оба типа номеров присваиваются административными органами, оба имеют диапазоны для общественного и частного использования и (к сожалению) имеют ограниченное количество значений.

Примечание

В США регистрацией и выдачей номеров автономных систем занимается организация ARIN (American Registry for Internet Numbers – Американский реестр интернет-номеров). Перечень выданных номеров автономных систем и их владельцев можно найти по адресу: *http://rs.arin.net/netinfo/asn.txt*.

Частные ASN относятся к диапазону 64 512 – 65 535. Как и частные IP-адреса, они не могут быть объявлены в Интернете. Эти номера используются для IBGP-маршрутизации внутри большой сети BGP. Номера из частного диапазона могут свободно использоваться кем угодно.

Существуют три типа автономных систем. Нужен ли вашей сети ASN для общественного использования, зависит от типа вашей автономной системы. АС может быть:

- Тупиковой
- Многопортовой
- Транзитной

Чтобы иметь право получить общественный ASN, сеть должна быть многопортовой автономной системой.

Тупиковая АС – это сеть, имеющая только одно соединение с Интернетом. Тупиковые автономные системы обычно рассматриваются как расширение более крупной АС. Так как существует всего один маршрут в тупиковую АС и из нее, то не требуются никакие дополнительные действия. Также нелогично поддерживать большой список маршрутов BGP на шлюзе, которому доступен всего один маршрут. На рис. 18.3 изображена тупиковая сеть, подключенная к интернет-провайдеру.

Примечание

Многие автономные системы относятся к категории тупиковых. Но возможна ситуация, когда к большой сети присоединены несколько тупиковых сетей и они считаются частью одной автономной системы. Таким тупиковым автономным системам не нужен собственный частный ASN. Они могут рассматриваться как часть крупной АС и использовать ее номер.

Многопортовые системы – это сети, имеющие несколько связей с внешними автономными системами. Многопортовая АС принимает маршрутизируемую информацию от всех систем, с которыми она связана, но маршрутизирует только свои внутренние данные. На рис. 18.4 показана многопортовая автономная система.

Автономная система 1000 на рис. 18.4 является многопортовой, так как она связана с АС 2000 и АС 3000. Так как многопортовая АС маршрутизирует только внутренние данные, то АС 2000 не сможет переслать информацию АС 3000 через АС 1000. (АС 1000 принимает и отправляет только данные, имеющие отношение к ее внутренней се-

ти.) Для того чтобы взаимодействовать с AC 3000, AC 2000 должен иметь собственный канал связи с AC 3000.

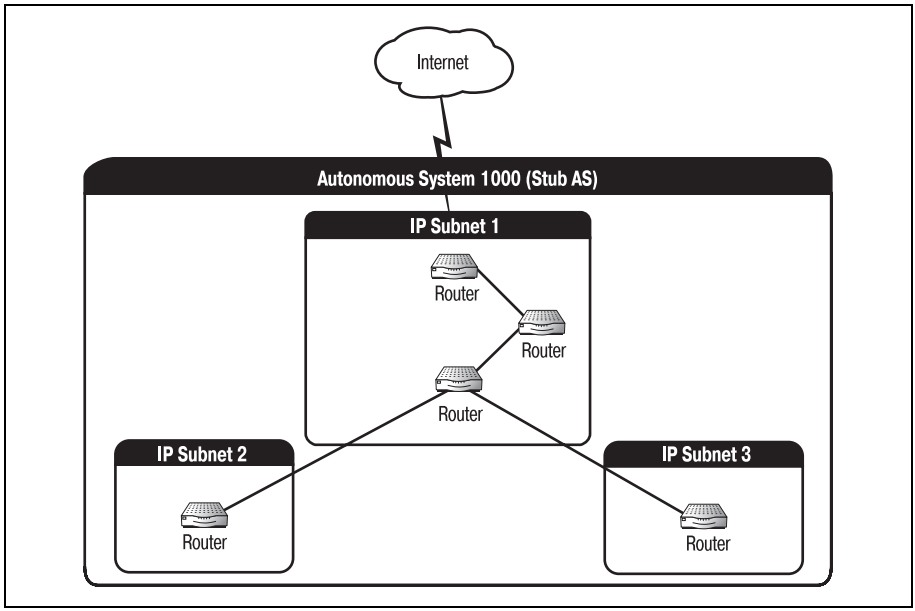

Рис. 18.3. *Тупиковая AC*

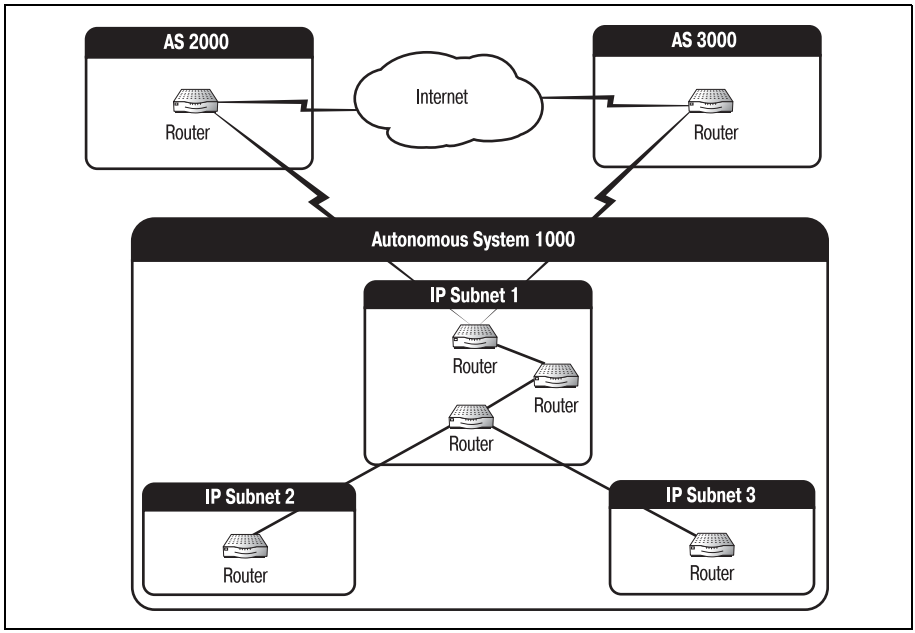

Рис. 18.4. *Многопортовая AC*

Третий тип автономных систем – транзитные АС. Транзитные автономные системы – это многопортовые системы, которые принимают и передают информацию от других внешних автономных систем. Если бы многопортовая АС на рис. 18.4 была транзитной, то АС 2000 мог бы отправлять данные АС 3000 через АС 1000.

Примечание

Об использовании номеров автономных систем и их диапазонах можно прочитать в RFC 1930.

ASN и IP-адреса

Связь между номерами автономных систем и IP-адресами заключается не только в схожих правилах использования, она гораздо глубже. IP имеет важное значение для функционирования BGP и образования автономных систем. Номер ASN непосредственно связан с IP-адресами той автономной системы, которой он принадлежит. Другими словами, ASN должен быть поставлен в соответствие тем сегментам IP-сети, которым он назначается. Тем самым обеспечивается направление нужного трафика в надлежащую автономную систему.

На рис. 18.5 изображена сеть, которая вскоре станет автономной системой. На рисунке указаны адреса всех сегментов IP-сети.

Шлюз будет отправлять данные только на IP-адреса, специально назначенные для ASN. Конфигурирование правильного IP-адреса для кон-

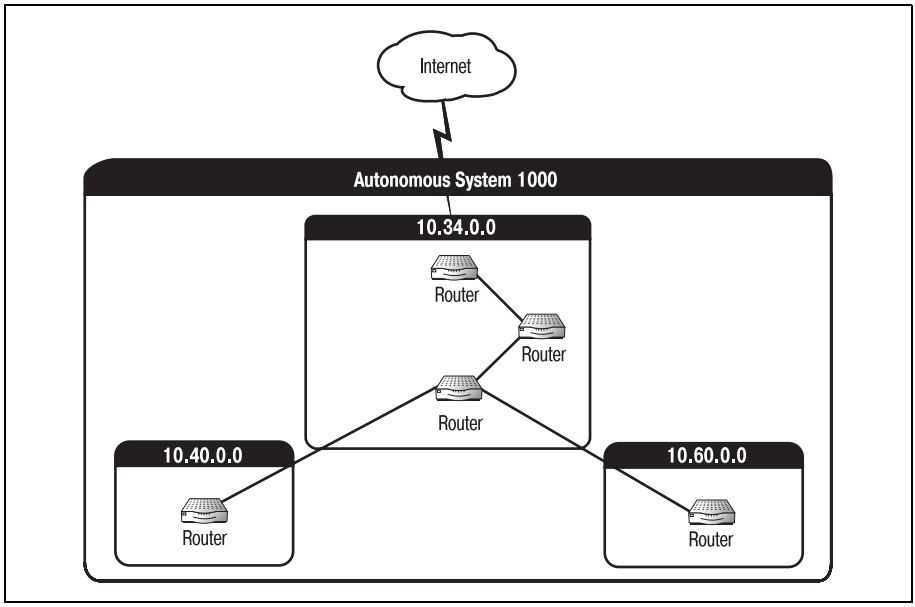

Рис. 18.5. *Сегментированная IP-сеть*

кретного ASN обеспечивает отправку соответствующих данных на ваши пограничные шлюзы со стороны любых внешних автономных систем. (Это правило справедливо для всех типов АС.) Если вы исключите какую-то подсеть или измените схему адресации, не изменив ASN, то сеть не сможет корректно принимать предназначенные для нее данные.

Поговорив о том, что же такое автономные системы, мы можем заняться изучением того, как они работают. В оставшейся части главы будут представлены и описаны части автономной системы. Три основные составляющие автономной системы – это BGP-узел (BGP speaker), пограничный шлюз и BGP-сосед (BGP peer).

BGP-узлы

Все маршрутизаторы АС (настроенные на работу с BGP) называются BGP-узлами. На BGP-узлах должен быть сконфигурирован ASN той автономной системы, к которой они принадлежат. Если АС состоит из более чем одной IP-подсети, то все IP-сети должны быть сопоставлены ASN.

Если BGP-узел автономной системы имеет IP-адрес, не связанный с ASN, то узел не сможет работать в АС. Следовательно, любые другие системы или BGP-узлы, находящиеся за не участвующим в работе узлом, также не смогут получать данные от остальной части АС. Автономная система, один из узлов которой имеет IP-адрес, не входящий в заданные для ASN диапазоны, представлена на рис. 18.6.

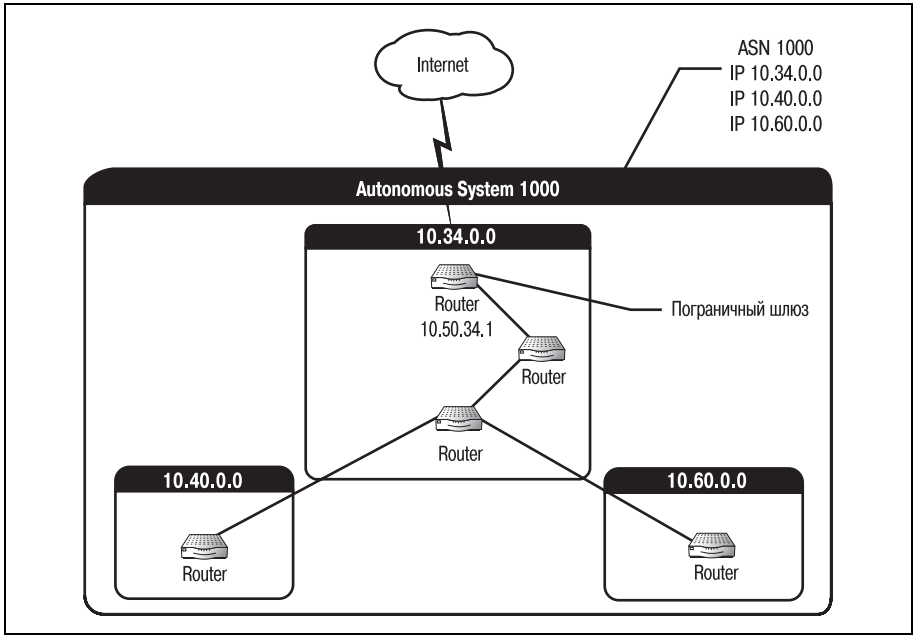

Рис. 18.6. *Не полностью функциональная АС*

На рис. 18.6 изображены два BGP-узла (и подключенные к ним оконечные системы), расположенные позади узла 10.50.34.1, которые не могут участвовать в работе АС. Несмотря на то что их диапазоны IP-адресов сопоставлены ASN системы, данные не могут пройти через узел 10.50.34.1, так как его адрес не задан для ASN.

Команда для разрешения использования BGP на маршрутизаторе (создания BGP-узла), как и для других протоколов маршрутизации, которые уже были рассмотрены, – это команда `router` с параметром:

```
Router(config)#router bgp
```

Однако это еще не полная команда. Для того чтобы BGP-узел начал работать, необходимо присвоить ему ASN. Правильная командная структура выглядит так:

```
Router(config)#router bgp 400
```

Хотя существуют и другие команды для включения всех функций BGP-узла, но команда `router BGP` все же самая первая (о других командах мы поговорим в следующих разделах).

Пограничные шлюзы

BGP-узлы, которые расположены между двумя или более автономными системами, называются пограничными шлюзами. Пограничные шлюзы действуют как мосты между автономными системами и являются еще одним маршрутом для информации, перемещаемой между системами. На рис. 18.7 изображена автономная система с пограничным шлюзом.

В автономной системе присутствие пограничного шлюза не обязательно. Пограничный шлюз необходим, только если АС использует EBGP для связи с другими автономными системами в Интернете. И, наоборот, одна автономная система может иметь несколько пограничных шлюзов. Если автономная система взаимодействует с несколькими внешними АС на нескольких BGP-узлах, то АС имеет несколько пограничных шлюзов.

Задача пограничного шлюза заключается в объявлении автономной системы (и любых других маршрутов, о которых он знает) всем внешним BGP-узлам, с которыми он связывается. Это описание может показаться слишком общим, но дочитайте главу до конца, и вы все поймете.

Сессии одноранговой обмена информацией (BGP–соседи)

Как и все остальные протоколы маршрутизации, BGP работает, распространяя информацию о маршрутах среди всех участников сети. То есть каждый BGP-узел для обеспечения успешной маршрутизации

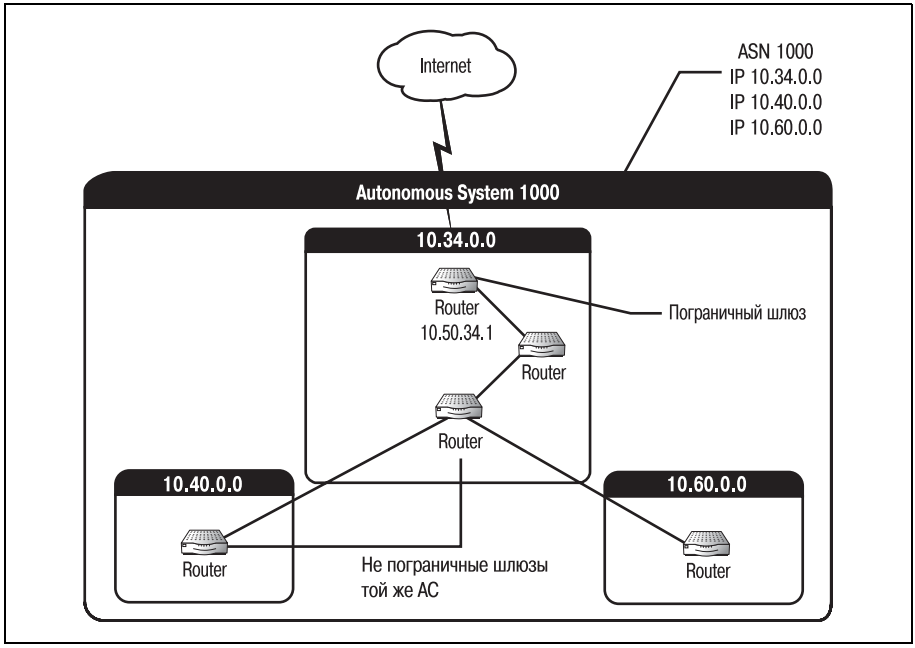

Рис. 18.7. *Автономная система с пограничным шлюзом*

данных должен обмениваться маршрутной информацией (о топологии и метриках) с другими BGP-узлами.

Такой обмен информацией о маршрутах происходит во время *BGP-сессий однорангового обмена информацией* (BGP peering sessions). Когда маршрутизатор BGP готов к обмену маршрутной информацией с другим BGP-узлом, он открывает сессию однорангового обмена со вторым узлом. Два BGP-узла, между которыми установлено такое соединение, называются BGP-соседями.

Примечание

Так как BGP в качестве базового протокола использует TCP, то для всех сессий однорангового обмена используется TCP-порт 179.

Для корректной работы BGP необходимо, чтобы BGP-узел был BGP-соседом для всех остальных узлов автономной системы. То есть BGP-узлы должны образовывать логическую *маршрутную сетку*. В зависимости от количества BGP-узлов в вашей АС эта сетка может быть достаточно большой и трудно контролируемой.

Примечание

Логическая сетка маршрутов – это конфигурация, в которой все маршрутизаторы среды имеют логическую (не обязательно физическую) связь друг с другом.

BGP-соседи не обязательно должны быть непосредственно связаны друг с другом (если они используют EBGP, а не IBGP), но они должны взаимодействовать друг с другом. Следовательно, для установления соединения между двумя BGP-узлами необходимо наличие стандартного пути между ними.

Примечание

BGP может устанавливать сессию одноранговой обмена между двумя маршрутизаторами, не связанными напрямую. Такие сессии называются сессиями *многошагового обмена EBGP* (multihop peering). Используя возможности EBGP, BGP-узел может установить соединение с узлами, находящимися от него в нескольких переходах.

Однако наиболее распространенной (и наименее сложной) формой одноранговой обмена является обмен информацией между двумя непосредственно связанными узлами.

Для установления соединения между двумя BGP-узлами используйте команду `neighbor`:

```
Router(configure)#router bgp 65000
Router(configure-router)#neighbor 10.115.48.1 remote-as 100
```

Вместе две эти команды конфигурируют маршрутизатор как BGP-узел автономной системы с номером 65000 и устанавливают сессию одноранговой обмена информацией с маршрутизатором 10.115.48.1, находящимся в AC 100. Для создания полнофункциональной связи между двумя маршрутизаторами необходимо повторить те же команды на 10.115.48.1 в отношении вашего маршрутизатора.

Приведенный выше пример предполагает, что между маршрутизаторами существует физическое соединение. Если же вы хотите установить многошаговое EBGP-соединение, используйте такие команды:

```
Router(configure)#router bgp 65000
Router(configure-router)#neighbor 10.115.48.1 remote-as 100
Router(configure-router)#neighbor 10.115.48.1 ebgp-multihop
```

После того как команды будут выполнены на соседнем маршрутизаторе, двум EBGP-маршрутизаторам, ставшим соседями, будет разрешен полный обмен информацией. Они передают друг другу данные о маршрутах, содержащиеся во внутренних системах, и обновляют информацию за счет полученных от соседа данных.

Примечание

Помните, что поскольку два маршрутизатора в EBGP-сценарии физически не связаны друг с другом, они должны взаимодействовать по IGP. Для этого подойдет любой, соответствующий ситуации протокол внутреннего шлюза, например OSPF или IGRP.

После открытия сессии обмена информацией BGP-узлы обмениваются полными таблицами маршрутов. Это может значительно растянуть процесс достижения конвергенции. Если количество объявляемых

каждым устройством маршрутов велико, то первое обновление может быть очень объемным. BGP-узлы могут хранить сотни и тысячи объявляемых маршрутов. И все эти маршруты должны войти в первое обновление, отправляемое новому BGP-узлу.

Примечание

BGP-соседи обмениваются полными таблицами маршрутов всякий раз, когда соседство устанавливается впервые. Это относится и к перезапускам. При каждом перезапуске маршрутизатор обменивается полной копией своих таблиц маршрутов со всеми своими соседями.

Минимизация количества перезагрузок маршрутизаторов снижает нагрузку на сеть (которая иначе оказывается перегруженной трафиком обмена полными таблицами BGP).

Однако все последующие обновления, отправляемые существующим BGP-соседям, состоят только из изменений таблицы маршрутов. Чем дольше маршрутизатор работает в автономной системе, тем менее интенсивным будет обмен таблицами.

После того как на BGP-узле сделаны настройки для маршрутизации, он готов к отправке и получению данных. В зависимости от потребностей сети узел использует один из двух протоколов: IBGP или EBGP. В следующем разделе мы поговорим об особенностях маршрутизации EBGP (IBGP будет описан чуть позже в этой же главе).

Конфигурирование протокола EBGP

Этот раздел посвящен протоколу EBGP (Exterior Border Gateway Protocol – внешний протокол граничного шлюза). Протокол EBGP устанавливает соединение между BGP-узлами, находящимися в разных автономных системах. Он может применяться для связи между интернет-провайдером и точкой присутствия или большим предприятием и несколькими поставщиками услуг связи. Функциональных отличий между EBGP и IBGP немного, но они достаточно глубоки для того, чтобы рассказать о протоколах по отдельности. (IBGP будет рассмотрен в следующем разделе.)

Маршрутизаторы BGP узнают о том, что их окружает, от других маршрутизаторов BGP. То есть во время сессий BGP-маршрутизаторы сообщают всем соседям пути, которые им известны. Эти маршруты составляют основу работы BGP, особенно для узлов, находящихся в разных автономных системах. Ключом к ровной (иногда) и быстрой работе Интернета служит способность BGP организовывать взаимодействие маршрутизаторов и обмен маршрутами между узлами разнородных сетей.

Для того чтобы BGP-узел, находящийся вне вашей АС, мог успешно маршрутизировать ваши данные, он должен знать о вашем местополо-

жении и о том, какой адрес вы представляете. Но маршруты – это не только адреса сетей внутри АС. BGP-узлы часто объявляют известные им маршруты к другим автономным системам.

BGP-маршрут просто сообщает: «Я знаю, как доставить информацию, предназначенную для XXX.XXX.XXX.XXX, отсюда туда». Маршруты указывают IP-адрес сети или адрес следующего маршрутизатора, если сеть не входит в АС отправляющего устройства. Адрес маршрутизатора, являющегося следующим переходом на пути достижения цели, – это адрес пограничного шлюза, который нужно использовать. Эта информация позволяет BGP-узлам объявлять маршруты, которые не связаны с ними непосредственно. Благодаря этому данные из разных автономных систем могут достигать практически любого адресата.

Примечание

К сожалению, часто встречается такая ошибка, как неправильное объявление маршрута. Например, если администратор неправильно введет номер сети, в объявлении будет присутствовать чужой IP-адрес и в результате трафик не попадет в объявленную сеть.

Маршрутизатор BGP, объявляющий маршрут, будет получать весь трафик, предназначенный для такого адреса. Однако на самом деле маршрутизатор не знает, как отправить данные по этому адресу, и ничто в АС не соответствует данному адресу, поэтому вся полученная информация будет отброшена. Но вся информация, относящаяся к правильно объявленным этим маршрутизатором путям, будет передана без задержек.

Когда у BGP-узла появляется информация о маршруте, которую необходимо отправить (как обновление) другим BGP-узлам (внешним или внутренним), он объявляет маршрут. Объявление (как и можно было предположить) – это тот способ, которым один узел предлагает другим узлам маршруты или обновления маршрутов.

Самой распространенной формой объявления маршрутов BGP является перераспределение маршрутов. Давайте посмотрим, как это работает. На рис. 18.8 изображены четыре автономные системы, связанные по EBGP.

На этом рисунке маршрутизатор автономной системы 2000 объявляет (автономным системам 3000 и 4000), что он умеет направлять весь трафик для IP-сети 10.34.0.0 в АС 1000 и весь трафик для IP-сети 10.60.0.0 в саму АС 2000. Он также объявляет (маршрутизатору 1000), что может направить весь трафик для IP-сетей 10.80.0.0 и 10.35.0.0 в автономные системы 3000 и 4000 соответственно. Маршрутизатор 2000 «перераспределяет» свое знание этих маршрутов маршрутизаторам 1000, 3000 и 4000.

Теперь, когда маршрутизатор 3000 получит данные для IP-сети 10.34.0.0, он будет знать, что если он перешлет их маршрутизатору 2000, то информация попадет к адресату (АС 1000).

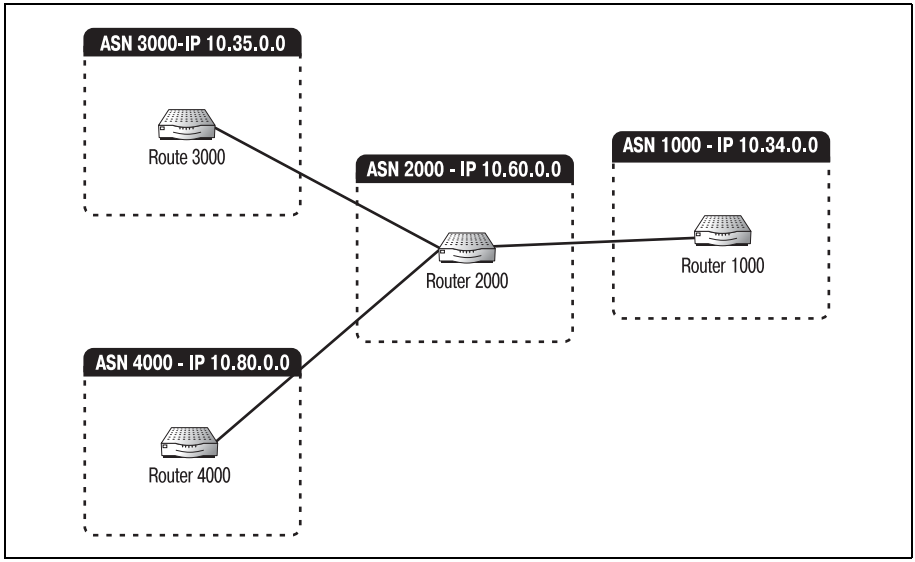

***Рис. 18.8.** Четыре автономные системы, связанные EBGP*

Перераспределение маршрутов помогает BGP-узлам получать сведения друг о друге и о тех сетях, которые они представляют. Объявляемые маршруты могут быть двух видов: динамические и статические.

Динамический BGP-маршрут – это маршрут, о котором узел узнает вместе с другими маршрутизаторами из обновлений IGP. В нашем примере (рис. 18.8) маршруты, о которых узнают маршрутизаторы 3000 и 4000 (из обновлений EIGRP), считаются динамическими.

Примечание

EIGRP – это общепринятый IGP, использующийся для связи между пограничными шлюзами.

Примечание

Процесс перераспределения маршрутов, сведения о которых получены динамически, называется динамическим перераспределением маршрутов BGP (в отличие от статического перераспределения маршрутов BGP, которое требует некоторого участия с вашей стороны).

Давайте настроим маршрутизатор BGP на динамическое перераспределение маршрутов. Сначала следует сконфигурировать IGP, чтобы разрешить перераспределение его маршрутов, а затем настроить BGP так, чтобы он передавал далее эти перераспределенные маршруты:

```
Router(configure)#router eigrp 40
Router(configure-router)#network 10.153.0.0
```

```
Router(configure-router)#redistribute bgp 65100
Router(configure-router)#redistribute connected
```

В режиме конфигурирования BGP выполните следующие команды:

```
Router(configure)#router bgp 65100
Router(configure-router)#neighbor 10.154.0.1 remote-as 65100
Router(configure-router)#distribute-list 1 out
Router(configure-router)#redistribute eigrp 10
```

Примечание

Команда `distribute-list` ссылается на список доступа, хранящийся на маршрутизаторе. Подробно мы поговорим о списках доступа в главе 21; пока же вы должны понимать, что они являются важной частью маршрутизации EBGP.

Для того чтобы пути успешно перераспределялись, они должны быть указаны в локальном списке доступа. Списки доступа содержат сетевые IP-адреса тех узлов, к которым маршрутизатор имеет доступ.

Есть и другой вариант: администратор может запрограммировать маршрут для узла BGP. Такие маршруты называются статическими. Используя предыдущий пример (рис. 18.8), предположим, что маршрутизатор 2000 узнает о сетях, входящих в AC 1000, посредством статически определенного маршрута. Когда маршрутизатор 2000 передает эту информацию маршрутизаторам 3000 и 4000, этот процесс рассматривается как статическое перераспределение путей BGP.

Чтобы настроить маршрутизатор BGP на статическое перераспределение маршрутов, необходимо определить статический путь:

```
Router(configure)#ip route 10.153.40.0 255.255.0.0 null 0
```

Примечание

Обратите внимание на то, что интерфейс установлен в `null 0`. Обычно это привело бы к отбрасыванию всех пакетов для указанной сети. Однако перераспределение маршрутов произойдет раньше, чем пакеты будут отброшены.

После того как статический путь задан, можно конфигурировать статическое перераспределение для BGP. Используйте команду `redistribute static`:

```
Router(configure-router)#neighbor 10.153.60.1 remote-as 700
Router(configure-router)#redistribute static
```

Эти простые команды обеспечивают распространение маршрутов от узла к узлу. Однако возможна такая ситуация, когда вы не захотите, чтобы распространялись все маршруты. Тогда вам необходимо сконфигурировать карту маршрутов.

Карты маршрутов BGP

Карты маршрутов BGP используются для фильтрации перераспределения маршрутов от АС к АС. Представьте себе то количество маршрутов BGP, которыми системы обмениваются в Интернете каждый день. Если каждый BGP-узел будет перераспределять все пути, о которых он знает, то этот трафик приведет к остановке Интернета. Карты маршрутов дают возможность определить, какие из известных вашему маршрутизатору путей следует перераспределять.

Примечание

Карты маршрутов работают только на уровне обновлений. То есть любая имеющаяся карта маршрутов будет воздействовать только на отправляемые пути, но не на присылаемые.

Карта маршрутов может состоять из перечня критериев, выполнение которых будет означать разрешение или запрет на объявление маршрута или изменение некоторой метрики пути. Например, возвращаясь к сетям, изображенным на рис. 18.8, администратор может создать карту маршрутов для маршрутизатора 2000, в которой будет говориться: «Не распространять маршруты, информация о которых получена от автономной системы 2000». (Технический язык карты маршрутов зависит от производителя маршрутизатора.)

Если эта карта маршрутов будет использоваться, то маршрутизатор 1000 будет продолжать получать обновления от маршрутизатора 2000. Но он будет накладывать на них карту маршрутов и определит, что пересылать эти маршруты маршрутизаторам 3000 и 4000 не следует.

Но давайте предположим, что АС 3000 связана и с АС 1000, как показано на рис. 18.9.

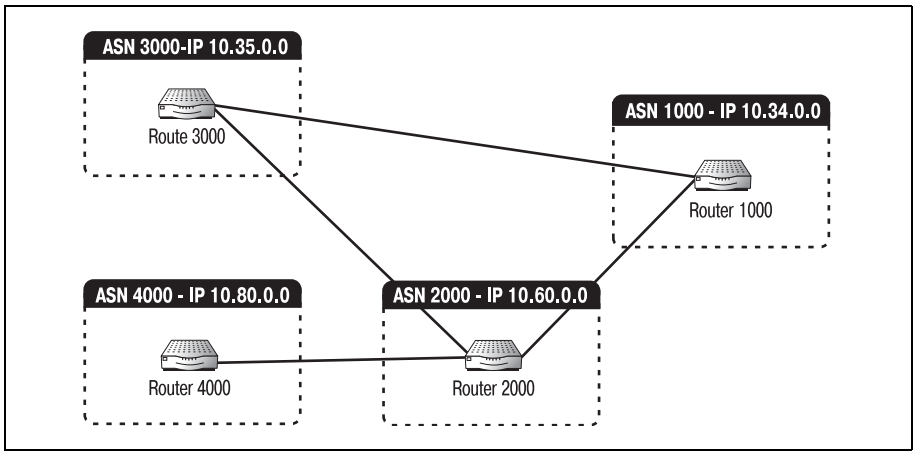

Рис. 18.9. *Четыре связанные автономные системы*

На рис. 18.9 показано, что после того как все обновления BGP обработаны, маршрутизатор 3000 может выбрать для отправки данных в АС 1000 один из двух возможных маршрутов. Он может отправить данные через маршрутизатор 2000 или же послать их прямо на маршрутизатор 1000. Когда возникает необходимость выбора, алгоритм маршрутизации выбирает, какой путь использовать, основываясь на наименьшем значении стоимости или метрики.

Администратор АС 2000, зная, что маршрутизатор 2000 является одним из возможных путей к АС 1000, мог создать карту маршрутов, которая обеспечит преимущество прямого пути (маршрутизатор 3000 – маршрутизатор 1000) над непрямым (маршрутизатор 3000 – маршрутизатор 2000 – маршрутизатор 1000). Предполагая, что обновления маршрутов, приходящие от маршрутизатора 1000, отправляются с метрикой по умолчанию 4, карта маршрутов маршрутизатора 2000 могла бы указывать: «Перераспределять все пути, полученные от маршрутизатора 1000, с метрикой 10». Тогда после того, как маршрутизатор 3000 получил бы все свои BGP-обновления, он бы стоял перед выбором – по какому маршруту отправить данные в АС 1000. Данные можно было бы отправить непосредственно маршрутизатору 1000 со стоимостью 4 или же через маршрутизатор 2000 со стоимостью 10. В девяносто девяти случаях из ста алгоритм маршрутизации остановился бы на прямом пути.

И наоборот, если ваш узел использует более совершенную технологию, и вы хотите, чтобы ваш пограничный шлюз был предпочтительным маршрутом к другой АС, то можете установить более низкую стоимость для пути. Реализуя карту маршрутов, которая изменяет метрики другой автономной системы на меньшее значение, вы обеспечите превосходство пути, проходящего через ваш узел, над другими путями, ведущими в эту АС.

Такая возможность может быть полезной в случае приобретения компании. Если одна компания покупает другую и хочет, чтобы (до тех пор пока не будут сделаны изменения сети) предпочтительный путь к маленькой сети проходил через большую, она может использовать карту маршрутов. Задание карты маршрутов производится за несколько шагов. Сначала необходимо создать список доступа, чтобы определить правила:

```
Router(configure)# ip as-path access-list 1 permit ^65000$
```

Эта команда создает список доступа, который разрешает доступ к любым пакетам, полученным от АС-65000. Затем следует создать карту маршрутов:

```
Router(configure-router)#route-map MYMAP permit 10
Router(configure-router)#match as-path 1
Router(configure-router)#set weight 30
```

Команда `match as-path 1` указывает, что карта маршрутов должна искать соответствия правилам, установленным в списке доступа 1. Команда `set weight` говорит, что карта маршрутов должна установить метрику каждого пакета, который соответствует правилам, в 30.

Наконец, настраиваем BGP на использование карты маршрутов:

```
Router(configure-router)#neighbor 10.153.50.1 route-map MYMAP in
```

Эта строка команд указывает BGP, что нужно применять карту маршрутов MYMAP для любых входящих обновлений, поступающих с адреса 10.153.50.1.

В предыдущих разделах вы познакомились с основами BGP-соединений в EBGP-среде. Но все мы знаем, что мир – это не совершенная среда маршрутизации. Не все начинает работать с первой попытки. Несмотря на наличие лучшей конфигурации и хорошего планирования, вы можете столкнуться с такой проблемой BGP, как пульсация маршрутов.

Пульсация маршрутов BGP и торможение

Если BGP-узлу не удается соединиться с одним из маршрутизаторов, присутствующих в его таблице (в ходе сессии одноранговогo обмена), то такой маршрут называют *пульсирующим*. То есть, если маршрутизатор A узнает о маршрутизаторе B (рис. 18.10) в результате динамического перераспределения маршрутов и не может установить с ним соединение, то говорят о пульсации пути к маршрутизатору B.

Причиной пульсации маршрута может быть все что угодно, начиная с временной недоступности линии T1 и заканчивая «падением» целой сети. В любом случае пульсация маршрута может значительно затруднить работу процессора. Если некий маршрут пульсирует, и каждый BGP-узел, связанный с этим маршрутом, продолжает распространять информацию о нем по Интернету, то огромный объем информации перегрузит маршрутизаторы. Такие информационные заторы могут привести к остановке работы Интернета.

Чтобы справиться с этой проблемой, применяется процедура, называемая *торможением маршрутов*. Алгоритм торможения заключается в помещении маршрутов, пульсирующих к течение некоторого времени *x* (обычно зависящего от частоты обновлений BGP), в «черный спи-

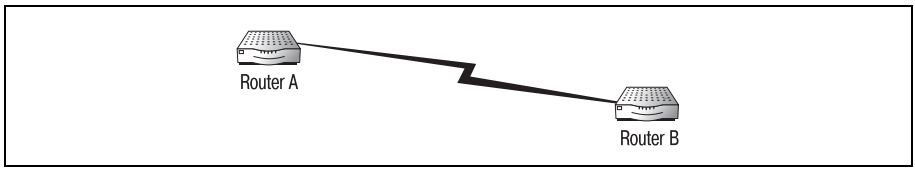

Рис. 18.10. *Пульсирующий маршрут*

сок». Когда маршрут «тормозится», он удаляется из таблиц маршрутов его BGP-соседей. Тогда дальнейшие обновления не будут включать в себя данный маршрут. Это приводит к тому, что маршрут стирается из коллективной памяти Интернета. Маршрут становится невидимым до тех пор, пока не будут решены проблемы связи данного маршрутизатора с соседями.

Примечание

Зачастую должно пройти какое-то определенное время (после «возвращения к жизни»), чтобы временно отключенные маршруты могли быть снова объявлены, что приводит к более длительному неоправданному простою маршрута. Это один из недостатков торможения пульсирующих маршрутов.

Администратору не требуется настраивать торможение пульсирующих путей на маршрутизаторе – это свойство самого протокола. Но осознание процесса необходимо для понимания работы BGP в целом.

Если соединение вдруг пропадает, сначала проконсультируйтесь со своим провайдером. Часто оказывается, что проблема, вызванная неполадками у провайдера (например, отошедший кабельный разъем), приводит к тому, что маршруты, ведущие к вам, «тормозятся» и вы на несколько часов становитесь невидимы.

BGP, как и большинство протоколов маршрутизации, при выборе маршрута полагается на использование метрики. Мы уже говорили, что при помощи карты маршрутов значение метрики можно изменить на большее или меньшее (в зависимости от того, хотите ли вы, чтобы маршрут часто использовался). Теперь давайте посмотрим, как заголовок BGP собирает и хранит эти метрики. BGP – это сложная и трудная для понимания тема, и для того чтобы логически продолжить ее изучение, необходимо изучить заголовки BGP.

Заголовки BGP

BGP-сообщение может иметь четыре вида заголовков. Каждый заголовок содержит информацию, специфичную для типа отправляемого сообщения: open, update, notification или keep-alive. Однако каждому из специфичных заголовков предшествует общий заголовок. Общий заголовок BGP предваряет сообщение и указывает другим системам, что оно предназначено для маршрутизаторов BGP.

Общий заголовок BGP

Общий заголовок BGP, называемый также протокольным, присутствует во всех пакетах, отправляемых маршрутизатором BGP. Общий заголовок обозначает пакет как BGP-пакет и указывает, информация какого типа в нем содержится. Поля общего заголовка BGP изображены на рис. 18.11.

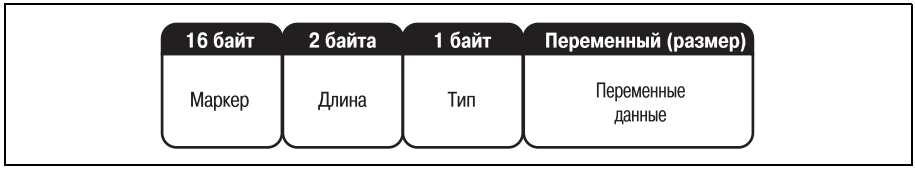

Рис. 18.11. *Общий заголовок BGP*

Общий заголовок BGP состоит из четырех полей:

- Маркер: это 16-байтное поле, идентифицирующее сообщение как BGP-сообщение. Получающий маршрутизатор BGP использует внутренние расчеты, чтобы предсказать, как должно выглядеть поле маркера. Если маршрутизатор получает пакет, в котором поле маркера отличается от рассчитанного, то маршрутизатор знает, что пакет отправлен в нарушение последовательности и должен быть отброшен.
- Длина: это 2-байтное поле, содержащее общую длину (включая заголовок) пакета BGP (а не длину заголовка).
- Тип: 1-байтное поле, указывающее, к какому типу принадлежит сообщение: open, update, notification или keep-alive.
- Переменные данные: в этом поле содержится собственно сообщение. Данным предшествует заголовок, соответствующий определенному типу сообщения. Размер поля зависит от типа сообщения.

Заголовок BGP-сообщения open

Сообщение «open» (открытие) отправляется при установлении соединения с другим маршрутизатором BGP. После того как между маршрутизаторами установлено TCP-соединение, они обмениваются сообщениями «open» для обозначения формального начала сессии. На рис. 18.12 приведены поля заголовка сообщения «open».

Рис. 18.12. *Заголовок сообщения open*

Перечислим поля заголовка сообщения «open»:

- Версия: поле указывает, какую версию BGP использует отправитель. Эта информация помогает маршрутизаторам определить, совместимые ли протоколы они используют.
- Номер АС: в этом 2-байтном поле указан номер автономной системы маршрутизатора, отправившего сообщение.

- Время удерживания: отправляющий маршрутизатор использует это поле для мониторинга активности соединения с получателем. Если отправитель сообщения «open» не получает ответа в течение времени, определенного в этом поле, то соединение с получателем считается закрытым.
- BGP-идентификатор: 4-байтное поле, которое ссылается непосредственно на отправителя сообщения. Обычно это его IP-адрес.

Два оставшихся поля являются необязательными:

- Длина необязательных параметров: это 1-байтное поле содержит длину следующего поля – «необязательные параметры». Если такие параметры не заданы, то поле устанавливается в 0.
- Необязательные параметры: поле содержит любые параметры, которые отправляющий маршрутизатор хочет передать получателю. В настоящее время (в BGP4) в сообщении «open» можно отправить только один необязательный параметр – «Аутентификационные данные», который применяется в тех случаях, когда перед использованием пакета его необходимо аутентифицировать.

Заголовок BGP-сообщения update

Сообщение «update» (обновление) распространяется между маршрутизаторами BGP для изменения информации в таблицах маршрутов. Такие сообщения имеют заголовок, состоящий из пяти полей. Вместе с этим заголовком они относятся к «переменным данным» общего заголовка BGP. Поля заголовка сообщения «update» приведены на рис. 18.13.

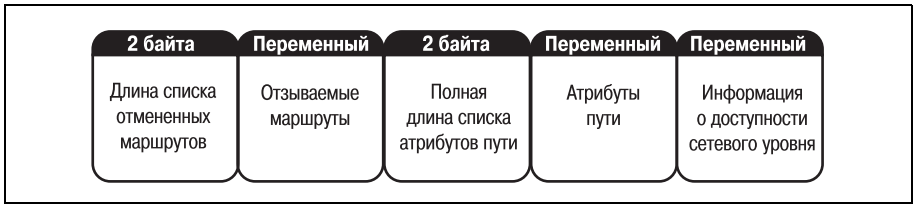

Рис. 18.13. *Заголовок сообщения update*

Два первых поля заголовка сообщения «update» относятся к путям, которые должны быть удалены из таблиц маршрутов получателя. Такие пути называются отзываемыми (withdrawn).

- Длина списка отмененных маршрутов: это 2-байтное значение указывает длину списка отмененных маршрутов, представленного в следующем поле. Если нет отменяемых маршрутов, то поле устанавливается в 0.
- Отзываемые маршруты: это поле переменной длины, содержащее IP-префиксы всех путей, которые следует удалить из таблицы маршрутов.

- Полная длина списка атрибутов пути: это 2-байтное значение длины следующего поля.
- Атрибуты пути: поле содержит метрики, используемые BGP для присвоения значений отдельным путям (поговорим об этом чуть позже в этом же разделе).
- Информация о доступности сетевого уровня: это поле переменной длины содержит IP-префиксы путей, которые следует добавить в таблицу маршрутов.

BGP-сообщения notification и keep-alive

Маршрутизаторы BGP обмениваются сообщениями «notification» (уведомление) при обнаружении ошибки. Когда у маршрутизатора возникает проблема, он отправляет сообщение «notification» и закрывает все открытые сессии. Такое сообщение состоит из трех полей: кода ошибки, субкода ошибки и данных.

Сообщение «keep-alive» (еще жив) отправляется, пока еще не истекло время удерживания. Сообщение «keep-alive» состоит из одного поля, которое сообщает получателю, чтобы он не принимал во внимание истечение времени удерживания.

AS-Path и AS-Set

AS-Path – это атрибут, который прикрепляется к обновлению маршрута BGP. Атрибут AS-Path представляет совокупность автономных систем, через которые прошло обновление, прежде чем попасть к адресату (рис. 18.14).

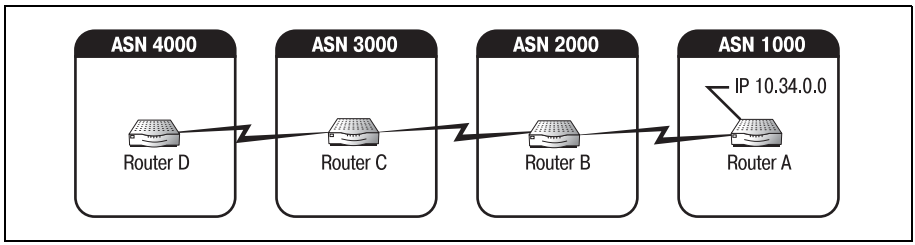

Рис. 18.14. *Цикл AS-Path*

На рис. 18.14 маршрутизатор A объявляет, что сеть 10.34.0.0 принадлежит автономной системе с номером ASN 1000. Когда маршрутизатор B обрабатывает обновление, к обновлению в виде префикса добавляется AS-Path, который сообщает, откуда поступило обновление. Теперь в сообщении говорится: «AC 1000 говорит, что 10.34.0.0 относится к ASN 1000».

Затем маршрутизатор B объявляет тот же маршрут маршрутизатору C. После того как C обработает обновление, оно будет выглядеть следующим образом: «ASN 2000 говорит, что ASN 1000 говорит, что 10.34.0.0

относится к ASN 1000». Наконец, когда обновление достигает маршрутизатора D, AS-Path содержит цепочку «ASN 3000 – ASN 2000 – ASN 1000».

Примечание

Полный набор атрибутов AS-Path называется AS-Set.

Теперь, если маршрутизатор D имеет сообщение для IP-сети 10.34.0.0, ему следует просто посмотреть на AS-Set обновления. AS-Set предоставляет маршрутизатору последовательность атрибутов AS-Path, которая ведет к пункту назначения. В нашем примере маршрутизатор D будет пересылать любые данные для 10.34.0.0 по адресу первого AS-Path (ASN 3000). Маршрутизатор C перенаправит данные на то устройство, которое для него является первым AS-Path – ASN 2000, и так будет продолжаться, пока информация не достигнет ASN 1000.

Next hop

Атрибут «next hop» (следующий переход) работает аналогично AS-Path. Он указывает IP-адрес порта маршрутизатора, который используется для достижения конкретной AC. Возвращаясь к примеру (рис. 18.14), недостаточно просто сказать маршрутизатору D, что AS-Path для данных, направляемых в сеть 10.34.0.0, – это ASN 3000, если он не знает, как добраться до ASN 3000. Необходимо настроить маршрутизатор D так, чтобы он понимал, где находится ASN 3000.

Атрибуту «next hop» для маршрутизатора D необходимо присвоить значение IP-адреса физического порта, через который D может получить доступ к маршрутизатору C (ASN 3000). Задавая атрибут «next hop», вы назначаете постоянный путь для данных, направляемых определенному адресату. Сети BGP могут быть очень сложными, и такие простые средства, как установка атрибута «next hop», помогут навести некоторый порядок.

Origin

Атрибут «origin» может иметь одно из трех значений:

- IGP
- EGP
- Incomplete

Примечание

Не путайте IGP и EGP с IBGP и EBGP. В то время как IBGP и EBGP являются отдельными протоколами, IGP и EGP – это обозначения таких категорий протоколов, как протоколы внутреннего и внешнего шлюзов. Это совсем не обязательно IBGP и EBGP. На самом деле вы будете часто сталкиваться с тем, что для передачи BGP-данных используются и другие протоколы. Для маршрутизации данных IBGP могут также использоваться такие протоколы, как IGRP (Interior Gateway Routing Proto-

col – протокол маршрутизации внутреннего шлюза) и EIGRP (Enhanced Interior Gateway Routing Protocol – усовершенствованный протокол маршрутизации внутреннего шлюза).

Поэтому BGP-атрибут origin применяется для указания (в общем) того типа протокола, которым был переправлен маршрут.

Значение «IGP» указывает, что информация о маршруте получена от внутреннего протокола, такого как IBGP. BGP-узел считает, что все маршруты, значение атрибута «origin» которых равно «IGP», находятся в его собственной АС. Значение «EGP» указывает, что сведения о маршруте получены через внешнее обновление. BGP-узел будет использовать EBGP для достижения любого маршрута, значение атрибута «origin» которых равно «EGP». Наконец, значение «Incomplete» (неизвестно) указывает, что информация о маршруте получена не с помощью протокола внешней или внутренней маршрутизации. В большинстве случаев это означает, что сведения о маршруте получены в результате перераспределения маршрутов.

Local preference

Локальный приоритет (local preference) – это метрика, используемая для определения предпочтения маршрута в случае, если два маршрута ведут в одну точку. Когда маршрутизатор стоит перед выбором одного из двух (кажущихся одинаковыми) маршрутов, ведущих к одному адресату, он сравнивает атрибуты локального приоритета, чтобы определить, какой из путей использовать. Чем выше локальный приоритет маршрута, тем с большей вероятностью он будет использован. Если локальный приоритет не установлен, то используется значение по умолчанию – 100.

Конфигурирование протокола IBGP

Одно из основных отличий логики маршрутизации IBGP от EBGP заключается в использовании адреса обратной связи (loopback address). В предыдущем разделе рассказывалось, что BGP при установлении соединения между двумя узлами использует определенные физические и логические порты. То есть сессии BGP всегда устанавливаются посредством порта 179, открытого на указанном физическом Ethernet-порту (IP-адресе). В EBGP это вызывает одну проблему: если указанный физический порт недоступен, то установить соединение невозможно. В IBGP проблему решает использование loopback-адреса (адреса обратной связи).

Адрес обратной связи – это IP-адрес, представляющий группу физических портов маршрутизатора. Представляя несколько физических портов, адрес обратной связи гарантирует, что соединение можно будет установить вне зависимости от доступности физических портов.

Адреса обратной связи применяются только в IBGP из-за особенностей достижимости маршрутизаторов. Маршрутизаторы, соединенные EBGP, почти всегда связаны только одним физическим портом, в то время как маршрутизаторы, соединенные IBGP, обычно совместно используют всю сетевую среду.

Конфедерации BGP

Как уже говорилось в этой главе, для обеспечения успешной работы все BGP-узлы автономной системы IBGP должны быть BGP-соседями. То есть каждый маршрутизатор BGP (использующий IBGP) некоторой AC должен иметь физическое соединение со всеми остальными IBGP-узлами. Большая полносвязная AC изображена на рис. 18.15.

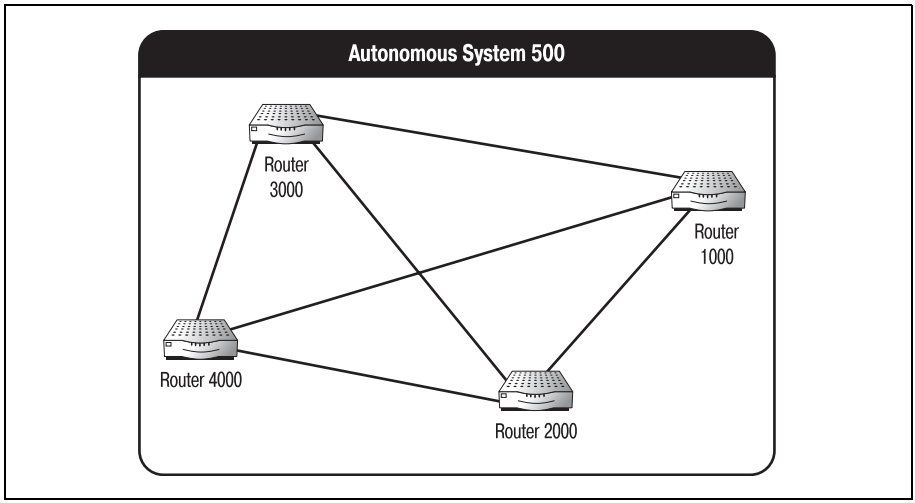

***Рис. 18.15**. Полносвязная автономная система*

Абсолютно очевидно (см. рис. 18.15), что при увеличении количества IBGP-узлов количество соединений между ними стремительно возрастает. Использование, отслеживание и администрирование такого большого числа соединений может быть тяжелым и утомительным даже для самых опытных сетевых специалистов. Необходимо как-то ограничить количество физических соединений маршрутизаторов, не уменьшая при этом количество IBGP-узлов. Одним из таких способов является создание нескольких конфедераций BGP.

Конфедерация BGP – это подгруппа автономной системы. То есть одна большая AC может быть разделена на несколько более мелких AC, но сохраняющих тот же ASN, за счет чего уменьшается количество физических соединений между узлами.

Функционально конфедерации BGP сравнимы с подсетями IP (см. главу 11). Подобно тому как подсети IP уменьшают физический размер

IP-сети, сохраняя все особенности большей сети, конфедерации создают небольшие автономные подобласти, которые сохраняют все внешние характеристики большой исходной АС.

Чтобы разделить АС на несколько конфедераций, необходимо немного подумать. Спланируйте на бумаге, где будут проходить границы ваших конфедераций. Это поможет вам представить, как нужно будет настроить маршрутизаторы внутри конфедераций. После того как вы приняли решение о том, как будет разбита АС на конфедерации, необходимо присвоить новым подгруппам АС идентификаторы конфедераций.

Примечание

Идентификатор конфедерации – это номер, присваиваемый конфедерации для того, чтобы отличать ее от всех других конфедераций той же АС. Идентификаторы конфедерации действуют как номера автономных систем (и придерживаются тех же правил).

Помните, что конфедерация – это маленькая АС. Другими словами, внутри конфедерации все IBGP-соседи должны быть связаны друг с другом. Однако IBGP-соседи одной конфедерации не обязаны быть соединены с каждым из IBGP-соседей другой конфедерации (даже в пределах одной автономной системы). Отсутствие соединений между конфедерациями делает среду более управляемой.

После того как конфедерациям присвоены идентификаторы, можно разорвать физические соединения между узлами разных конфедераций. Теперь у вас есть полностью независимые конфедерации внутри АС. Но как конфедерации будут взаимодействовать друг с другом? Чтобы обеспечить это взаимодействие, тем самым сделав автономную систему полнофункциональной, необходимо определить IBGP-соседей конфедераций.

Соседи конфедераций – это маршрутизаторы, которые соединяют одну конфедерацию с другой. Они работают как BGP-узлы для отдельных конфедераций. Определение соседей конфедераций может быть несколько запутанным из-за одного небольшого нюанса: соседи конфедераций общаются друг с другом по протоколу EBGP.

Так как конфедерации имеют разные идентификаторы, для установления связи между двумя или более конфедерациями требуются EBGP-узлы. Однако так как эти узлы находятся в IBGP-среде, они придерживаются всех правил, установленных для IBGP-узлов. То есть соседи конфедераций совместно используют маршруты как IBGP-соседи, а не как EBGP-узлы (даже если технически они используют EBGP).

После того как соседи конфедераций определены, соединение всех входящих в вашу АС конфедераций одним физическим каналом приведет к возникновению полнофункциональной группы конфедераций, которая для внешнего мира выглядит как одна большая АС. На рис. 18.16 изображена АС, разделенная на несколько конфедераций.

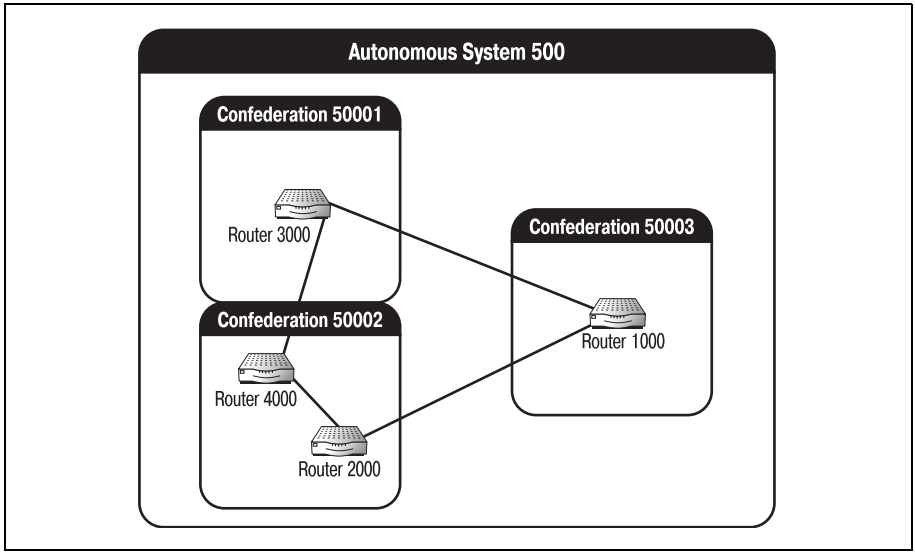

Рис. 18.16. *Группа конфедераций*

Конфигурирование конфедерации

Параметр `confederation` команды `bgp` позволяет создавать конфедерации на маршрутизаторах Cisco, использующих BGP. Чтобы создать конфедерацию из трех маршрутизаторов (10.198.56.1, 10.198.56.2 и 10.198.56.3), которая взаимодействует с одним маршрутизатором вне конфедерации, используйте следующие команды:

```
Router(configure)#router bgp 65000
Router(configure-router)# bgp confederation identifier 500
Router(configure-router)# bgp confederation peers 65100
Router(configure-router)# neighbor 10.198.56.2 remote-as 65000
Router(configure-router)# neighbor 10.198.56.3 remote-as 65000
Router(configure-router)# neighbor 10.198.56.7 remote-as 65100
```

Настраивается BGP-маршуртизатор, находящийся в автономной системе 65000. Этот маршрутизатор добавляется в конфедерацию (ID 500) вместе с двумя другими маршрутизаторами. После того как маршрутизаторы добавлены в конфедерацию, команда `neighbor` используется еще один раз для установления связи с BGP-маршрутизатором 10.198.56.7, находящимся вне конфедерации.

Синхронизация BGP

Может случиться так, что в вашей BGP-среде маршрутизаторы используют несколько протоколов. То есть все маршрутизаторы, которые не используют BGP, должны использовать какой-то другой прото-

кол маршрутизации для обеспечения доставки пакетов. Помните, что BGP – это протокол маршрутизации граничного шлюза, а для маршрутизации данных в оставшейся части вашей сети необходим протокол внутреннего шлюза (IGP).

На ваших маршрутизаторах может работать OSPF, IS-IS или любой другой IGP-протокол (в тех частях сети, которые не обслуживаются BGP). Протокол внутреннего шлюза, применяемый такими маршрутизаторами, взаимодействует с локальными BGP-узлами и предоставляет им табличную информацию. BGP-узлы используют эту информацию для взаимного обновления сведений о текущем состоянии сети.

Но одновременная работа двух или более протоколов может привести к серьезным проблемам. Каждый протокол использует свои собственные обновления маршрутов. Проблема в том, что если один маршрутизатор использует и IGRP, и BGP, и получает обновления таблицы маршрутов из разных источников, то обновлениям какого протокола отдавать предпочтение?

Синхронизация BGP помогает маршрутизатору определить, какие обновления следует включать в его таблицу маршрутов (и затем передавать другим маршрутизаторам BGP). Благодаря BGP-синхронизации маршрутизатор BGP может удерживать свои BGP-обновления до тех пор, пока все маршрутизаторы не сообщат, что получили обновления от IGP. Например, на рис. 18.17 показана сеть, состоящая из двух автономных систем. В каждой АС работают BGP и OSPF. Обратите внимание на то, что некоторые маршрутизаторы используют два протокола, в то время как на других применяется только OSPF.

Прежде чем BGP сможет начать свои обновления маршрутов, он должен подождать, пока все маршрутизаторы получат обновление OSPF (рис. 18.17). Это гарантирует получение каждым маршрутизатором наиболее точной информации.

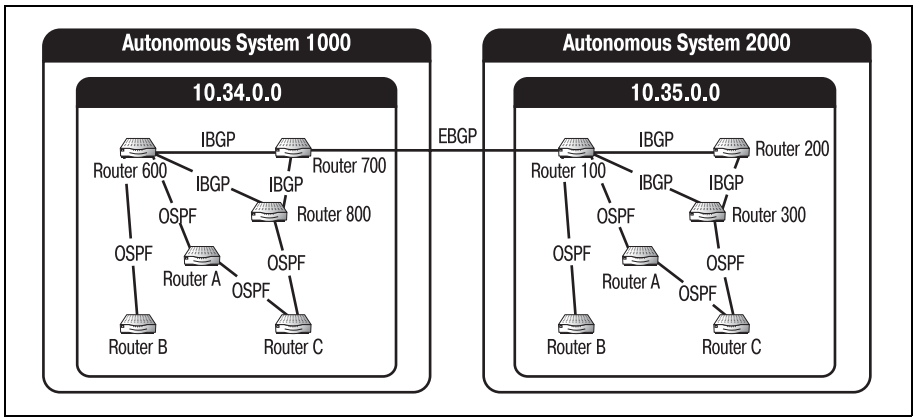

Рис. 18.17. *АС, в которых работают два протокола маршрутизации*

Чтобы отменить синхронизацию BGP, используйте команду no:

```
Router(configure-router)#no synchronization
```

С синхронизацией BGP тесно связана такая тема, как отражение маршрутов BGP. Ранее в этой главе уже были кратко рассмотрены карты маршрутов и фильтры. Теперь затронем более сложный вопрос – отражение маршрутов.

Отражение маршрутов BGP

Как уже не раз говорилось, все IBGP-соседи в пределах автономной системы должны быть полностью взаимосвязаны. Одной из причин этого является то, что IBGP-сосед не может перераспределять информацию о маршрутах, полученную от одного IBGP-соседа, другому. IBGP-соседи передают только обновления, непосредственно связанные с их собственными путями. IBGP-маршрутизаторы передают только обновления, полученные из первых рук, благодаря чему снижается количество некорректных или устаревших маршрутов. Но очевидна и проблема – все IBGP-соседи должны быть физически соединены друг с другом. Решением проблемы являются отражатели маршрутов BGP.

IBGP-соседи, сконфигурированные как отражатели BGP-маршрутов, могут распределять (или отражать) пути, информация о которых получена от одного BGP-соседа, другому BGP-соседу. Иными словами, BGP-маршрутизатор может отправить обновление другому BGP-маршрутизатору, с которым он физически не связан. Отражение маршрутов показано на рис. 18.18.

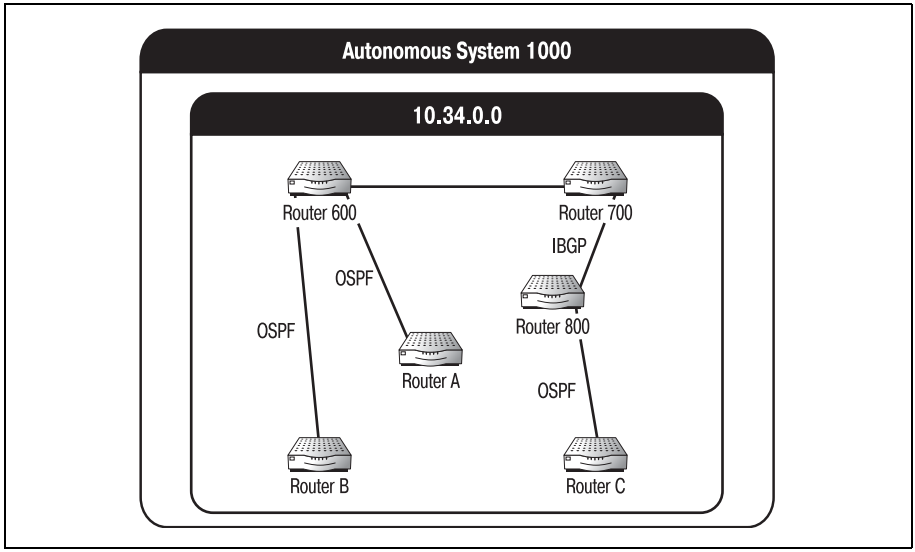

Рис. 18.18. *AC с отражателем маршрутов*

Маршрутизатор 700 может отражать маршруты, полученные от маршрутизатора 600, маршрутизатору 800. Отражатели маршрутов – это очень полезное для администраторов средство, которое помогает уменьшить количество необходимых физических соединений между IBGP-соседями. Однако если маршрутизаторы используются в качестве отражателей маршрутов, они испытывают существенно большие нагрузки, чем обычные BGP-маршрутизаторы.

Упражнения

1. Сконфигурируйте маршрутизатор 10.198.24.1 для работы с BGP в АС 200.

Решение

```
Router#configure terminal
Router(configure)#router bgp 200
Router(configure-router)#^Z
```

2. Определите статическое перераспределение маршрутов для маршрутизатора 10.16.4.1 (входящего в АС 700).

Решение

```
Router#configure terminal
Router(configure)# neighbor 10.16.4.1 remote-as 700
Router(configure-router)#redistribute static
```

3. Создайте конфедерацию с идентификатором 670, включающую в себя маршрутизаторы 10.156.4.1 и 10.156.4.2 автономной системы 5000.

Решение

```
Router(configure)#router bgp 5000
Router(configure-router)#bgp confederation identifier 670
Router(configure-router)#neighbor 10.156.4.1 remote-as 5000
Router(configure-router)#neighbor 10.156.4.2 remote-as 5000
```

19

Изучение IS–IS

Протокол IS-IS (Intermediate System to Intermediate System – связь между промежуточными системами[1]) относится к протоколам маршрутизации состояния канала. IS-IS был разработан ISO (на основе DECnet Phase V) для работы с CLNP (Connectionless Network Protocol – сетевой протокол без установления соединения). Но благодаря своей гибкости протокол может работать с архитектурами OSI и DNA, являясь своего рода протокольным мостом между принадлежащей DECnet моделью DNA и широко распространенной моделью OSI.

Эта глава посвящена конфигурированию маршрутизаторов Cisco для работы с IS-IS. В отличие от других протоколов, рассматриваемых в этой книге, IS-IS изначально создавался для такой запатентованной топологии, как DECnet. Поэтому он не обязан был соответствовать тем же стандартам OSI, которым соответствуют такие протоколы, как IGRP и BGP. В результате некоторые структуры и понятия IS-IS могут казаться «инородными».

В главе будут представлены следующие темы:

- IS-IS и DECnet
- Связь IS-IS с CLNP
- IS-IS как протокол состояния канала

---

[1] Промежуточная система (IS, intermediate system) – это, по определению, маршрутизирующая система, получающая данные от конечной системы-отправителя или другой промежуточной системы и передающая их конечной системе-получателю или следующей промежуточной системе. То есть в нашей терминологии это маршрутизатор (в DECnet Phase IV использовалось это название). – *Примеч. перев.*

- Метрики и алгоритмы IS-IS
- Адресация IS-IS, области и домены
- Пакеты IS-IS
- Маршрутизация IS-IS
- Конфигурирование IS-IS

IS–IS и DECnet

В 1975 году Digital Equipment Corporation разработала стек протоколов DECnet для своей популярной линии компьютеров VAX. Каждая редакция (версия) DECnet называлась фазой: Phase I, Phase II и т. д. Как и всем протоколам, для перемещения информации из одной системы в другую DECnet необходимо было придерживаться специальной архитектуры. Была создана модель протокола, которая поясняла, как данные должны перемещаться от пользовательского интерфейса ПК одной системы через среду передачи к системе назначения. Сегодня большая часть протоколов маршрутизации придерживается архитектуры модели OSI. Первые четыре фазы DECnet для перемещения данных использовали запатентованную архитектуру.

Примечание

IS-IS был основан на DECnet Phase V; однако большинство основных механизмов DECnet Phase V было реализовано уже в DECnet Phase IV.

DECnet был (и остается) патентованной системой, используемой для обеспечения связи между устройствами DEC. Поэтому первые четыре версии DECnet не следуют модели OSI – они базируются на модели DNA (Digital Network Architecture – сетевая архитектура Digital). Позже компания Digital разработает соответствующую OSI-версию DECnet, которая получит название DECnet Phase V или DECnet OSI/DNA. DECnet Phase IV DNA включает в себя восемь уровней, которые более или менее сопоставимы семи уровням модели OSI. На рис. 19.1 представлены восемь уровней Phase IV DNA и показано, как они связаны с моделью OSI.

Примечание

Все фазы DECnet обратно совместимы. Например, протокол DECnet Phase III обратно совместим с Phase II. Когда специалисты DEC приступали к созданию DECnet Phase V, поддерживающей OSI, они осознавали, что реализуемый протокол должен быть обратно совместим с архитектурой DECnet DNA. Это стало важной чертой IS-IS.

Если вы знакомы с моделью OSI, то заметите некоторые небольшие отличия между DNA компании Digital и OSI, созданной ISO. Основное различие заключено в верхних уровнях двух протоколов. Многие верхние уровни DNA захватывают по несколько уровней OSI. Однако

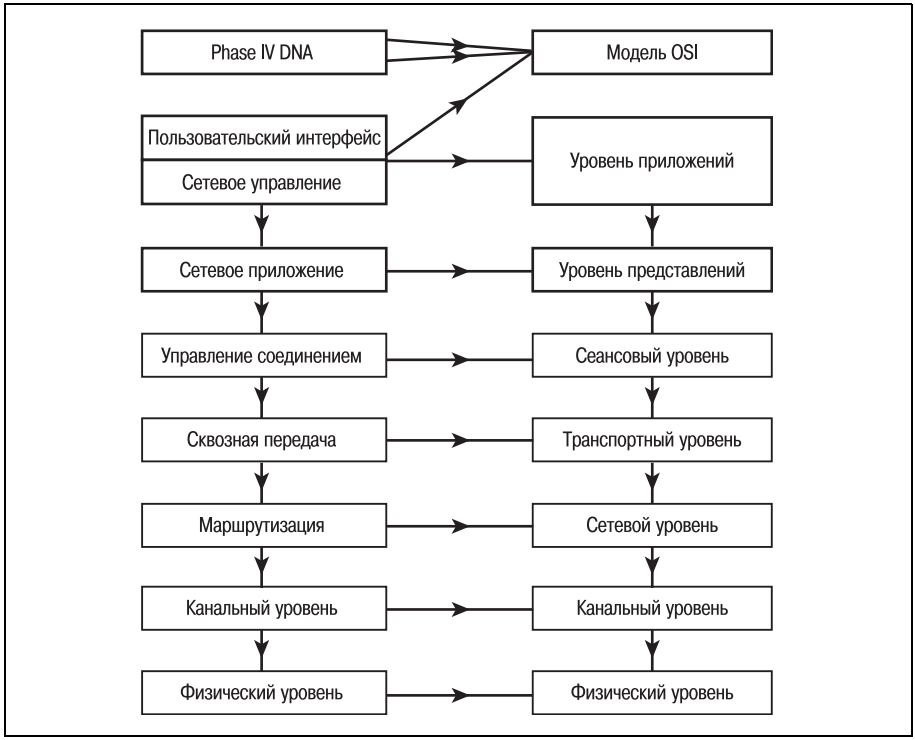

Рис. 19.1. *Восемь уровней архитектуры DECnet Phase IV DNA*

нижние уровни (наиболее важные для маршрутизации) практически одинаковы (это облегчает для Digital переход с DNA на OSI).

Из-за различий верхних и нижних слоев моделей DNA и OSI адреса DECnet устроены не так, как адреса других протоколов. Адреса DECnet состоят из двух частей: области и узла (это понятие будет использоваться и в IS-IS).

Области и узлы DECnet

Адресация DECnet несколько отличается от других протоколов. Адреса DECnet имеют длину 16 бит. Первые 6 бит DECnet-адреса определяют *область*. Адрес области входит в диапазон от 1 до 63 (64 адреса, при этом 0 недействителен). Последние 10 бит адреса – это *узел*. Адрес узла может иметь значения от 1 до 1023 (1024 адреса, 0 недействителен). Итого в сети DECnet может быть 64 449 узлов. Пример адреса DECnet приведен на рис. 19.2.

Например, если одна сеть DECnet состоит из трех областей (5, 6 и 7), в каждой из которых по четыре узла (1, 2, 3 и 4), то адресация организована так, как это показано на рис. 19.3.

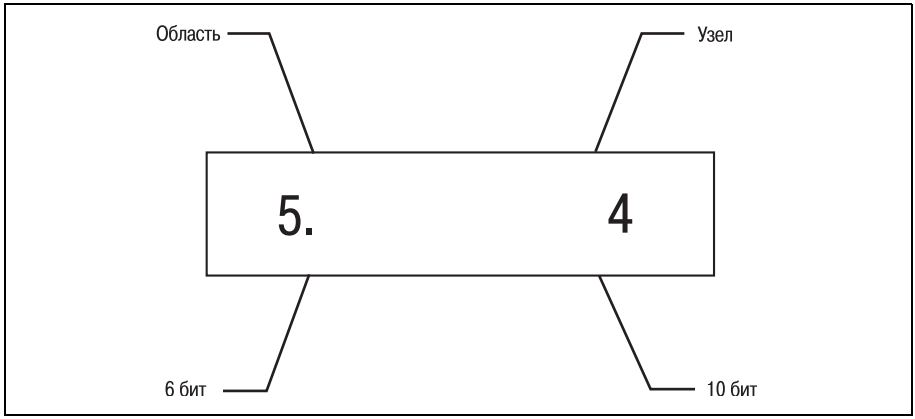

Рис. 19.2. *Разделение адреса DECnet*

Еще одной особенностью адресации DECnet, которую требовалось перенести в IS-IS, является обработка MAC-адресов. MAC-адреса присваиваются системам сети DECnet динамически, на основе их адреса вида область/узел. Например, MAC-адрес конечной системы (поговорим о конечных системах позже) с адресом 4.2 (область/узел) – это AA-00-04-00-02-10. Чтобы получить такой адрес, DEC выполняет следующие действия:

1. Сначала DECnet-адрес преобразуется в двоичный формат:

   4.2 = 000100.0000000010 (помните, что адрес разбит на две части: 6 бит и 10 бит)

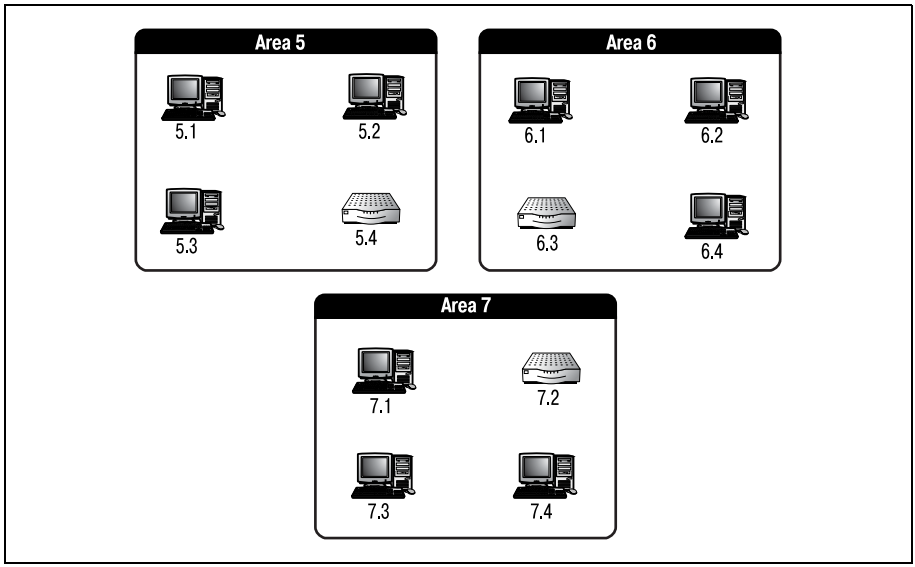

Рис. 19.3. *Сеть DECnet, снабженная адресами*

2. MAC-адреса, по существу, являются последовательностью 8-битных сегментов, поэтому необходимо разбить наш двоичный адрес область/узел на 8-битные части:

   00010000 00000010

3. Затем эти части меняются местами:

   00000010 00010000

4. Переставленные местами двоичные части преобразуются в шестнадцатеричный формат:

   02 10

5. Новые шестнадцатеричные пары добавляются в конец Ethernet-MAC-кода производителя DEC AA-00-04. (Нулевая пара 00 помещается между кодом производителя и адресом область/узел в качестве заполнителя.)

Результатом является MAC-адрес AA-00-04-00-02-10.

Примечание

MAC-адрес («прошитый» адрес) – это обычно статический номер, присваиваемый производителем Ethernet для обеспечения уникальности в больших сетях. Существует возможность, но она редко востребована, изменить этот номер вручную для удовлетворения потребностей определенных сетей.

Такие динамические MAC-адреса используются DECnet (и IS-IS) и для маршрутизации информации, и для отсылки пакетов обновлений другим узлам. Тем, кто знаком с адресацией и маршрутизацией IP, адресная схема DECnet может сначала показаться немного необычной. Эта необычная архитектура является ключом к пониманию того, как работает IS-IS.

Узлы DECnet

Как и у остальных изученных нами протоколов, у DECnet есть своя терминология, и одно из ее наиболее важных понятий – это *узел*. Узел DECnet может относиться к одному из трех типов:

- Конечная система (ES, end system)
- Маршрутизатор первого уровня (L1)
- Маршрутизатор второго уровня (L2)

Конечная система – это все, что не является маршрутизатором. Наиболее распространенным конечным узлом является ПК пользователя, но к ES относятся также принтеры, сканеры и другие сетевые устройства. Конечный узел имеет только один адресуемый интерфейс и может взаимодействовать только с маршрутизаторами первого уровня. В обычной сетевой среде конечных узлов должно быть больше, чем любых других.

ПК не может отправить данные другому ПК без чьей-либо помощи. Поток данных может проходить через концентратор или маршрутизатор, но в любом случае необходимо какое-то устройство, которое обеспечит взаимодействие более чем двух ПК. (Исключением является случай, когда вам требуется соединение только между двумя ПК: их можно соединить перекрестным кабелем.) ПК не поддерживают функции маршрутизации, так что это конечные системы.

Конечная система может рассматриваться как пункт назначения в сети. Большая часть данных, пересылаемых по сети, или отправляется конечной системой, или же ей адресована. Даже несмотря на то, что они не участвуют в маршрутизации, конечные узлы – это важная составляющая процесса маршрутизации в сетях DECnet.

Вторая категория узлов сети DECnet – это L1 (level 1 router – маршрутизаторы первого уровня). Маршрутизатор первого уровня работает внутри области, то есть маршрутизирует данные только в пределах своей области. В терминах маршрутизации это означает, что L1 не имеет сведений о топологии сети за пределами своей области. Это свойство было перенесено и в протокол IS-IS. Не занимаясь маршрутизацией трафика вне собственной области, L1 значительно снижает объем сетевого трафика в отдельной среде.

Широковещательные сообщения могут тормозить работу сети. Указывая, что каждый L1 должен обслуживать только свою область, DECnet избавляется от избытка широковещательных пакетов. Когда маршрутизатор обслуживает несколько сетей, в рассмотрение принимается множество факторов. Если он имеет несколько интерфейсов для нескольких сетей, ему необходимо отсылать множество широковещательных сообщений. Например, если маршрутизатор имеет четыре сетевых интерфейса (A, B, C и D) и получает сообщение для сети C, то обычно он посылает широковещательное сообщение всем интерфейсам. В сообщении спрашивается, с какой сетью связан интерфейс. Получив ответ от сети C, маршрутизатор отправляет ей сообщение. Такой процесс может привести к избытку широковещательных сообщений.

На рис. 19.3 изображены маршрутизаторы, относящиеся к первому уровню (адреса область/узел: 5.4, 6.3 и 7.2). Будучи маршрутизатором первого уровня, маршрутизатор 5.4 может перемещать данные только по области 5.

Еще одной важной особенностью маршрутизаторов L1 является способ их адресации в DECnet. Маршрутизаторы первого уровня могут иметь несколько интерфейсов, но всего один адрес типа область/узел. Это значит, что все интерфейсы (порты) маршрутизатора L1 будут иметь один и тот же адрес. Например, если L1 5.4 имеет четыре порта, соединенных с различными конечными системами, то адрес каждого порта будет 5.4. Поясним это на примере области 5. На рис. 19.4 показаны соединения внутри области 5.

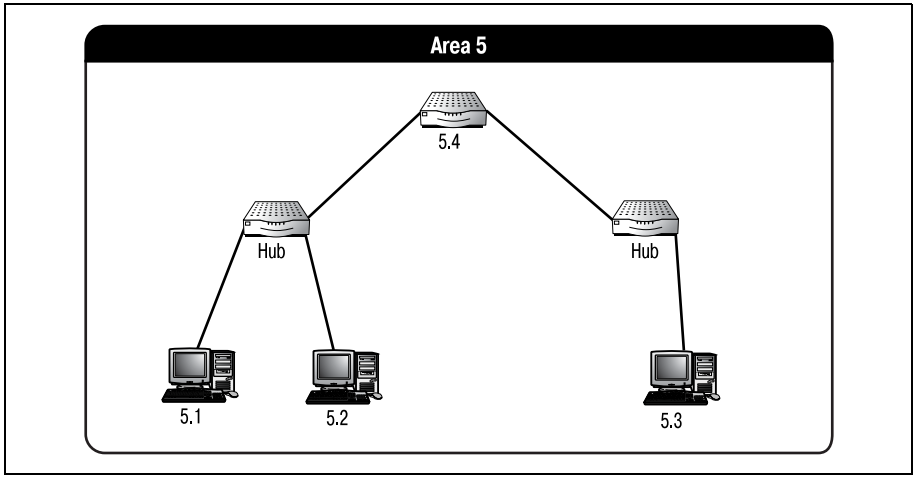

Рис. 19.4. *Маршрутизатор 5.4 с несколькими соединениями и одним адресом*

Обратите внимание на то, что концентраторы (hubs) области 5 подключены к разным интерфейсам маршрутизатора 5.4. Но при этом каждый интерфейс имеет адрес 5.4. Маршрутизатор 5.4 может общаться только с конечными системами 5.1, 5.2 и 5.3. Если конечная система 5.2 захочет отправить данные конечной системе 7.4, ей потребуются услуги маршрутизаторов второго уровня.

Маршрутизаторы второго уровня – это маршрутизаторы, которые могут взаимодействовать с другими маршрутизаторами первого и второго уровней, находящимися в других областях. Чтобы передавать информацию из одной области в другую и обратно, вам понадобится как минимум два маршрутизатора L2. На рис. 19.5 показаны разные пути передачи информации из системы в систему через две области.

На рис. 19.5 видно, что существует три возможных пути передачи информации из одной конечной системы в другую, принадлежащую другой области. Первая возможность – отправка данных маршрутизатору первого уровня той же области. Он перешлет пакеты маршрутизатору второго уровня области назначения. Маршрутизатор второго уровня затем передаст пакеты адресату.

Второй вариант – это вариант 1 наоборот. Данные покидают конечную систему и отправляются на L2 той же области. L2 пересылает данные маршрутизатору первого уровня области назначения, который и отправляет их адресату.

Последний вариант – передача данных от конечной системы на L2 той же области. L2 перешлет информацию другому L2, находящемуся в области адресата. Этот второй L2 передаст данные конечной системе.

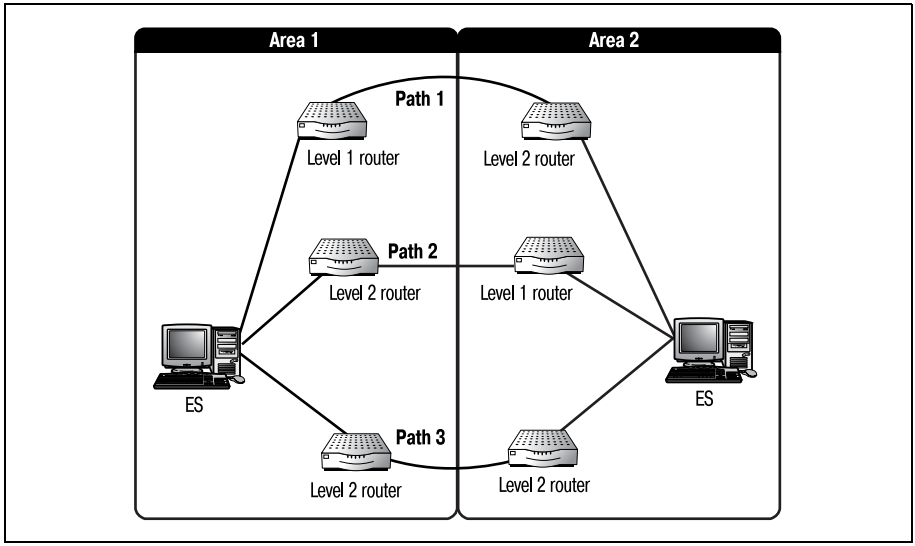

Рис. 19.5. *Поток данных через области*

Как видите, маршрутизаторы второго уровня также могут взаимодействовать с конечными системами (выполняя функции маршрутизаторов первого уровня). Как и маршрутизаторы первого уровня, L2 имеют только один адрес, разделяемый всеми интерфейсами. На рис. 19.6 изображена полностью адресованная маршрутизируемая сеть DECnet, включающая в себя три области.

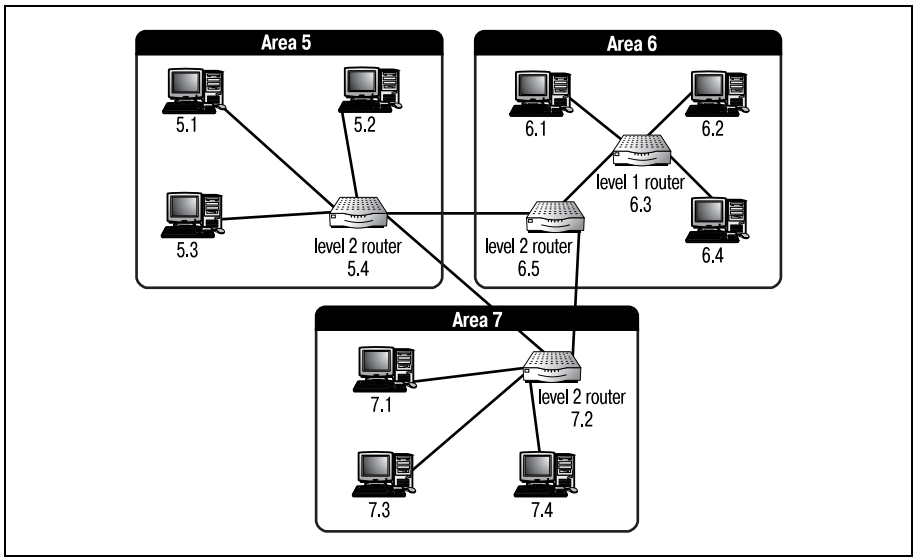

Рис. 19.6. *Маршрутизируемая сеть DECnet*

Основы маршрутизации DECnet

Предшественником IS-IS был протокол DRP (DECnet Routing Protocol протокол маршрутизации DECnet), относящийся к DECnet Phase IV. DRP маршрутизировал пакеты от системы к системе, используя систематический подход. Основой DRP является сообщение hello (приветствие). Сообщения hello – это пакеты Ethernet, которые регулярно отсылаются из системы в систему для обновления данных о состоянии остальных систем.

Отправляя сообщения hello, маршрутизаторы могут получить информацию о состоянии окружающих их каналов и маршрутизаторов. Конечная система отсылает сообщение hello каждому L1 или L2, с которым она взаимодействует. Схема рассылки сообщений hello внутри области изображена на рис.19.7.

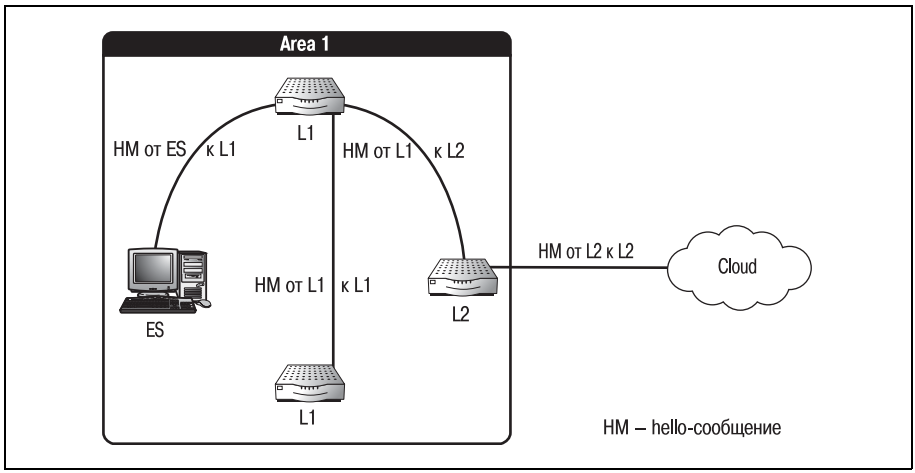

Рис. 19.7. *Жизненный цикл сообщения hello*

Конечная система отправляет сообщение hello на L1. Это сообщение говорит L1 о том, что конечная система функционирует в сети и готова принимать пакеты. L1 использует эту информацию для создания картины своей домашней области. Собирая сообщения hello от конечных систем и других L1, любой отдельный L1 может получить актуальное представление о топологии сети.

Хотя L1 и не отправляет сообщения hello обратно в конечную систему напрямую, конечные системы прослушивают сообщения L1–L1 и создают собственную картину текущей топологии сети. Маршрутизаторы первого уровня отправляют сообщения hello другим L1 и L2. Эти сообщения служат той же цели, что и отправляемые конечными системами. L1 извещает все остальные L1 и L2 той же области о своем текущем состоянии.

Маршрутизаторы второго уровня отправляют сообщения hello только другим маршрутизаторам второго уровня (помните, что L1 и ES имеют информацию только об их собственной области, а L2 имеют информацию о маршрутах к другим областям, поэтому обмен сообщениями hello невозможен).

При отправке сообщений в сеть DECnet конечные системы, так же как и L1 и L2, должны выполнить несколько шагов:

1. Большинство протоколов (и DECnet в том числе) позволяет хранить кэш маршрутов. Конечная система всегда проверяет свой кэш маршрутов, прежде чем отправить пакет в сеть. Кэш маршрутов содержит таблицу, связывающую адресатов с маршрутизаторами, которые могут доставить пакет по данному адресу.
2. Если конечная система не находит нужный маршрутизатор в кэше, она добавляет в пакет MAC-адрес пункта назначения (см. формулу в разделе «Области и узлы DECnet») и пересылает его ближайшему L1 (или L2).

Примечание

Кэш маршрутов – это часть памяти ПК, предназначенная для хранения адресов устройств, на которые часто отсылаются пакеты. Другими словами, всякий раз, когда ПК успешно отправляет пакет, адрес назначения и адрес маршрутизатора, через который был отправлен пакет, записываются в кэш маршрутов ПК. В следующий раз, когда нужно будет отослать пакет тому же устройству, ПК просто посмотрит в свой кэш и будет знать, куда пересылать данные.

3. Получивший пакет маршрутизатор изучает информацию об адресате и, используя свою таблицу маршрутов (созданную на основе различных пакетов hello), пересылает пакет конечному адресату. Если адресат находится в другой области, то L1 отправляет пакет L2, который может взаимодействовать с другой областью. В некоторых случаях может потребоваться участие нескольких L2.

Если нужно отправить пакет из конечной системы через несколько областей, такой пакет сначала попадет на L2 домашней области. Этот L2 найдет в своей таблице маршрутов L2, связанный с областью назначения. (Так как маршрутизаторы второго уровня разделяют информацию о разных областях, L2 домашней области должен знать местоположение L2 области назначения вне зависимости от количества разделяющих их областей.) Затем данные пересылаются собственно адресату.

DECnet Phase V

Когда начиналась разработка DECnet Phase V, компания DEC приняла решение принять в качестве основы для протокола модель OSI. Помните, что так как все фазы DECnet обратно совместимы, протокол DECnet Phase V должен был соответствовать двум стандартам: OSI и DNA.

Две архитектуры были схожи, поэтому DEC смогла успешно перенести DECnet в модель OSI. Но был необходим протокол маршрутизации, который мог бы работать и в архитектуре OSI, и в DNA, и маршрутизировать протокол DECnet (как и IP). Решением стало создание IS-IS. Этот протокол маршрутизации, соответствующий стандарту OSI, стал предпочтительным в сетях DECnet Phase V.

Связь IS–IS с CLNP

Еще до создания DECnet Phase V организация ISO разработала Connectionless Network Protocol (сетевой протокол без установления соединения). CLNP – это маршрутизируемый протокол OSI, обеспечивающий передачу данных между двумя конечными системами без установления соединения. CLNP – быстрый протокол достаточно небольшого размера (с небольшой нагрузкой на сеть), являющийся OSI-эквивалентом IP.

Примечание

Протокол без установления соединения, в отличие от протокола с установлением соединения, не требует подтверждения приема пакета. Поэтому такие протоколы быстрее, чем протоколы с установлением соединения (но зато они считаются менее надежными). IP – протокол без установления соединения.

Когда компания DEC начинала работу над DECnet Phase V OSI/DNA, ей был необходим полностью соответствующий модели OSI маршрутизируемый протокол, который бы работал на третьем уровне OSI. Их внимание привлек CLNP. Но существовало одно «но» – протокол должен был работать в обратно совместимой DECnet-среде, основанной как на модели OSI, так и на DNA. В то время не существовало маршрутизируемого протокола, который был бы настолько гибок, чтобы работать в большой среде без установления соединений, справляясь и с OSI-, и с DNA-архитектурами.

Примечание

Третий уровень моделей OSI и DNA используется для маршрутизации сетевых данных. В модели OSI это сетевой уровень, а в DNA – уровень маршрутизации. Такие сходства и сделали переход с DNA на OSI достаточно простым для DECnet.

Здесь две истории возникновения IS-IS сливаются воедино. DEC был необходим гибкий протокол маршрутизации для новой, соответствующей модели OSI, версии DECnet Phase V. ISO уже разработала маршрутизируемый протокол для работы в средах без установления соединений. Компания DEC помогла ISO создать IS-IS в OSI-среде DECnet (CLNP).

Примечание

Помните, что протокол маршрутизации – это протокол, используемый маршрутизаторами и другими сетевыми устройствами для переноса маршрутизируемых прото-

колов по сетевым соединениям. Маршрутизируемые протоколы (такие как IP и IPX) используются для инкапсуляции данных с целью их передачи другим устройствам.

IS-IS показал себя как идеальный протокол для маршрутизации в сетях OSI и DNA. ISO разрабатывала IS-IS как быстрый переносимый протокол маршрутизации состояния канала для CLNP. Как протокол маршрутизации IS-IS заполнил собой пробел в DECnet Phase V. Так как CLNP базируется на модели OSI и по определению является протоколом без установления соединения, то после минимальных переделок IS-IS сможет маршрутизировать и IP (помните, что IP также относится к протоколам без установления соединения).

Велись разговоры (и время от времени они возобновлялись) о том, чтобы сделать CLNP протоколом Интернета по умолчанию. Но через несколько лет стало понятно, что Интернет – это цитадель IP. Поэтому для полноты в IS-IS была введена поддержка IP.

Сам по себе протокол IS-IS не поддерживает IP. Когда в ISO начались работы по изменению IS-IS для поддержки IP, многие уже использовали его для маршрутизации OSI-протоколов. Поэтому, для того чтобы различать две реализации IS-IS, новую версию официально назвали Integrated IS-IS (интегрированный IS-IS). Integrated IS-IS предоставляет одновременную поддержку нескольких маршрутизируемых протоколов, таких как CLNP и IP.

IS–IS – протокол состояния канала

Протоколы маршрутизации делятся на категории в зависимости от того, какие алгоритмы они используют для маршрутизации данных. Одной из таких общепринятых категорий являются протоколы состояния канала. Почти каждый маршрутизатор, присутствующий сегодня на рынке, может работать с одним из множества протоколов состояния канала. И IS-IS, и Integrated IS-IS относятся к протоколам состояния канала.

Маршрутизаторы, использующие протоколы состояния канала, периодически отправляют друг другу обновления. Эти обновления, называемые LSA (link state advertisements – объявления состояния канала), сообщают каждому маршрутизатору статус (состояние) каждого маршрутизатора (канала) среды. У таких обновлений есть свои достоинства и недостатки, но, прежде чем говорить о них, давайте определим, что отличает IS-IS и другие протоколы состояния канала от остальных протоколов маршрутизации.

IS-IS, как и все протоколы состояния канала, проделывает такую процедуру, как затопление, в ходе которой маршрутизатор рассылает по сети большое количество объявлений LSA. Соседние маршрутизаторы получают эти объявления, обновляют свои таблицы маршрутов и от-

правляют обновления своим соседям. Пример объявления LSA приведен на рис. 19.8.

Затопление необходимо маршрутизаторам IS-IS для того, чтобы составить точную картину своего окружения. (Маршрутизатор использует информацию, содержащуюся в LSA, для создания динамической таблицы. Затем маршрутизатор выполняет над таблицей вычисления по своему алгоритму маршрутизации и определяет кратчайший путь для каждого пакета.) Продолжительность LSA-затопления зависит от количества обновлений (изменений) в среде маршрутизации, а также от времени конвергенции.

Примечание

Конвергенция в маршрутизации – это то количество времени, которое необходимо маршрутизаторам для того, чтобы согласовать определенный набор изменений. Другими словами, это время, необходимое всем маршрутизаторам для синхронизации (в результате которой все работающие маршрутизаторы имеют одно и то же представление о сети).

Однако в некоторых сетях затопление может привести к проблемам с трафиком. Наиболее мощные сетевые среды могут и не почувствовать влияния затопления, но у значительной части сетей оно вызовет задержки. Наиболее сильное воздействие затопление оказывает на большие сети, уже имеющие проблемы с пропускной способностью.

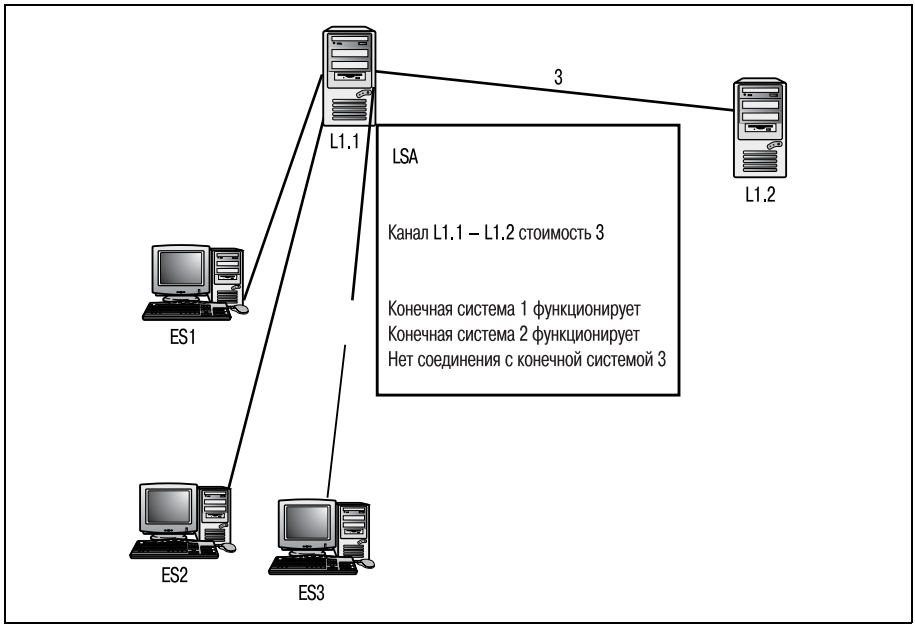

Рис. 19.8. *Пример объявления состояния канала (LSA)*

Большие сети с достаточной пропускной способностью и маленькие сети с небольшим количеством маршрутизаторов могут не сильно страдать от затопления LSA. У первых хватит возможностей для обработки и LSA, и обычного повседневного трафика, а у вторых не так много маршрутизаторов, чтобы организовать массивную рассылку объявлений.

В сети любого размера (где предполагается использование IS-IS) наличие достаточной пропускной способности является определяющим для того, чтобы справиться с затоплением LSA без заметных задержек. Если вы обдумываете возможность применения IS-IS в существующей сети, убедитесь, что пропускной способности достаточно для того, чтобы иметь дело с LSA. Это избавит вас от головной боли в дальнейшем.

Фактически затопление воздействует на сеть двумя способами. Первый, и наиболее очевидный, заключается в том, что сеть заполоняется LSA-пакетами (LSA packets, или LSP). Обновления LSA – это большой поток информации, затрудняющий перемещение «обычных» сетевых данных. Помните, что каждый маршрутизатор должен сообщить своим соседям (из той же области), какие изменения сделаны в его таблице маршрутов. Затем маршрутизаторы, получившие LSA, должны отправить еще LSA, сообщая всем об изменениях, которые они только что получили из первой волны LSA. И так продолжается до тех пор, пока все маршрутизаторы не будут иметь одинаковые данные о сети (а для этого требуется огромное количество пакетов).

Еще одно неудобство, вызываемое затоплением LSA, относится к объему памяти и мощности процессора, необходимых для «переваривания» обновлений. В зависимости от количества LSA в сети и количества обновлений в каждом объявлении маршрутизатор может посвящать большую часть доступной памяти и процессорного времени на обработку LSA (в попытке как можно скорее достичь конвергенции). Пакеты LSA всегда имеют более высокий приоритет (при обработке), чем пакеты маршрутизируемого протокола. Следовательно, те сетевые пакеты, которые вынуждены пробиваться через «пробки», образованные LSA, могут остаться необработанными.

Причина, по которой маршрутизатор состояния канала всегда обрабатывает LSA прежде, чем пакет маршрутизируемого протокола, проста. Вы хотите, чтобы среда достигла конвергенции как можно скорее. А чем быстрее будут обработаны и применены LSA, тем скорее маршрутизатор сможет возобновить свою обычную каждодневную работу.

Последний недостаток на самом деле не так уж страшен (насколько это вообще возможно). Обычно протоколы состояния канала обеспечивают более быструю конвергенцию, чем другие типы протоколов. Поэтому задержки, случающиеся в работе сети во время затопления, не очень длительны. Это должно быть хорошей новостью для администраторов, обеспечивающих работу IS-IS в больших сетях с минимальной пропускной способностью.

Управление затоплениями

IS-IS имеет несколько встроенных механизмов управления объемом LSA, отсылаемых в процессе затопления. Первый из них подобен правилу расщепления горизонта. То есть маршрутизатор IS-IS никогда не отправляет LSA тому же каналу, от которого он его получил. Благодаря этому обновление не пересылается бесконечно между двумя маршрутизаторами. Используем для пояснения рис. 19.9.

На рис. 19.9 показано, что маршрутизатор A получает обновление от маршрутизатора B на порт 1. L1A использует эту информацию для обновления своей таблицы маршрутов и отправки собственного LSA маршрутизатору C на порт 2. Но маршрутизатору B обновленная информация не отправляется. Дело в том, что маршрутизатор B инициировал затопление, и информация в обновлении касается только его собственных каналов, поэтому у B нет необходимости в получении обновленных табличных данных маршрутизатора A.

Если бы мы проследили этот пример до завершения процесса затопления, то увидели бы, что маршрутизатор C заблокировал путь между маршрутизатором A и собой. Далее все маршрутизаторы, получившие обновление от C, также блокируют соответствующие порты. Блокируя (аннулируя или отменяя, используется несколько терминов) канал между двумя маршрутизаторами после отправки сообщений, IS-IS минимизирует количество LSA, затапливающих сеть.

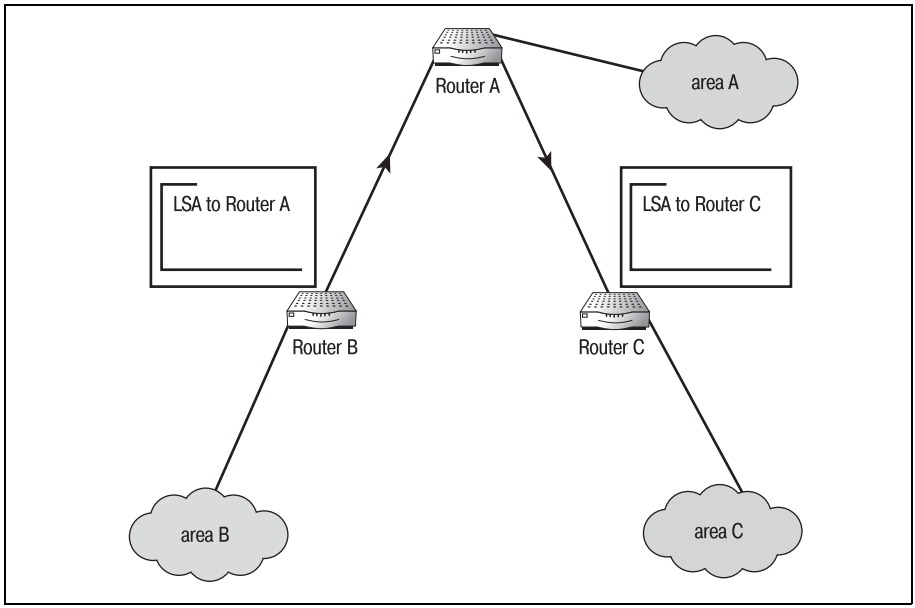

Рис. 19.9. *Маршрутизатор IS-IS, отправляющий и получающий обновления LSA*

Но что происходит, если маршрутизатор использует несколько портов? Справедливо ли вышесказанное? На рис. 19.10 изображен маршрутизатор, имеющий несколько портов для получения одних и тех же обновлений. Как маршрутизатор будет обрабатывать информацию?

На рис. 19.10 маршрутизатор A посылает свои обновления маршрутизаторам B и C. Оба эти маршрутизатора связаны с маршрутизатором D. Что произойдет, когда маршрутизатор D получит два LSA, содержащих одно и то же обновление, на разные порты? Понятно, что маршрутизатор может обработать только одно из обновлений. Но тогда он должен будет послать LSA тому маршрутизатору, от которого он до этого не получил обновление.

Вкратце ответ таков: маршрутизатор D обрабатывает первое полученное обновление, но не отправляет обновление ни одному из маршрутизаторов. Чтобы понять, почему так происходит, давайте заглянем внутрь пакета LSA. Два поля LSA определяют, может ли маршрутизатор IS-IS продолжать получать обновления и действительны ли обновления. Это поля *remaining life* и *sequence*.

Поле remaining-life – это временной параметр, используемый маршрутизатором после обработки LSA. Когда маршрутизатор исследует LSA, устанавливается таймер, соответствующий значению поля remaining life. И пока это время не истечет, маршрутизатор не может обрабатывать LSA, относящиеся к каналу (каналам), упомянутому в последнем объявлении.

Например, возвращаясь к рис. 19.10, маршрутизатор D получает LSA от маршрутизатора B, в котором сообщается, что канал между A и B недоступен. Маршрутизатор D обрабатывает LSA, устанавливает тай-

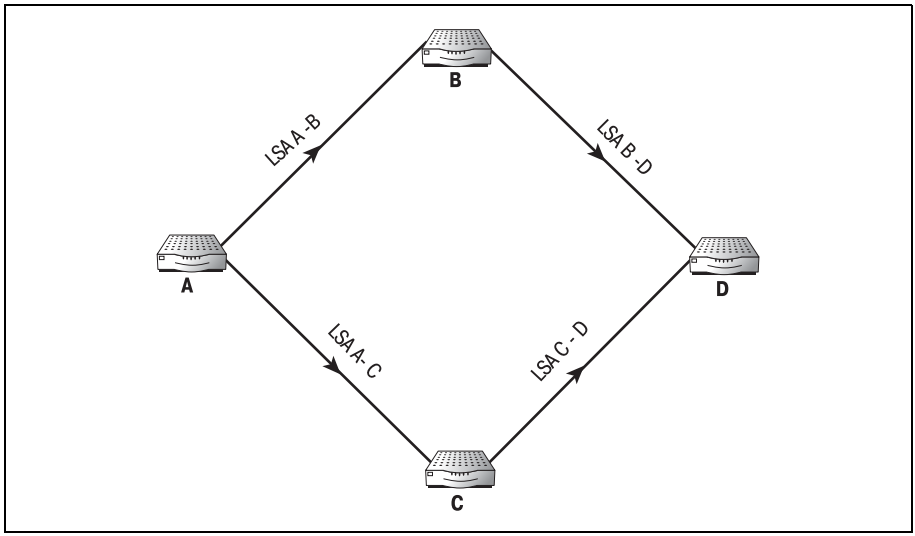

Рис. 19.10. *Маршрутизатор с несколькими принимающими портами*

мер remaining-life и обновляет свою таблицу маршрутов. Но маршрутизатор C, прежде чем получить такое же обновление, отправит свое обычное периодическое LSA. В этом LSA (направляемом маршрутизатору D) будет говориться, что канал между A и B функционирует.

Так как время, заданное таймером remaining-life, еще не истекло, маршрутизатор D (определив, что LSA относится к предыдущему обновлению) игнорирует обновление. Со временем маршрутизатор C получит обновленное LSA, и конвергенция будет достигнута. Кроме того, таймер remaining-life запрещает маршрутизатору D отправлять любые обновления, относящиеся к каналу между маршрутизаторами A и B, до тех пор, пока его время не истечет. Таким образом маршрутизаторы не получают ненужных противоречащих сообщений о состоянии канала и не отправляют многочисленных LSA, вызывающих те же проблемы.

Второе управляющее поле пакета LSA – это поле sequence (последовательность), которое используется принимающим маршрутизатором для идентификации обновления, содержащегося в LSA. Маршрутизатор не обрабатывает непоследовательные обновления LSA. Используя поля remaining life и sequence, IS-IS в состоянии контролировать затопления LSA, которые могут быть слишком массивными.

Еще одна хорошая новость для сетевых администраторов (которые умеют сводить сетевые изменения к минимуму) – они могут, в некотором роде, управлять затоплением LSA. В протоколах состояния канала реализован такой метод контроля (для ограничения количества ненужных потоков), как минимизация количества административных изменений сети.

Хотя это и не избавляет ото всех потоков, но уменьшает их частоту. Затопления LSA производятся периодически для выявления неадминистративных изменений и запускаются после такого рода изменений. Так что, ограничив количество административных изменений, вы уменьшите число затоплений. Не самый практичный способ снижения трафика, но очень действенный.

На самом деле администратор не может управлять периодичностью отсылки LSA, они будут отсылаться в любом случае. Но административные обновления инициируются изменением сетевой среды. Этим изменением может быть назначение новой метрики, неисправность канала или что-то еще, что изменяет возможный путь прохождения данных. События, приводящие к затоплению, представлены на рис. 19.11.

Обратите внимание на то, что все маршрутизаторы на рис. 19.11 находятся в рабочем состоянии, но процедура обновления все равно была запущена. Обновление вызвано изменением метрики, присвоенной каналу, соединяющему маршрутизаторы A и C. Эта присвоенная (произвольная) администратором метрика используется алгоритмом маршрутизации IS-IS для вычисления кратчайшего пути между двумя объектами.

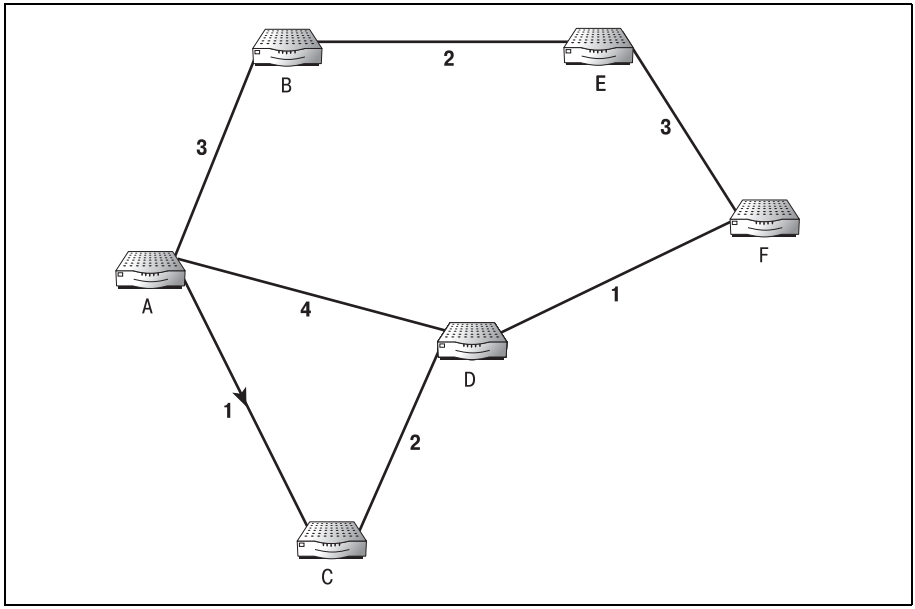

Рис. 19.11. *Запуск процесса затопления LSA*

Метрики и алгоритмы IS–IS

Протоколы, установленные на вашем маршрутизаторе, изначально не имеют сведений о том, какой путь является кратчайшим в любую точку сети. Они рассматривают и учитывают ряд элементов среды, производят с ними некоторые действия и получают окончательный результат. Метрики и алгоритмы помогают IS-IS принять решение о том, какой путь является кратчайшим в определенном направлении.

Метрики – это сетевые переменные, используемые для вычисления кратчайшего пути. Для алгоритмов состояния канала метрики – это значения, присваиваемые сетевым администратором. Многие сетевые протоколы применяют различные метрики, такие как пропускная способность, приоритет, стоимость и другие динамические или статические показатели. IS-IS упрощает процесс, используя всего одну метрику – стоимость.[1]

Наиболее важной задачей администратора среды IS-IS является назначение метрики стоимости всем каналам всех маршрутизаторов. На рис. 19.12 изображена сеть IS-IS, полностью снабженная метриками.

[1] В документе RFC 1142 «OSI IS-IS Intra-domain Routing Protocol» для этой метрики во избежание путаницы используется название «метрика по умолчанию», а под стоимостью (cost) маршрута понимается его совокупная оценка (то есть сумма всех метрик). – *Примеч. перев.*

Примечание

Существует еще три «необязательных» метрики, которые администратор может определить для канала. Но текущая реализация IS-IS распознает только метрику по умолчанию – стоимость. Все же перечислим дополнительные метрики:

- Задержка (delay) – время задержки для конкретного канала
- Затраты (expense) – величина затрат на поддержание работы данного канала
- Ошибка (error) – относительная величина, характеризующая количество ошибок, относящихся к данному каналу

Установка значений этих метрик никак не повлияет на работу алгоритма. Их смогут использовать будущие версии протокола IS-IS.

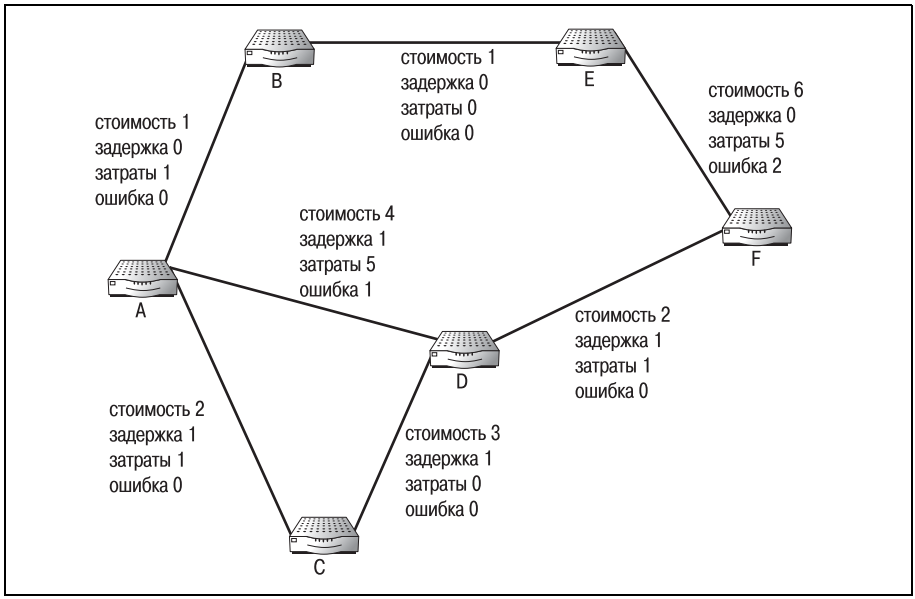

Рис. 19.12. *Сеть IS-IS, в которой всем каналам назначены метрики*

Что касается администратора IS-IS, он назначает метрику стоимости абсолютно произвольно. Нет никакой формулы или теории для выбора стоимости определенного канала. Существует лишь несколько правил, о которых следует помнить при назначении стоимости:

- Когда IS-IS применяет свой алгоритм ко всем метрикам сети, то кратчайший путь определяется наименьшей метрикой.
- Надежные каналы, например новые или не подверженные помехам (оптоволоконные каналы), всегда должны получать более низкие значения метрики.
- Наименее затратным каналам (например, принадлежащим компании, устанавливающей маршрутизаторы) должны присваиваться меньшие метрики, чем выделенным каналам, таким как ISDN.

- Каналам с более высокой пропускной способностью (таким как T3) следует назначать более низкие значения метрик, чем каналам с низкой пропускной способностью.

Какие бы метрики вы ни назначили, именно они будут решать, какой маршрут между двумя конечными точками следует выбрать в качестве кратчайшего. Но прежде чем вы займетесь присвоением метрик вашим каналам, знайте, что на метрику стоимости наложен ряд ограничений.

Стоимость отдельного канала не может превышать 64. Канал, которому присвоена стоимость 65, не опознается как действительный маршрут. Но повторяющиеся значения не запрещены. Поэтому, если у маршрутизатора более 64 портов, то вы можете определить для нескольких каналов одинаковые стоимости.

Примечание

Два любых канала с одинаковыми метриками, исходящие от одного маршрутизатора, имеют одинаковый вес для алгоритма. В этом случае IS-IS будет распределять пакеты по обоим каналам.

Еще одно ограничение касается полной стоимости пути (вычисляемой алгоритмом маршрутизации). IS-IS не может направлять данные по маршруту, общая стоимость которого превышает 1024.

Примечание

Если числа 64 и 1024 кажутся вам знакомыми, это объясняется тем, что они напрямую связаны с объемом памяти, используемым IS-IS для хранения маршрутной информации.

Прежде чем назначать метрики IS-IS, особенно в больших сетях, стоит немного подумать. Составьте полную схему маршрутизаторов и каналов. Назначьте все значения метрик на бумаге, сложите их и убедитесь в том, что они соответствуют правилам. Когда все проделано, вы можете вводить в эксплуатацию ваши маршрутизаторы IS-IS и передавать дела алгоритму маршрутизации.

Адресация IS-IS, области и домены

IS-IS, как и любой другой протокол, обращается к устройствам, используя их расположение в сети. IS-IS распознает устройства с помощью идентификатора местоположения, состоящего из двух частей. Идентификаторы указывают области и домены, которым принадлежит устройство. Чтобы понять адресацию IS-IS, давайте начнем с определения подразделений сети IS-IS: областей и доменов.

Разбивая сети IS-IS на области и домены, администраторы обеспечивают более полный контроль над потоком данных в сети. Эту идею удоб-

но пояснить в терминах карты дорог отдельного штата. Применив затем ту же логику к организации сетей, вы поймете обоснование разбиения сред IS-IS на области и домены.

Давайте рассмотрим в качестве примера штат Техас. В Техасе сотни тысяч дорог и магистралей. У каждой дороги есть свое название, а у каждой магистрали – номер. Так как в штат входит множество городов, то названия дорог в разных городах могут повторяться, номер же магистрали одного округа может использоваться в другом округе.

Города служат географическими маркерами внутри округов (а округа – в штате). Например, если вы ищете дорогу в Даллас, вы сразу же будете знать не только то, в какой части округа вам нужно оказаться, но и в какой части всего штата. Округа и города помогают удержать под контролем проблему поиска географических объектов.

Если же убрать все города и округа, то найти путь через штат будет очень тяжело. Попробуйте найти дом 123 по Main Street, не зная названия города или округа. Это практически невозможно. Все улицы городов и магистрали округов должны будут иметь уникальные имена, но все равно может случиться так, что вы никогда не сможете найти нужный вам адрес. Например, если вы ищете конкретную улицу, не зная ни города, ни округа, в котором она находится, вам нужно будет прочесать весь штат, чтобы найти ее. Даже если у вас будет карта дорог, но вы не будете знать, с какого места нужно начать поиск, это будет задачей для самых отчаянных.

Домены IS-IS являются эквивалентом округов штата, а области IS-IS – это города. Области и домены помогают маршрутизаторам быстро находить конечные системы, особенно в больших сетях.

Области IS-IS

Область IS-IS – это отдельный набор конечных и промежуточных систем. То есть сеть конечных и промежуточных систем, совместно использующих один номер (идентификатор) области, рассматривается как одна область. Область IS-IS изображена на рис. 19.13.

Примечание

Области и домены IS-IS имеют непосредственное отношение к архитектуре DECnet Phase IV. Все сети DECnet разделены на области.

Если вы разбираетесь в IP-сетях и организации подсетей IP, то вам несложно будет освоить области и домены IS-IS. Однако не существует прямого соответствия областей и доменов их IP-сородичам. Если говорить точнее, то области и домены IS-IS вместе эквивалентны подсети IP. Область IS-IS может пониматься как подподсеть.

Области обычно бывают достаточно маленькими и автономными. Проектировщику сети при создании областей не обязательно придержи-

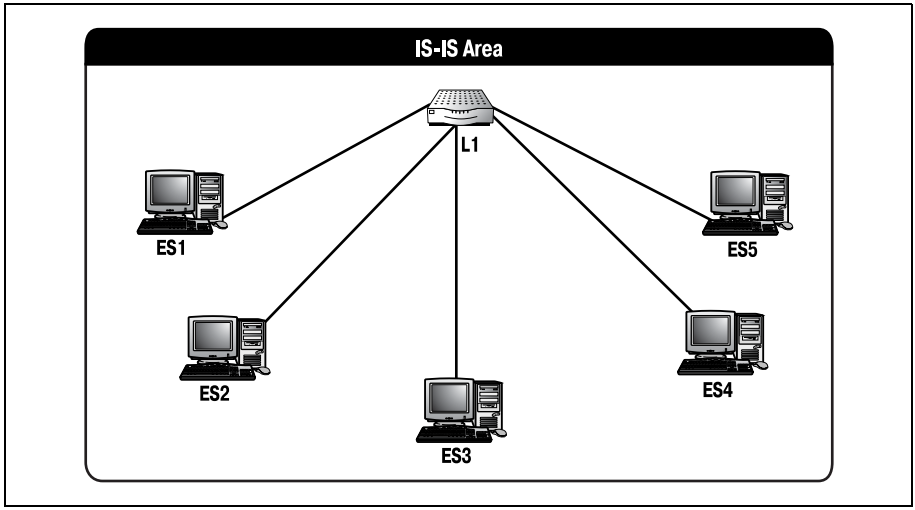

Рис. 19.13. *Область IS-IS*

ваться тех же правил, что и для IP-подсетей. Области IS-IS не так часто используются, как подсети IP. Их применение очень ограничено.

Маршрутизация внутри области, подробно описанная в разделе «Маршрутизация IS-IS» данной главы, осуществляется маршрутизаторами первого уровня. Как и в протоколе DECnet Phase IV, L1 используются для внутриобластной маршрутизации данных. Как и их аналоги в DECnet, маршрутизаторы L1 не могут маршрутизировать данные вне своей локальной области. Поэтому группа областей имеет больший набор требований к маршрутизации, чем сумма требований ее частей. (Другими словами, просто поместив несколько областей в одну комнату, вы не сможете гарантировать, что они будут взаимодействовать друг с другом. На самом деле они не будут этого делать.) Из-за наличия такого коммуникационного барьера группы связанных областей организуются в домены, которые имеют требования к маршрутизации, отличающиеся от имеющихся в более мелких, локализованных областях.

В терминах IS-IS домен – это группа областей, соединенных парой маршрутизаторов второго уровня. Только маршрутизаторы второго уровня могут образовывать домены. Одна область с маршрутизатором первого уровня, подключенная к маршрутизатору второго уровня (который также соединен с другой областью), не считается доменом. Полноценный домен IS-IS представлен на рис. 19.14. Обратите внимание на два L2, между которыми проходит магистраль. Именно наличие такой пары L2 (с областями по каждую строну) образует домен IS-IS. Ненастоящий домен IS-IS представлен на рис. 19.15.

Очень важно иметь план областей/доменов на бумаге, прежде чем начинать конфигурировать вашу сеть IS-IS. Если вы по невнимательнос-

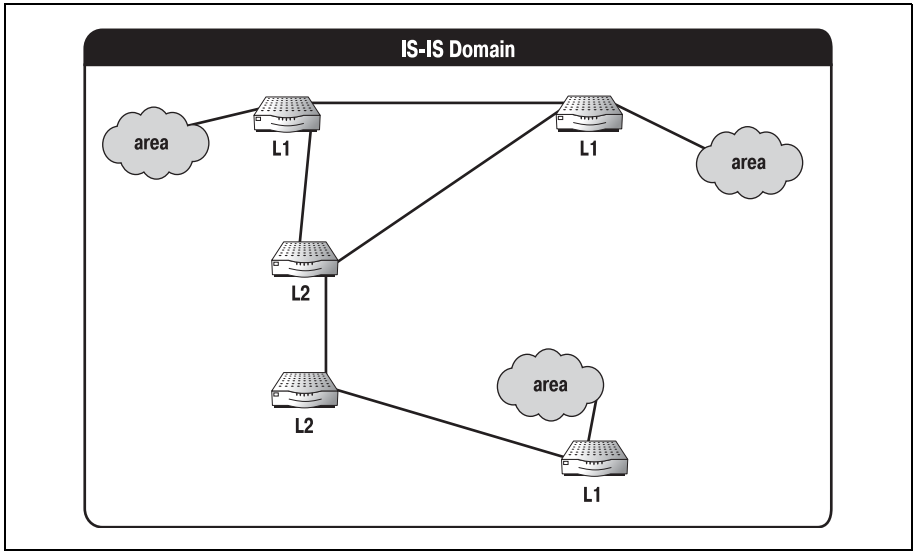

Рис. 19.14. *Правильно организованный домен IS-IS*

ти оставите конфигурацию область/домен в таком виде, как показано на рис. 19.15, то не сможете обеспечить правильную маршрутизацию.

Но, как и для любого другого протокола, просто правильного размещения устройств недостаточно для корректной работы. Физической топологии должны соответствовать протокольные адреса. Адресная схема IS-IS представляет собой смесь адресов, соответствующих правилам OSI, и MAC-адресов область/узел DECnet Phase IV.

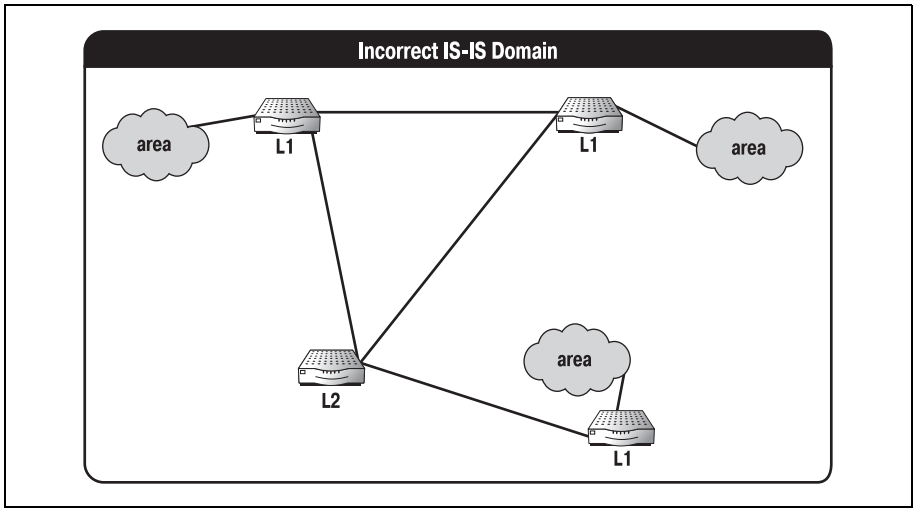

Рис. 19.15. *Неправильно организованный домен IS-IS*

Адреса IS–IS

Адресная схема IS-IS порождена схемами OSI и DECnet Phase IV. И это не случайно. Комбинирование двух архитектур было необходимо для обеспечения возможности взаимодействия IS-IS и DECnet, базирующегося на DNA. Протокол может без труда маршрутизировать данные на различных платформах для разных маршрутизируемых протоколов. Адреса IS-IS также являются ключом к определению областей и доменов маршрутизации IS-IS.

Чистокровные OSI-протоколы используют адрес, называемый NSAP (network service access point – точка доступа к сетевому сервису). NSAP определяет положение конечной системы в сети, вплоть до области, в которой она находится. Так как IS-IS соответствует стандарту OSI, он также использует NSAP-адреса. Но IS-IS – это не только OSI-протокол маршрутизации, поэтому он интерпретирует адрес несколько иначе, чем другие протоколы OSI.

Примечание

Действительное предназначение NSAP состоит в адресации некоторой точки сетевого уровня (модели OSI) на определенном устройстве. Вместо того чтобы произвольным образом присваивать адрес машине, в стандарте OSI было решено адресовать точку входа информации в устройства.

Форматы адресов устройств практически одинаковы, и NSAP-адрес не является исключением. Как и большинство протокольных адресов, NSAP-адрес разделен на две части, чтобы более точно отражать местоположение устройства. Адрес NSAP разбивается на IDP (initial domain part – исходная часть домена) и DSP (domain-specific part – специфичная часть домена). Сочетание этих двух частей определяет расположение точки входа на сетевом уровне устройства.

IDP – это часть адреса, относящаяся к домену, в котором находится устройство. IDP, в свою очередь, разбивается на AFI (authority and format identifier – идентификатор формата и полномочий) и IDI (initial domain identifier – исходный идентификатор домена). AFI определяет полномочное лицо, присваивающее адрес домену. IDI указывает общую информацию о домене.

Вторая часть NSAP, DSP (domain-specific part) содержит точную адресную информацию. DSP хранит адрес домена, которому принадлежит устройство. Это справедливо для OSI-протоколов; однако IS-IS не является чисто OSI-протоколом. IS-IS делит адрес NSAP на две основные части: адрес области и идентификатор системы (третья часть, прикрепленная в конец NSAP, – это селектор адреса, который почти всегда устанавливается в 0). Адрес области, включающий в себя IDP адреса NSAP, определяет область (внутри домена), которой принадлежит устройство, в то время как идентификатор системы (очень похоже на MAC-адрес) адресует само устройство.

NSAP-адреса областей

Адрес области, состоящий из IDP адреса NSAP, указывает область, которой принадлежит устройство. Все устройства одной области имеют одинаковые адреса области. Администратор назначает этот адрес, состоящий как минимум из одного шестнадцатеричного октета, на этапе проектирования сети.

В среде, включающей несколько областей, адрес области очень важен для маршрутизации. Все конечные системы и маршрутизаторы первого уровня имеют один и тот же адрес области, что обеспечивает корректную идентификацию всех систем области со стороны устройств, находящихся за ее пределами.

Идентификатор системы в NSAP

Идентификатор системы определяет систему внутри области. Подобно тому как системы DECnet Phase IV используют идентификатор область/узел для изменения MAC-адреса, системы IS-IS используют измененный MAC-адрес для создания идентификатора системы в NSAP. Характеризующий устройство идентификатор системы никогда не разделяется другими устройствами и не дублируется в сети IS-IS. Поэтому идентификатор системы является надежным маркером системы в области.

Пакеты IS-IS

IS-IS маршрутизирует данные в сетевой среде, используя пакеты IS-IS. Эти пакеты инкапсулируют данные, которые IS-IS перемещает по сети. Помните, однако, что не все пакеты предназначены для пересылки данных. Некоторые пакеты используются для внутренних задач устройств IS-IS. Такие пакеты помогают IS-IS определить конфигурацию и топологию устройств, соединенных со средой IS-IS.

В сетях IS-IS используется три типа пакетов: приветственные сообщения (hello message – HM), пакеты состояния канала (link state packet – LSP) и пакеты номеров последовательностей (sequence number packet – SNP). В этом разделе будет рассказано о том, как IS-IS использует каждый из вышеперечисленных типов для содействия маршрутизации данных.

Если вы проведете анализ любого пакета, то увидите, что его можно разделить на три основные части: заголовок, данные и контрольную сумму. Типичный пакет изображен на рис. 19.16.

Большая часть заголовков пакетов имеет фиксированную длину, определяемую типом пакета. Чаще всего заголовки (например, в пакетах Ethernet или IS-IS) состоят из 8 байт, но бывает и иначе. В некоторых случаях после заголовка может стоять индикатор длины заголовка. Индикатор длины заголовка сообщает принимающему устройству

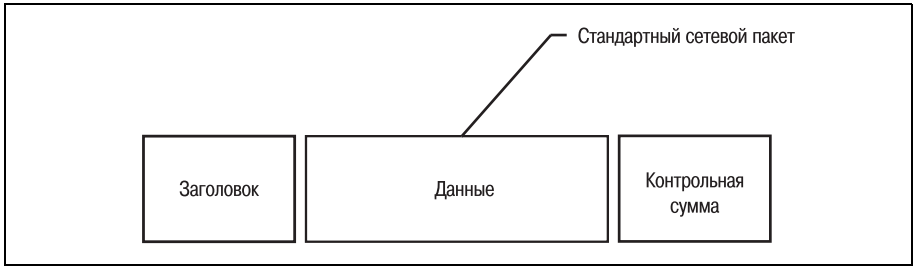

Рис. 19.16. *Стандартный пакет*

длину заголовка. Благодаря этому устройство может без труда определить, где заканчивается заголовок и начинаются данные.

Часть «данные» – это и есть те данные, которые одна система хочет отправить другой. Эта часть пакета имеет переменную длину (зависящую от объема отсылаемой информации).

Контрольная сумма – это обычно последняя часть пакета. Контрольная сумма представляет собой содержимое пакета в сжатой форме, которую устройство может воспроизвести. Заново вычислив контрольную сумму для полученных данных и сравнив результат с полученным значением контрольной суммы, устройство может определить, не поврежден ли пакет и доставлен ли он полностью,

Все пакеты IS-IS имеют один и тот же базовый формат. Пакет IS-IS состоит из заголовка, данных и расширенных данных (рис. 19.17).

Рассмотрим восьмибайтный заголовок пакета. Вне зависимости от типа пакета IS-IS (HM, LSP или SNP) заголовок состоит из восьми 1-байтных полей, которые предоставляют принимающему устройству всю информацию, необходимую для корректной обработки пакета.

На рис. 19.18 представлен пакет IS-IS с заголовочной информацией. Восемь полей заголовка IS-IS включают:

- Идентификатор протокола – идентифицирует пакет как пакет IS-IS
- Длина – хранит величину, равную длине заголовка

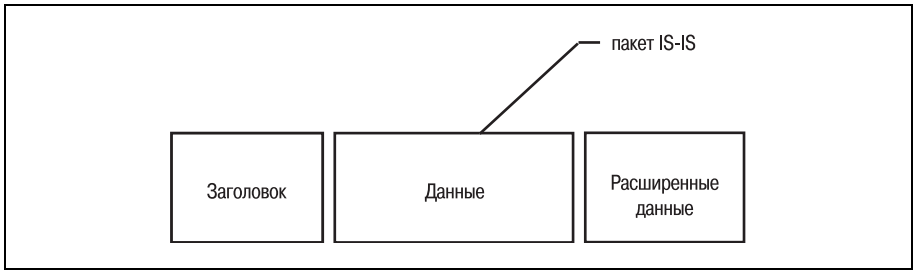

Рис. 19.17. *Пакет IS-IS*

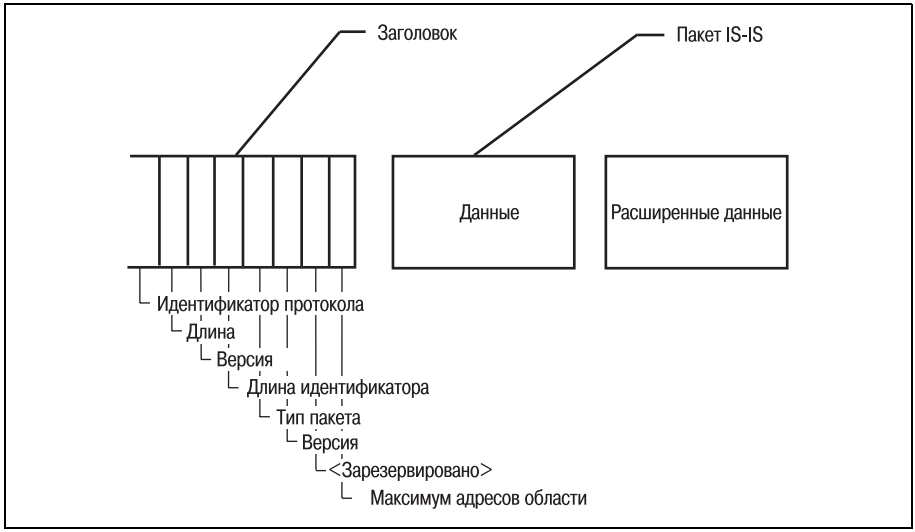

Рис. 19.18. *Пакет IS-IS с заголовочной информацией*

- Версия (1) – указывает версию IS-IS
- Длина идентификатора – указывает длину той части адреса получателя, которая занята идентификатором
- Тип пакета – определяет тип пакета (LSP, HM или SNP)
- Версия (2) – повторно сообщает номер используемой версии IS-IS
- Зарезервировано – поле зарезервировано для дальнейшего использования
- Максимум адресов области (MAA – maximum address area) – указывает максимальное количество адресов для данной области

Такой формат имеют все пакеты IS-IS. А вот поля, следующие за заголовком, зависят от типа пакета. Поля после заголовка – это сами данные, отправляемые из системы в систему.

Каждый из трех типов пакетов IS-IS содержит свои уникальные поля, которые и будут рассмотрены в следующих разделах.

Сообщения hello

Сообщения hello используются устройствами IS-IS для извещения остальных о своем присутствии и функционировании в сети. Полный пакет сообщения hello с заголовочной информацией изображен на рис. 19.19. Сообщение hello протокола IS-IS содержит следующие поля:

- Идентификатор источника – идентификатор системы для устройства, отправившего сообщение
- Заданный вручную адрес – список адресов области

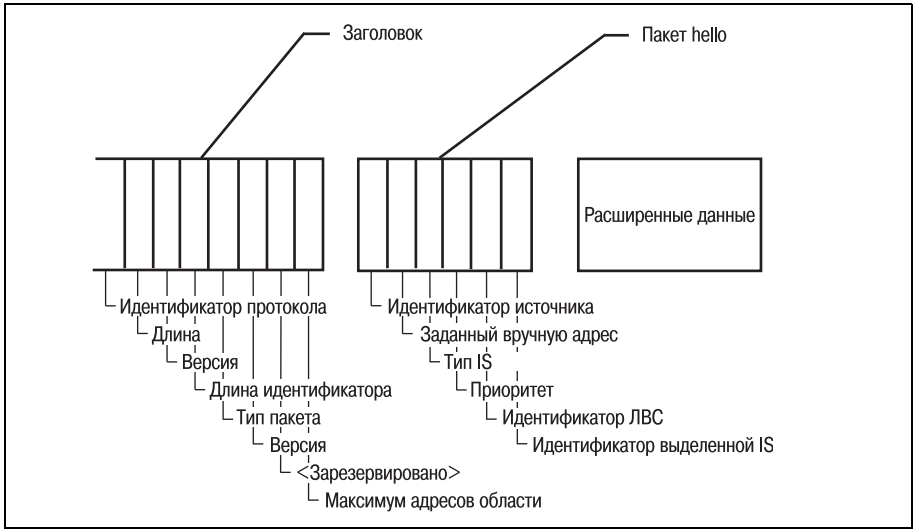

Рис. 19.19. *IS-IS сообщение hello*

- Тип промежуточной системы – указывает уровень (L1 или L2) промежуточной системы (IS), отправляющей обновление
- Приоритет – указывает назначенный администратором приоритет маршрутизатора
- Идентификатор ЛВС – указывает идентификатор локальной сети
- Идентификатор выделенной IS – указывает идентификатор системы для выделенной IS

Промежуточная система периодически отправляет сообщение hello, чтобы информировать другие IS в сети о том, что она в состоянии принимать информацию. Пакет также сообщает принимающей IS назначенный администратором приоритет, тип промежуточной системы и адрес IS, отправившей сообщение hello.

Когда IS получает сообщение hello, она отвечает на него своим собственным сообщением hello. Это подтверждает, что оба канала функционируют, и в сетевой среде не произошло никаких изменений. После подтверждения наличия связи между двумя устройствами две промежуточные системы добавляют информацию друг о друге в свои списки соседних IS.

Пакеты состояния канала

LSP (пакеты состояния канала) – это основа всех протоколов состояния канала. LSP служит средством доставки всех обновлений маршрутов. В процессе затопления все промежуточные системы сети отправляют LSP для обновления таблиц маршрутов других маршрутизаторов среды.

LSP должен содержать всю информацию, которая необходима маршрутизатору для корректной пересылки данных от одной IS к другой. В условиях отсутствия корректной актуальной информации данные, отправленные в сеть, могут никогда не достигнуть своего адресата. LSP протокола IS-IS содержит следующие поля:

- Идентификатор источника – идентификатор системы для устройства, отправившего сообщение
- Заданный вручную адрес – список адресов области
- Тип промежуточной системы – указывает уровень (L1 или L2) промежуточной системы (IS), отправляющей обновление
- Идентификатор системы (со стоимостью) – содержит всех соседей отправляющей IS и их стоимости
- Идентификатор выделенной IS – указывает идентификатор системы для выделенной IS
- Статическая смежность – содержит все статически сконфигурированные IS и их стоимости

В протоколе IS-IS существуют лишь два отличия в полях сообщения hello и LSP. Но эти два отличия – это отличия маршрутизируемой сети от немаршрутизируемой. LSP содержит поля для обновления таблиц маршрутов. Когда промежуточная система получает LSP, она вставляет идентификатор системы отправляющего устройства в свою таблицу маршрутов. Затем она добавляет к этой записи список непосредственных соседей данной IS и их стоимости.

Примечание

Помните, что, так как маршрутизаторы первого уровня имеют сведения только о своей локальной области, любая внешняя информация, которая может быть получена, отбрасывается.

После того как LSP получены и обработаны, промежуточная система запускает для новых данных алгоритм Дейкстры, чтобы вычислить новые оптимальные пути. На рис. 19.20 изображен пакет состояния канала.

Пакеты номеров последовательностей

Пакеты номеров последовательностей (sequence number packet – SNP) гарантируют, что каждая промежуточная система, обновляемая LSP, получает корректный LSP. Если IS получает SNP, идущий не по порядку, то сопутствующее пакету обновление состояния канала игнорируется. SNP содержит следующие поля:

- Оставшееся время жизни – оставшееся время жизни для LSP (в течение этого времени промежуточная система не может принимать другие обновления)

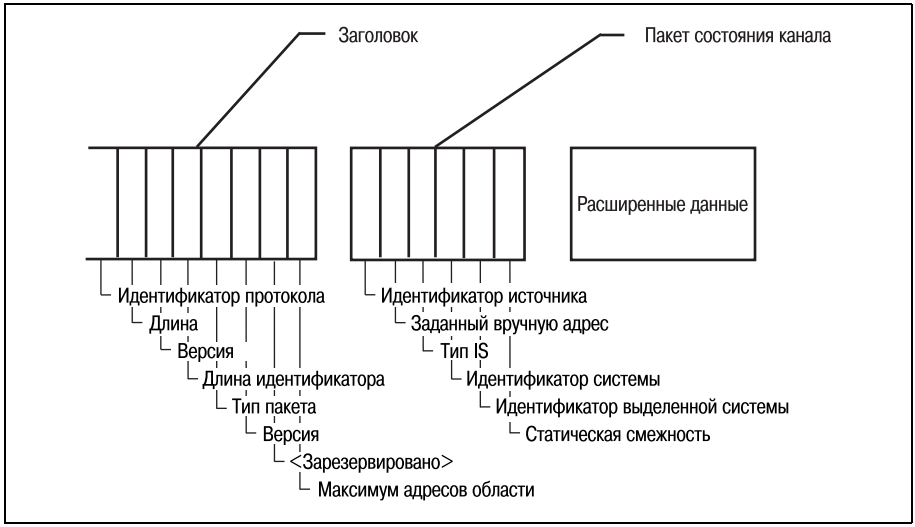

Рис. 19.20. *Пакет состояния канала*

- Идентификатор LSP – идентификационный номер LSP
- Порядковый номер LSP – порядковый номер обновления
- Контрольная сумма – используется для проверки целостности пакета

Эти поля сообщают принимающей IS, корректно ли полученное обновление и в течение какого времени оно действительно. На рис. 19.21 представлен пакет номеров последовательности, включая заголовочную информацию.

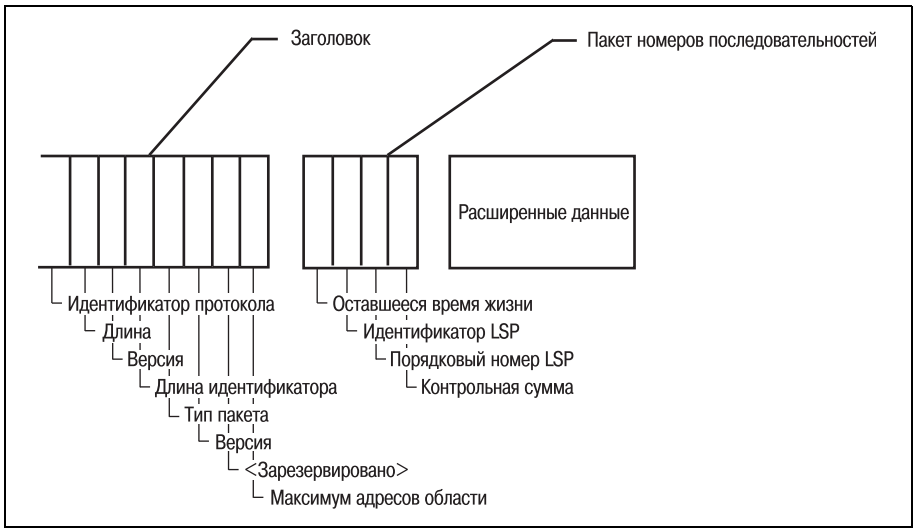

Рис. 19.21. *Пакет номеров последовательностей*

Все описанные пакеты играют важную роль в успешной работе среды IS-IS. В следующем разделе мы поговорим о том, как все эти элементы используются при маршрутизации данных в сети IS-IS.

Маршрутизация IS–IS

Теперь, когда вы познакомились с историей IS-IS и технологией, лежащей в основе работы протокола, приступим к изучению маршрутизации IS-IS. Маршрутизацию IS-IS обеспечивают успешные своевременные обновления топологической информации, относящейся к локальным областям IS-IS.

Для эффективной маршрутизации сетям IS-IS необходима некоторая «связность». То есть все промежуточные системы сети IS-IS не могут просто делать все, что им захочется. Им нужно от кого-то (или, точнее, от чего-то) получать указания.

Протокол IS-IS в своих областях и доменах назначает одну IS главной, или выделенной. Такая IS отвечает за действия среды.

Примечание

Понятие «главного узла», присматривающего за работой сети, не ново. Если вы знакомы с главным броузером сетей Microsoft или активным монитором Token Ring, то вам должно быть понятно, что такое главный узел.

Первым принципом, который мы рассмотрим, будет назначение выделенной промежуточной системы. Узлы IS-IS выбирают выделенную IS, которая будет инициатором сетевых событий.

Выделенная IS

Основная работа выделенной системы заключается в отсылке пакетов состояния канала, включающих информацию обо всей локальной сети. Выделенная IS (собрав обновления состояния каналов со всей сети) генерирует полную картину сетевого окружения и рассылает ее всем остальным промежуточным системам локальной сети. Тем самым ограничивается количество избыточных LSP, которые могут приходить промежуточной системе.

Когда IS получает пакет состояния канала от выделенной IS, по нему проверяется правильность всех остальных LSP. Это помогает ускорить конвергенцию.

В процессе выбора промежуточной системы на роль выделенной промежуточные системы сравнивают свои приоритеты. После проведения полного сравнения промежуточная система с наивысшим приоритетом назначается выделенной. Если несколько IS имеют одинаково высокие приоритеты, производится отбор по идентификатору.

Примечание

Приоритет промежуточной системы – это одно из значений, присваиваемых администратором. Сетевой администратор должен назначить каждой IS приоритет (произвольное числовое значение, при этом наивысшим является 1). Это значение может быть изменено и, как и метрика стоимости, задается только администратором.

Поле приоритета также используется в сообщениях hello протокола IS-IS для указания статуса IS. (Если приоритет промежуточной системы изменяется, это приводит к затоплению LSP и новым выборам.)

Промежуточные системы, имеющие одинаково высокие приоритеты, сравнивают свои MAC-адреса. Выделенной назначается IS, MAC-адрес которой обладает наибольшим числовым значением. Выделенная промежуточная система продолжает работу по сбору и рассылке LSP от имени всей ЛВС в целом (для которой вводится понятие «псевдоузел»).

Если одна промежуточная система имеет интерфейсы, которые соединяют несколько локальных сетей, то она участвует в выборах выделенной промежуточной системы во всех из них. Другими словами, выборы происходят в той локальной сети, в которой присутствует промежуточная система. Если промежуточная система присутствует в нескольких сетях, то она участвует в нескольких выборах. Поэтому существует возможность того, что одна IS будет выбрана выделенной для нескольких сетей IS-IS. В любом случае выделенная IS рассматривает все остальные промежуточные системы своей ЛВС как один объект, который и называется псевдоузлом.

Псевдоузлы

После того как выделенная IS выбрана, остальные IS группируются вместе в псевдоузел. Псевдоузел интерпретируется как один объект, хотя на самом деле является совокупностью промежуточных систем. Сети IS-IS используют такие псевдоузлы для получения информации о локальной сети в целом.

Промежуточная система, входящая в псевдоузел, принимает LSP от выделенной IS. В процессе затопления LSP каждая IS отправляет LSP каждой IS, с которой она связана. (Очевидно, что IS может отправить обновление только устройству, с которым она непосредственно связана.) Такой LSP (называемый непсевдоузловым LSP) содержит всю информацию о каналах, непосредственно подключенных к IS.

На рис. 19.22 изображено затопление LSP для четырех IS. В процессе этого затопления маршрутизатор A получает LSP от маршрутизатора B, сообщающего, что все его каналы (маршрутизатор C) находятся в рабочем состоянии.

В своей таблице маршрутов маршрутизатор A помечает, что B и C работают должным образом. Но у маршрутизатора A нет возможности узнать о состоянии маршрутизатора D.

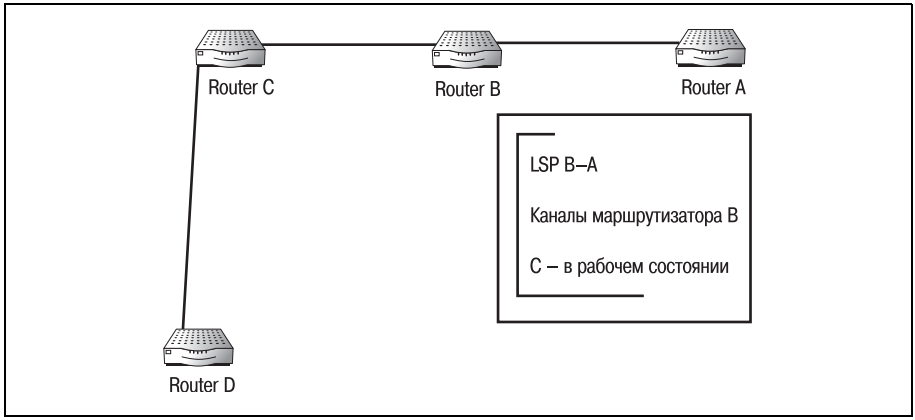

Рис. 19.22. *Затопление пакетами о состоянии канала*

Если бы каждый маршрутизатор отправлял всем остальным маршрутизаторам обновления для всей сети, то возник бы бесконечный цикл обновлений. Маршрутизатор A получал бы обновления, касающиеся маршрутизатора D, от маршрутизаторов B, C и D. Проблему решает введение понятия псевдоузла.

Задача выделенной IS заключается в сборе информации об обновлениях и создании «главной таблицы» (master table) или картины сетевой топологии. Затем выделенная IS отправляет одно обновление псевдоузлу. Все IS, входящие в псевдоузел, обрабатывают эту информацию, сравнивая с обновлениями, полученными от соседей. Все IS обновлены, а сеть избежала зацикливания обновлений.

Каждая IS для обеспечения уникальности имеет свой, отличный от остальных адрес. Протокол IS-IS должен принять меры для того, чтобы адресовать пакеты группе узлов как одному псевдоузлу. Это достигается за счет использования групповой адресации.

Существуют четыре Ethernet-адреса, которые IS-IS может использовать для групповой адресации. Иначе говоря, используемый адрес зависит от той группы устройств, к которой он применяется. В табл. 19.1 перечислены групповые адреса, используемые протоколом IS-IS, и их пункты назначения.

Таблица 19.1. Групповые адреса Ethernet для IS-IS

Адрес	Описание
0180C2000014	Используется для связи только со всеми L1
0180C2000015	Используется для связи только со всеми L2
09002B000005	Используется для связи со всеми промежуточными системами; именно этот адрес используется для общения с псевдоузлом
09002B000004	Используется для связи только со всеми конечными системами

Все узлы сети IS-IS умеют обрабатывать пакеты, отправленные на групповой адрес. Выделенной IS нужно просто направить одно обновление на групповой адрес псевдоузла, и оно будет обработано всеми устройствами.

Маршрутизация IS–IS

Маршрутизаторы первого и второго уровней по-разному перемещают данные в среде IS-IS. Причина этого проста: маршрутизаторы второго уровня имеют более широкий круг обязанностей, чем L1. Маршрутизаторы второго уровня отвечают за перемещение данных в области, домене и всей сетевой среде, в то время как маршрутизаторы первого уровня занимаются только локальными областями.

Исходная конечная система отправляет пакет, подлежащий маршрутизации, на ближайший (непосредственно связанный с ней) маршрутизатор L1. Так как конечная система сама не может перемещать данные, они должны быть отправлены маршрутизатору L1 или L2. Теперь L1 должен принять решение о дальнейшей судьбе пакета на основе информации, содержащейся в заголовке пакета.

Маршрутизатор L1 просматривает заголовок, ища адрес назначения пакета (адрес назначения содержит два важных элемента информации: область и домен назначения). Найдя адрес назначения, L1 распознает область назначения.

Маршрутизатор L1 сравнивает область назначения пакета со своей собственной областью. (Помните, что L1 обладает знаниями только о своей области, и если область назначения не является домашней областью L1, он пересылает пакет следующему маршрутизатору.) Проведя сравнение, L1 приходит к одному из двух возможных заключений: пакет необходимо перемещать внутри локальной области или же его следует переслать на соседний маршрутизатор.

Примечание

К этому времени L1 уже получил LSP, и для обновленной информации был запущен алгоритм Дейкстры. Поэтому можно считать, что L1 работает, имея полную и актуальную картину сети.

Если область назначения совпадает с локальной областью, то маршрутизатор заново просматривает заголовок в поиске системы назначения. Адрес системы назначения, который также является частью адреса назначения, указывает L1, какой именно конечной системе следует переслать пакет. Если область назначения совпадает с локальной областью, но система назначения маршрутизатору неизвестна, то пакет удаляется.

Если область назначения не совпадает с локальной областью L1, то маршрутизатор исследует свою таблицу маршрутов в поиске соседнего маршрутизатора, который мог бы доставить пакет в область назначе-

ния. Пакет переправляется тому L1, который управляет областью назначения. Если не удается найти область назначения, то пакет удаляется. Процесс маршрутизации изображен на рис. 19.23.

Для того чтобы осознать все это, посмотрите на схему, представленную на рис. 19.23. Хотя и без этой схемы понять процедуру маршрутизации L1 несложно. Протокол IS-IS – это эффективный протокол маршрутизации, который достаточно быстро перемещает данные с одного маршрутизатора на другой. Решений о выборе маршрута приходится принимать не слишком много, а разделение обязанностей между маршрутизаторами первого и второго уровней делает процесс еще проще, позволяя маршрутизатору сосредоточиться на одной области и не думать обо всей большой сети.

Процесс маршрутизации для L2 несколько отличается от вышеописанного. Когда L2 получает пакет, первые выполняемые им шаги идентичны L1: маршрутизатор просматривает заголовок в поиске адреса назначения, сравнивает этот адрес со своим собственным. Так как L2 может быть напрямую связан с несколькими областями, он определяет область назначения и сравнивает ее с собственной группой «областных» L1. Если область назначения совпадает с одним из соседей первого уровня, то пакет передается маршрутизатору первого уровня области назначения.

Однако если адрес назначения не совпадает ни с одной из известных маршрутизатору областей, то он смотрит на префиксную информацию.

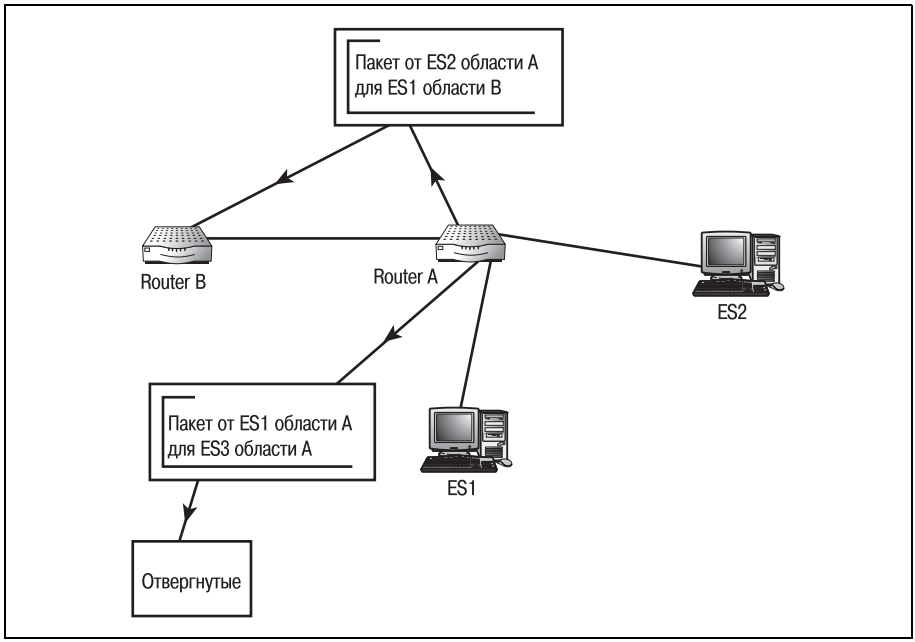

Рис. 19.23. *Маршрутизация L1 в IS-IS*

Примечание

К областям, которые не являются локальными для L2, в его таблице маршрутов добавлен префикс. Это не только позволяет маршрутизатору различать локальные и нелокальные области, но и помогает определить, куда отправлять нелокальные пакеты.

Локальные и нелокальные области называют внутренними и внешними маршрутами соответственно. Обычно внутренние маршруты находятся внутри домена. То есть в маршрутизации будут участвовать всего два маршрутизатора L2. Внешние маршруты могут проходить через несколько доменов и любое количество L2.

Если пакет предназначен для внутреннего пути, то он пересылается соответствующему маршрутизатору L1. Рисунок 19.24 иллюстрирует внутреннюю маршрутизацию.

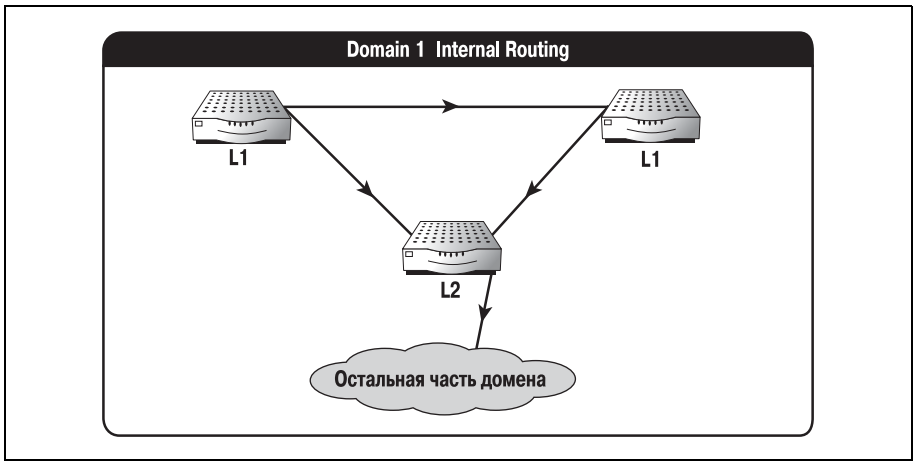

***Рис. 19.24.** L2-маршрутизация внутренних путей*

Если пакет предназначен для внешнего пути, то маршрутизатор L2 пересылает пакет своему L2-соседу. Внешняя маршрутизация поясняется на рис. 19.25.

Если не удается определить область назначения, пакет удаляется.

Маршрутизация IS-IS может быть сложной и требовать дополнительных знаний. Но в этой главе мы хотели представить краткий обзор IS-IS, показав, чем он отличается от других протоколов маршрутизации. Последний раздел будет посвящен конфигурированию протокола IS-IS на маршрутизаторе Cisco.

Конфигурирование IS–IS

Первым шагом в настройке маршрутизатора Cisco на использование IS-IS является разрешение маршрутизации IS-IS и присвоение номера области. Конфигурирование IS-IS выполняется в режиме глобального

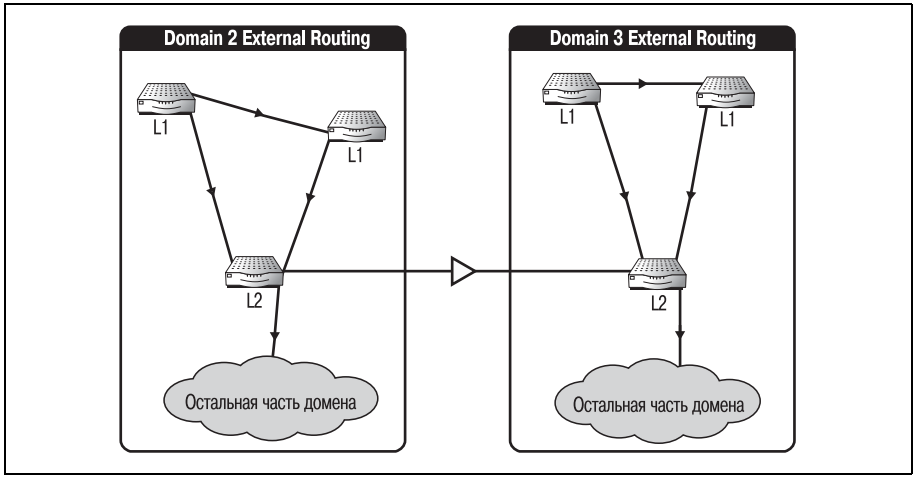

Рис. 19.25. *L2-маршрутизация внешних путей*

конфигурирования. Для разрешения маршрутизации IS-IS выполните команду `router isis`:

```
Router#configure terminal
Router(config)#router isis 45
```

Параметр `45` в приведенном фрагменте – это номер области, которую нужно маршрутизировать. После того как команда `isis` выполнена, необходимо установить сетевой адрес. Команда `net` присваивает маршрутизатору сетевой адрес. Введите команду `net` следующим образом:

```
Router(config-router)# net 24.0001.001a.0000.0017.00
```

Эти две команды разрешают маршрутизатору Cisco участвовать в работе сети IS-IS для области 45. Для создания IS-IS-маршрутизатора необходимы только две данные команды. Но существует также несколько необязательных команд, которые могут оказаться полезными. Это команды `is-type` и `access-list`.

Команда `is-type` используется для определения того, является ли данный маршрутизатор областным или же магистральным. В команде `is-type` маршрутизатор первого уровня – это областной маршрутизатор, а маршрутизатор второго уровня – магистральный. Задать тип маршрутизатора при помощи команды `is-type` можно так:

```
Router(config-router)# is-type level-1
```

Наконец, вы можете захотеть определить несколько маршрутов по умолчанию, ведущих в вашу автономную систему. Для установки маршрута по умолчанию необходимо выполнить несколько шагов. Во-первых, нужно создать список доступа. Список доступа задает правила, по которым работает маршрут по умолчанию. Во-вторых, нужно создать карту маршрутов, ссылающуюся на список доступа. Затем нужно

сопоставить маршрут по умолчанию новой карте маршрутов. Используйте для задания маршрута по умолчанию следующие команды:

```
Router(config)#access-list 1 permit 10.0.0.0 0.255.255.255
Router(config)#route-map new
Router(config-route-map)#match ip route-source 1
Router(config-route-map)#^Z
Router(config)#router isis 399
Router(config-router)#default-information originate route-map new
Router(config-router)#^Z
```

Примечание

Об использовании и синтаксисе списков доступа будет подробно рассказано в следующей главе.

После выполнения таких команд маршрутизатор будет настроен для работы с IS-IS-областью 399, и будет определен маршрут по умолчанию, пропускающий входящие данные от сети 10.0.0.0.

Хотя есть еще много других команд, относящихся к работе протокола IS-IS, тех, которые мы рассмотрели здесь, достаточно для того, чтобы получить первое представление о функционировании маршрутизаторов Cisco.

Резюме

- IS-IS – это протокол маршрутизации, который был создан для использования в системе DECnet.
- Позже протокол IS-IS был адаптирован для работы с IP и превратился в надежный протокол маршрутизации с широкими функциональными возможностями.
- Будучи протоколом состояния канала, IS-IS использует для вычисления наилучшего пути следующие метрики: стоимость (метрика по умолчанию), задержка, затраты и ошибка.

Вопросы и ответы

Вопрос Существует множество протоколов маршрутизации, которые уже работают с IP, зачем же нужно было адаптировать IS-IS, созданный для DECnet?

Ответ IS-IS является одним из наиболее устойчивых среди используемых протоколов состояния канала. Размер IP-сетей увеличился, а структура усложнилась настолько, что традиционно применявшиеся в IP протоколы маршрутизации стали испытывать проблемы с их поддержкой. Поэтому, так как очередная версия DECnet уже склонялась к поддержке IP, адаптация IS-IS представлялась логичным следующим шагом.

Тест

Вопросы

1. Сколько метрик по умолчанию использует IS-IS?
2. Где в доменах IS-IS находятся маршрутизаторы L2?
3. Какие три шага необходимо выполнить для конфигурирования маршрута по умолчанию для маршрутизатора IS-IS?

Ответы

1. Одну метрику – стоимость.
2. На магистрали IS-IS.
3. Определить список доступа, создать карту маршрутов, ссылающуюся на этот список, и сопоставить маршрут по умолчанию карте маршрутов.

Упражнение

1. Сконфигурируйте маршрут по умолчанию для маршрутизатора в IS-IS-области 4. Маршрутизатор должен принимать весь трафик от сети 198.118.26.0.

20

Основы обеспечения безопасности Cisco

Данная глава посвящена двум основным методам обеспечения безопасности, реализованным в маршрутизаторах Cisco. Спросите любого, кто знаком с маршрутизаторами Cisco, что сегодня считается наиболее важным в электронном бизнесе, и ответ будет один – безопасность. Связано ли это с предотвращением утечки ваших внутренних данных или с минимизацией рисков путем ограничения доступа к внутренней сети, вопросы безопасности остаются на первом месте.

Понятие безопасности, даже в применении к Cisco, является весьма обширным и сложным. В этой главе представлены два основных метода, знакомство с которыми обязательно даже для начинающего пользователя Cisco: списки доступа IP и трансляция сетевых адресов (NAT – Network Address Translation). Комбинируя NAT и списки доступа, можно создать простую и в то же время достаточно безопасную систему. Обе эти технологии доступны на большинстве маршрутизаторов Cisco и в большинстве версий IOS.

Трансляция сетевых адресов выполняется обычно на шлюзе, то есть на маршрутизаторе, имеющем непосредственный выход в Интернет. Принцип действия NAT заключается в том, что одному из интерфейсов маршрутизатора назначается маршрутизируемый IP-адрес (или группа адресов), а другой интерфейс получает немаршрутизируемый адрес, принадлежащий внутренней сети. Задачей NAT является трансляция внутреннего немаршрутизируемого трафика во внешний, маршрутизируемый. Трансляция адресов хорошо подходит для обеспечения начального уровня безопасности, так как значительно усложняет задачу злоумышленникам, решившим «подсесть» на входящий и исходящий трафик сети.

Трансляция адресов работает рука об руку с еще одним базовым методом обеспечения сетевой безопасности: списками доступа IP. Списки доступа предоставляют администратору механизм для разрешения и запрещения трафика в соответствии с установленными им правилами. Эти правила могут учитывать порт, используемый входящим трафиком, адрес отправителя пакета и т. п.

Списки доступа должны быть предметом каждодневной заботы администратора. Возвращаясь к предыдущим главам этой книги, можно заметить, что в большинстве протоколов, используемых маршрутизаторами, применяются списки доступа. Списки доступа могут быть двух типов: IP и IPX. В этой главе мы изучим структуру и принципы работы списков доступа IP.

Списки доступа IP

Список доступа (access list) представляет собой набор инструкций, сообщающих маршрутизатору, как он должен обращаться с различными пакетами. Маршрутизаторы Cisco используют списки доступа для управления входящими и исходящими потоками данных. Хотя в этой главе рассматриваются только два типа списков доступа (стандартный IP и расширенный IP), всего на маршрутизаторах Cisco их может быть настроено до 11 видов:

- Стандартный IP
- Расширенный IP
- Код типа протокола
- DECnet
- Appletalk
- MAC (Media Access Control layer)
- Стандартный IPX
- Расширенный IPX
- IPX SAP
- Расширенный MAC
- IPX summary address

Примечание

Управление доступом может быть настроено на маршрутизаторах Cisco практически для всех протоколов. Однако наиболее популярным остается IP.

Конфигурирование списков доступа на маршрутизаторе Cisco осуществляется в два приема. Сначала создается собственно список доступа. Другими словами, формулируются общие правила, определяющие, что маршрутизатор должен делать с определенными пакетами. На следующем этапе список доступа связывается с некоторым интерфейсом.

Таким образом, разные списки доступа могут быть связаны с разными интерфейсами.

Каждый список доступа идентифицируется номером. Этот номер, присвоенный списку доступа, частично определяется его типом. Каждому типу списков доступа назначен диапазон из ста номеров (в предположении, что может быть создано до ста списков доступа каждого типа). В табл. 20.1 приведены диапазоны номеров, выделенные каждому из типов списков доступа.

Таблица 20.1. Номера списков доступа

Тип списка доступа	Диапазон номеров
Стандартный IP	1–99
Расширенный IP	100–199
Код типа протокола	200–299
DECnet	300–399
Appletalk	600–699
MAC (media access control sublayer)	700–799
Стандартный IPX	800–899
Расширенный IPX	900–999
IPX SAP	1000–1099
Расширенный MAC	1100–1199
IPX summary address	1200–1299

Стандартный список доступа IP всегда получает номера от 1 до 99, а расширенный – от 100 до 199. Эти два типа списков доступа мы и рассмотрим в данной главе.

Для конфигурирования списка доступа на маршрутизаторе Cisco используйте команду `access-list` в режиме глобального конфигурирования. Режим глобального конфигурирования используется потому, что на данном этапе лишь определяется набор правил. А вот связывание списка доступа с интерфейсом должно выполняться в режиме конфигурирования интерфейса. Приведенная ниже последовательность команд создает стандартный список доступа IP:

```
Router#configure terminal
Router(config)#access-list 1 permit 198.42.16.1
Router(config)#^Z
```

Примечание

Маршрутизаторы Cisco (в том, что касается списков доступа) придерживаются правила, известного как неявное запрещение. При неявном запрещении считается запрещенным все, что не упомянуто в списке доступа. Следуя этому правилу (на основании только что созданного списка доступа), всем адресам, кроме 186.42.16.1, запрещается доступ к ресурсам, расположенным за данным маршрутизатором.

Команда для конфигурирования стандартного списка доступа IP имеет такой формат:

```
#access-list <access-list number> <action> <source address>
```

Список доступа, созданный в предыдущем примере, устанавливает правило: «Разрешить прохождение трафика, приходящего с адреса 198.42.16.1». Другое возможное значение параметра <action> – это deny. Следующая команда создает стандартный список доступа IP, запрещающий прохождение трафика с адреса 10.36.149.8:

```
Router#configure terminal
Router(config)#access-list 1 deny 10.36.149.8
Router(config)#^Z
```

Стандартные списки доступа прекрасно подходят для небольших сред, в которых разрешение или запрет одного-двух адресов может способствовать повышению безопасности сети. Стандартные списки позволяют создавать общие правила, разрешающие или запрещающие весь трафик с определенного адреса, но такое решение нельзя назвать очень гибким.

Расширенные списки доступа предлагают пользователям Cisco значительно более широкие возможности в назначении правил в зависимости от типа пакетов. Команда создания расширенного списка доступа имеет такой формат:

```
#access-list <access-list number> <action> <protocol> <source address>
<destination address> <port>
```

При конфигурировании расширенных списков доступа пользователю доступно большее количество параметров, чем для стандартных списков. Эти дополнительные параметры позволяют маршрутизаторам Cisco фильтровать трафик, основываясь на адресах отправителя и получателя, типах протоколов и номерах портов. Например, чтобы запретить трафик протокола Telnet (порт 23) с адреса 10.98.12.1 на 10.99.36.5, используем следующие команды:

```
Router#configure terminal
Router(config)#access-list 100 deny IP host 10.98.12.1 host 10.99.36.5 23
Router(config)#^Z
```

Примечание

Протокол Telnet использует TCP-порт 23. В табл. 20.2 перечислены наиболее распространенные сервисы и соответствующие им номера портов.

Таблица 20.2. Общепринятые номера портов

Сервис	Порт
SSH	22
Telnet	23
SMTP	25

Сервис	Порт
HTTP	80
LDAP	389
MS NetMeeting	1024, 1503
HTTPs	443
SOCKS	1080
MS NetShow	1755
MSN Messenger	1863
Mirabilis ICQ	1024
AOL Instant Messenger	5190
AOL ICQ	5190
AOL	5190–5193
Dialpad.com	5354, 7175, 8680–8890, 9000, 9450–9460
pcAnywhere	5631
VNC	5800+, 5900+
Netscape Conference	6498, 6502
Common IRC	6665–6669
RealAudio and Video	7070
VocalTec Internet Conference	22555
MSN Gaming Zone	28800–29000
DirectX Gaming	47624, 2300–2400

В предыдущем примере использовано ключевое слово `host` перед адресами отправителя и получателя. Если необходимо расширить правило, например запретить HTTP-трафик для всей сети, то можно использовать ключевое слово `any`:

```
Router#configure terminal
Router(config)#access-list 147 deny IP any 10.0.0.0 0.255.255.255 any 0.0.0.0
255.255.255.255 80
Router(config)#^Z
```

Данное правило запрещает всем локальным пользователям (сети 10.0.0.0) доступ к любым веб-сайтам (во всем диапазоне 0.0.0.0 – 255.255.255.255 на 80 порту). Однако использование ключевого слова `any` таит в себе и некоторую опасность. Пользователи и сервисы, нуждающиеся в легитимном доступе к некоторым адресам, могут быть заблокированы. Составление списков доступа потребует от вас большой тщательности.

Эти два типа списков доступа, стандартный и расширенный, могут комбинироваться произвольным способом, позволяя фильтровать трафик так, как администратор сочтет наиболее подходящим. Однако со-

здание списка доступа – это лишь половина дела; после этого список должен быть связан с одним или несколькими интерфейсами.

При связывании списка доступа с интерфейсом используется ключевое слово access-group команды ip в режиме конфигурирования интерфейса. Такая команда допускает только один параметр: in или out. Этот параметр определяет, к каким пакетам применяется указанное правило – входящим или исходящим. В приведенной ниже последовательности команд создаются два списка доступа – один стандартный и один расширенный, затем они связываются с интерфейсом маршрутизатора – ethernet 0.

```
Router#configure terminal
Router(config)#access-list 13 permit 128.53.12.1
Router(config)#access-list 108 deny IP host 198.20.13.118 host 128.53.12.1 80
Router(config)#^Z
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-if)# ip address 198.20.13.115
Router(config-if)# ip access-group 108 out
Router(config-if)# ip access group 13 in
```

В этом примере созданы два списка доступа. Первый разрешает входящий трафик с адреса 128.53.12.1, а второй запрещает весь трафик с адреса 198.20.13.118 на 80 порт (HTTP) адреса 128.53.12.1.Таким образом, пользователю с адресом 198.20.13.118 запрещен просмотр вебстраниц по адресу 128.53.12.1:80, и в то же время сервисы этого сайта имеют доступ к локальной сети.

В процессе работы со списками доступа важно постоянно помнить о правиле неявного запрещения. Любой пакет, не соответствующий правилам, установленным в списке доступа, будет отброшен.

Списки доступа – простой и в то же время эффективный способ управления входящим и исходящим трафиком сети. Еще одним, столь же эффективным инструментом, использующим списки доступа, является трансляция сетевых адресов.

NAT

Трансляция сетевых адресов (NAT, Network Address Translation) – это средство, позволяющее шлюзу отображать внешние IP-адреса на множество внутренних IP-адресов. Этот способ обычно используется на шлюзах, обеспечивающих доступ в Интернет. В связи со все возрастающей нехваткой IP-адресов многие сетевые администраторы предпочитают использовать общедоступные IP-адреса. Проблема с использованием таких адресов заключается в том, что они не могут маршрутизироваться в Интернете.

При использовании NAT один выделенный организации IP-адрес присваивается шлюзу, выполняющему преобразование адресов всего вхо-

дящего и исходящего трафика таким образом, что весь исходящий трафик помечается одним внешним адресом, а входящий корректно перенаправляется на соответствующие внутренние адреса.

Трансляция сетевых адресов может служить эффективным инструментом обеспечения безопасности благодаря тому, что в процессе преобразования изменяются адреса отправителя и получателя, что существенно осложняет злоумышленникам доступ к сети через службы IP. Помимо этого, NAT использует в своей работе списки доступа, что также повышает безопасность трансляции. В основе работы NAT, как и у списков доступа, лежит концепция «внутреннего» и «внешнего» трафиков. Чтобы полностью сконфигурировать NAT на маршрутизаторе Cisco, необходимо определить внутренние и внешние характеристики используемых интерфейсов. Обычно на шлюзе хотя бы один интерфейс имеет адрес во внутренней сети и один – в Интернете (или в другой внешней сети). Внутренний интерфейс конфигурируется по «внутренним» правилам NAT, а внешний – по «наружным» правилам. Конфигурирование NAT состоит из четырех основных шагов:

- Определение интерфейсов NAT
- Конфигурирование адресов интерфейсов
- Конфигурирование пула адресов NAT
- Сопоставление списков доступа пулу адресов

Первый шаг конфигурирования NAT состоит в назначении внутренних и внешних интерфейсов. Эта операция выполняется в режиме конфигурирования интерфейса. В следующем примере порт `ethernet 0` конфигурируется в качестве внутреннего, а порт ISDN – в качестве внешнего интерфейса.

```
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-if)#ip nat inside
Router(config-if)#interface bri 0
Router(config-if)#ip nat outside
Router(config-if)#^Z
```

Второй шаг конфигурирования NAT заключается в назначении выбранным интерфейсам соответствующих IP-адресов. Другими словами, внутренний интерфейс должен получить адрес во внутренней сети, а внешний – во внешней.

Примечание

Обычно адрес для назначения внешнему интерфейсу предоставляется интернет-провайдером.

Используйте команды, подобные приведенным ниже, для установки внутреннего и внешнего адресов:

```
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-if)#ip address 198.65.1.1 255.255.0.0
Router(config-if)#interface bri 0
Router(config-if)#ip address 186.91.108.1 255.255.0.0
Router(config-if)#^Z
```

Здесь интерфейсы маршрутизатора конфигурируются для работы с NAT. Однако маршрутизатор еще не знает, что ему делать с этой информацией. Ему необходимо указать пул адресов, подлежащих преобразованию. То есть маршрутизатору надо объяснить, по каким адресам он должен отправлять прибывающие данные.

Формирование адресного пула NAT выполняется в режиме глобального конфигурирования. Команда создания адресного пула NAT имеет такой формат:

```
#ip nat pool <name> <address range> netmask <subnet mask>
```

Приведенный ниже код иллюстрирует создание адресного пула NAT:

```
Router#configure terminal
Router(config)#ip nat pool my_pool 198.65.1.1 198.65.254.254 netmask 255.255.0.0
Router(config)#^Z
```

Здесь использована команда `nat_pool` для создания адресного пула с именем `my_pool`, который будет использоваться при трансляции адресов. Данный пул использует адреса от 198.65.1.1 до 198.65.254.254. Наличие у адресных пулов имен позволяет назначать разные пулы разным интерфейсам.

Прежде чем использовать пул `my_pool` для трансляции адресов, следует создать список доступа, содержащий правила обработки входящих в этот пул адресов. После этого при конфигурировании NAT можно будет указать, что адреса пула используются согласно правилам, установленным списком доступа. В данном примере список доступа просто разрешает трафик для адресов нашего пула. Список доступа создается командой следующего вида:

```
Router#configure terminal
Router(config)#access-list 1 permit 198.65.0.0 0.0.255.255
Router(config)#^Z
```

Последнее, что надо сделать, – это сопоставить список доступа адресному пулу NAT. Эта операция выполняется такой командой:

```
Router#configure terminal
Router(config)#ip nat inside source list 1 pool my_pool
Router(config)#^Z
```

Команда `nat inside source` определяет, что весь входящий трафик, соответствующий правилам списка доступа `1`, может транслироваться в

адресный пул `my_pool`. После того как все правила установлены, маршрутизатор готов к работе. Любой входящий пакет, запрещенный списком доступа, не обрабатывается NAT, а отбрасывается.

Резюме

Комбинация NAT и списков доступа IP образует простой, но эффективный способ обеспечения безопасности, необходимый всем пользователям Cisco. Эти два инструмента позволяют создать правила обработки трафика, подходящие для большинства деловых сетей. Необходимо помнить о следующих факторах обеспечения безопасности:

- Списки доступа представляют собой основанные на правилах командные структуры, определяющие, как маршрутизатор должен поступать с пакетами.
- В списках доступа используется правило неявного запрета.
- Любой пакет, не удовлетворяющий правилам списка доступа, отбрасывается.
- Трансляция сетевых адресов NAT используется для замены адресов отправителя и получателя пакета.
- В процессе трансляции NAT использует правила, изложенные в списках доступа.

Последняя глава этой книги посвящена основам маршрутизации на коммутаторе Cisco Catalyst и PNNI.

Вопросы и ответы

Вопрос Обязательно ли устанавливать NAT на шлюзе?

Ответ Нет, не существует правил, утверждающих, что NAT непременно должен быть установлен на пограничном маршрутизаторе или каком-либо внутреннем маршрутизаторе, соединяющем несколько сетей. Однако NAT часто применяется для контроля над трафиком Интернета.

Тест

Вопросы

1. Какой диапазон номеров выделен для расширенных списков доступа IP?
2. Списки доступа какого типа позволяют блокировать отдельные порты?
3. Как конфигурируется NAT: глобально или для отдельного интерфейса?

Ответы

1. 100–199.

2. Расширенный IP.
3. В обоих режимах. Адресный пул создается в глобальном режиме, а внешний и внутренний интерфейсы определяются в режиме конфигурирования интерфейса IOS.

Упражнения

1. Сконфигурируйте расширенный список доступа, блокирующий входящий трафик SMTP.

Решение

```
Router#configure terminal
Router(config)#access-list 100 deny IP any 0.0.0.0 255.255.255.255 any
0.0.0.0 255.255.255.255 25
Router(config)#^Z
```

2. Сконфигурируйте на маршрутизаторе NAT для внутренней сети 10.0.0.0 и внешнего интернет-соединения с адресом 115.68.43.1. Внутренний интерфейс – ethernet 0, внешний – serial 0; адрес внутреннего интерфейса – 10.101.23.1.

Решение

```
Router#configure terminal
Router(config)#interface ethernet 0
Router(config-if)#ip address 10.101.23.1 255.0.0.0
Router(config-if)#ip nat inside
Router(config-if)#interface serial 0
Router(config-if)#ip nat outside
Router(config-if)#ip address 115.68.43.1 255.0.0.0
Router(config-if)#^Z
Router#configure terminal
Router(config)#ip nat pool exercise 10.0.0.0 10.254.254.254 netmask
255.0.0.0
Router(config)#access-list 1 permit 10.0.0.0 0.255.255.255
Router(config)#ip nat inside source list 1 pool exercise
Router(config)#^Z
```

21

Основы маршрутизации на коммутаторе Cisco Catalyst и PNNI

В последней главе этой книги рассмотрена еще одна технология маршрутизации Cisco, о которой мы еще не упоминали. Высокотехнологичные коммутаторы Cisco Catalyst могут помимо своей основной задачи выполнять еще и задачи маршрутизации. В частности, они могут работать с весьма полезным протоколом маршрутизации PNNI (Private Network to Network Interface – интерфейс «частная сеть–сеть»). Поскольку у вас, вероятно, нет доступа к коммутатору Cisco Catalyst (вряд ли кто-нибудь станет приобретать коммутатор ради изучения), эта глава будет по возможности короткой. Основная ее цель – познакомить вас с протоколами маршрутизации, работающими на устройствах, не являющихся маршрутизаторами, в данном случае – на коммутаторах Cisco.

В этой главе вы познакомитесь с протоколом PNNI и командами для его конфигурирования на коммутаторе Cisco. Мы рассмотрим следующие темы:

- Архитектура ATM (Asynchroronous Transfer Mode)
- Протокол сигнализации UNI
- Иерархия PNNI
- Протокол маршрутизации PNNI
- PNNI и QOS
- Протокол сигнализации PNNI
- Специальный механизм блокирования PNNI (crankback)
- Конфигурирование PNNI

Протокол PNNI используется при передаче данных между коммутируемыми сетями ATM (Asynchronous Transfer Mode, асинхронный ре-

жим передачи). Этот протокол выполняет маршрутизацию входящих и исходящих данных для нескольких ATM-групп как локально, так и в глобальном масштабе.

Собственно термином PNNI обозначается логический интерфейс, предназначенный для соединения и взаимодействия множества сетей ATM. Спецификация PNNI содержит рекомендации и правила маршрутизации данных между большими ATM-сетями.

PNNI создан организацией ATM Forum в 1996 году с целью эффективной маршрутизации данных между сетями ATM. Когда в ATM Forum начинали работу над стандартом, получившим впоследствии название PNNI, то постарались положить в основу спецификаций уже имеющиеся технологии. Такой подход позволил членам ATM Forum избежать многолетних затрат на создание нежизнеспособных решений, равно как и дублирования уже выполненных разработок.

Примечание

Организация ATM Forum (основанная в 1991 году) представляет собой альянс корпораций, совместно работающих над внедрением технологии ATM в бизнес. Корпорации-члены ATM Forum устанавливают и применяют стандарты ATM-коммутации и способствуют внедрению этой технологии.

Фактически PNNI выполняет две основные функции: протокола маршрутизации и протокола сигнализации. Протокол маршрутизации PNNI перемещает данные от одного ATM-кластера (группы) к другому. Для эффективной маршрутизации очень быстро перемещающихся ATM-ячеек в ATM Forum было решено разрабатывать протокол маршрутизации PNNI на основе других протоколов состояния канала.

Примечание

В основу протокола маршрутизации PNNI положены протоколы состояния канала. То есть узел сети с помощью пакетов обновления может передать другим узлам сведения о состоянии своих каналов. В целом, протоколы состояния канала позволяют выполнять маршрутизацию быстро и эффективно.

PNNI выполняет также функции протокола сигнализации. Сигнальный протокол применяется для создания и удаления соединений между коммутаторами. Эта функция является критичной для функционирования ATM-сетей. Прежде чем PNNI сможет начать работу в ATM-среде, необходимо обработать запросы на соединения между двумя (или более) коммутаторами.

Сигнальный протокол PNNI основан на еще одной спецификации ATM Forum, UNI (user-to-network interface – интерфейс «пользователь–сеть»). Сигнальный протокол UNI был разработан членами ATM Forum в качестве быстрого и гибкого протокола сигнализации в частных и общедоступных сетях. Путем небольших изменений UNI был превращен в сигнальный протокол PNNI.

Эта глава познакомит вас с технологией, лежащей в основе PNNI и двух его протоколов. Однако прежде чем мы начнем изучение PNNI, нам предстоит познакомиться с основами ATM-сетей.

Архитектура ATM

Технология ATM (Asynchronous Transfer Mode – режим асинхронной передачи) была разработана с целью обеспечения высокоскоростного обмена данными, голосовыми сообщениями и видео по каналам передачи данных общего пользования. Сеть ATM, в отличие от многих знакомых нам сетей, является коммутируемой. То есть для надежной передачи данных между коммутаторами с помощью протокола с установлением соединений создается канал. На рис. 21.1 показана типичная ATM-архитектура.

Примечание

В ATM поддерживается передача данных без установления соединения, но такой режим используется редко. ATM позволяет передавать данные по сетям общего пользования со скоростью до 1 Гбит/с.

Большинство используемых в настоящее время сетевых технологий работают без установления соединений (например, широко распространенный IP). Передача данных без установления соединения неизбежно порождает проблему: нет никакой гарантии, что данные достигнут адресата. Однако ATM, устанавливая соединение, создает на время сеанса канал между маршрутизаторами. По завершении сессии (когда передача данных между коммутаторами закончена) ATM удаляет использованный канал. Это позволяет следующей сессии исполь-

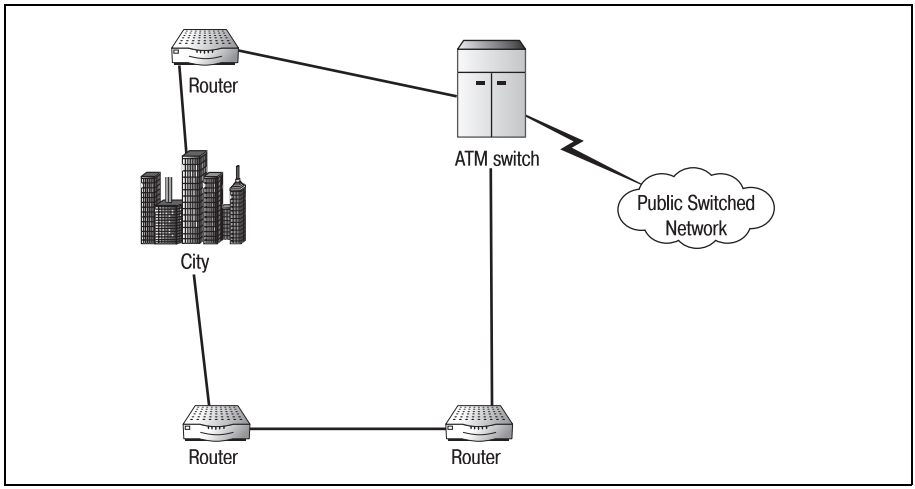

***Рис. 21.1.** Сеть ATM*

зовать один или оба порта для другого соединения. На рис. 21.2 показан процесс создания и удаления ATM-соединения.

Разумеется, передача данных с установлением соединения применяется не только в ATM. Ближайший родственник IP, протокол TCP, также является протоколом, ориентированным на соединение. Несомненно, ATM может служить прекрасным примером передачи данных с установлением соединения.

Ключом к высокоскоростной передаче данных в ATM служит используемый формат данных. ATM, в отличие от большинства протоколов, не передает пакетов переменной длины. Такие протоколы, как IS-IS (глава 19), передают пакеты, длина которых зависит от передаваемых данных. Поэтому устройство-приемник вынуждено просматривать заголовок пакета, определять размер данных, а затем обрабатывать пакет. При использовании пакетов переменной длины время обработки меняется от пакета к пакету. Чем больше пакет, тем дольше он обрабатывается.

В противоположность этому, ATM передает пакеты в ячейках фиксированной длины. Ячейки ATM независимо от количества передаваемых данных имеют длину 53 байта. Как показано на рис. 21.3, ячейки ATM состоят из заголовка длиной 5 байт и «полезной нагрузки», или сегмента данных, длиной 48 байт.

Примечание

Несмотря на то что ячейки ATM имеют фиксированную длину (что исключает потребность в поле длины в заголовке), заголовок все же необходим. В заголовке ATM-ячейки указываются отправитель и получатель сегмента данных.

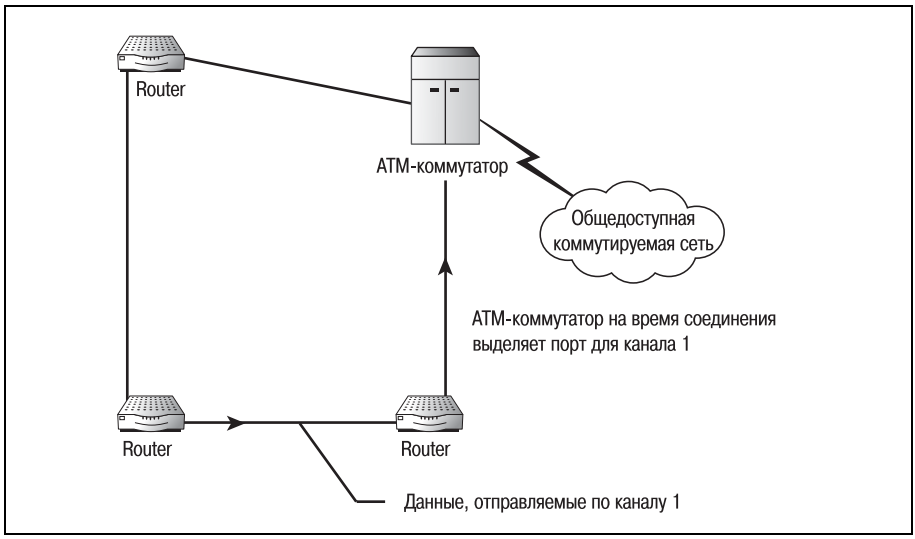

Рис. 21.2. *Соединение ATM*

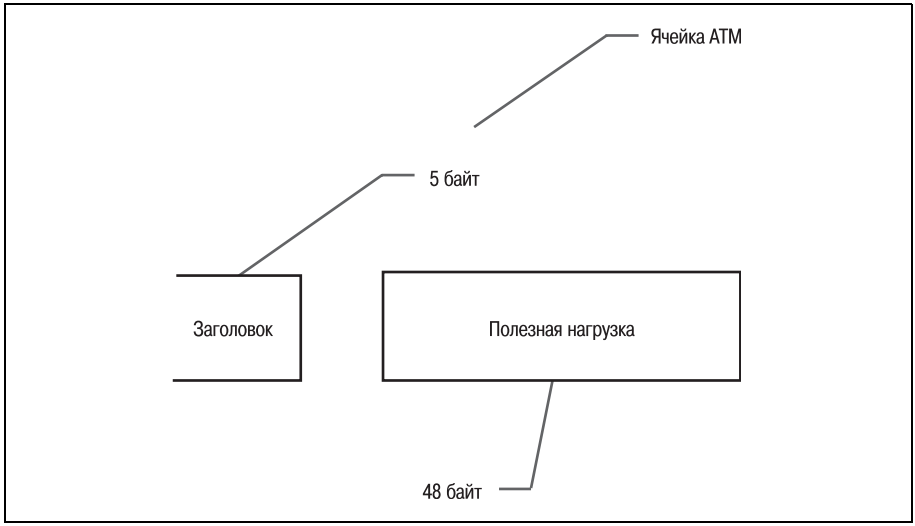

Рис. 21.3. *Ячейка ATM*

Фиксированная длина ячеек ATM делает их идеально подходящими для доставки аудио- и видеоданных. Благодаря тому что у принимающего ATM-устройства не возникает случайных колебаний нагрузки на процессор (зависящей от размера пакета), данные принимаются равномерно, с постоянной скоростью. Любое замедление в преобразовании сигнала скорее всего будет обусловлено увеличением частоты дискретизации и пройдет незамеченным.

Еще одной причиной, послужившей успеху ATM, стало то, что этот протокол – асинхронный. Асинхронность означает, что ATM-коммутаторы могут в одно и то же время выполнять прием и передачу данных. Таким образом, два коммутатора могут одновременно передавать данные друг другу, вдвое сокращая общее время передачи (и использования сети).

Асинхронность протокола ATM обусловлена фиксированным форматом его ячеек. Благодаря тому что они имеют постоянную длину 53 байта, любому коммутатору заранее известно время, необходимое для их приема. Поэтому в перерывах между отправкой ячеек коммутатор может выполнять прием любых данных, приходящих от коммутатора на другом конце канала.

Прежде чем углубляться в вопросы коммутации и сигнализации ATM (и, следовательно, PNNI), посмотрим, как организована типичная ATM-сеть. Так как PNNI не только разрабатывался специально для ATM-сетей, но и основывался на уже существующем протоколе (UNI версии 3), вам следует поближе познакомиться с организацией коммутируемых ATM-сетей. Понимание архитектуры ATM-сети поможет изучить принципы, на которых основан протокол PNNI.

Организация ATM-сети

Существующие ATM-сети могут быть разделены на два типа: общего пользования и частные. Первые находятся в публичном доступе. То есть, подобно Интернету, общедоступная сеть представляет собой объединение поддерживаемых телекоммуникационными компаниями сетей общего пользования. В противоположность им, частные ATM-сети существуют лишь в пределах одного предприятия.

Примечание

Учтите, что ATM-сети общего пользования состоят в основном из ATM-коммутаторов (называемых коммутаторами общего доступа). Количество конечных систем (ES) в этих сетях невелико.

Разумеется, как и большинство правил, касающихся сетей, такое определение не всегда справедливо. На самом деле у вас может быть две частных ATM-сети, соединенных при помощи сети общего пользования. В этом случае каждая из сетей относится к своему типу (частных или общедоступных сетей) и их нельзя рассматривать как одно целое. ATM-сети сравнительно просты в организации, но не в том, что касается технологии. Эти сети базируются на самой передовой и сложной технологии коммутации пакетов.

Каждое устройство ATM-сети представляет собой либо ATM-коммутатор, либо конечную систему. ATM-коммутатор – это устройство, обеспечивающее соединение с сетью. То есть это то устройство, с помощью которого осуществляется передача данных в сеть и обратно. На заре развития ATM коммутатор обычно был конечной точкой присутствия корпоративной сети.

Конечная система ATM – это любое устройство ATM-сети, не являющееся коммутатором. Следовательно, все ПК, маршрутизаторы и принтеры считаются конечными системами. В силу того что конечные системы не обладают способностью к коммутации, они отправляют все данные ATM-коммутатору для дальнейшей пересылки по сети.

На первом этапе развития ATM такая классификация устройств делала процесс проектирования сети достаточно легким. Использование ATM началось с сетей общего пользования. Локальная сеть компании соединялась с ATM-коммутатором. Данные с этого коммутатора пересылались по коммутируемой сети общего пользования (обычно это была общедоступная телефонная компания). По этой сети информация доставлялась на коммутатор сети назначения. На рис. 21.4 показана общедоступная ATM-сеть.

Для первых общедоступных ATM-сетей был разработан свой набор протоколов: UNI (User-to-Network Interface – интерфейс «пользователь-сеть») и NNI (Network-to-Network Interface – интерфейс «сеть-сеть»). Эти протоколы были созданы исключительно для целей достав-

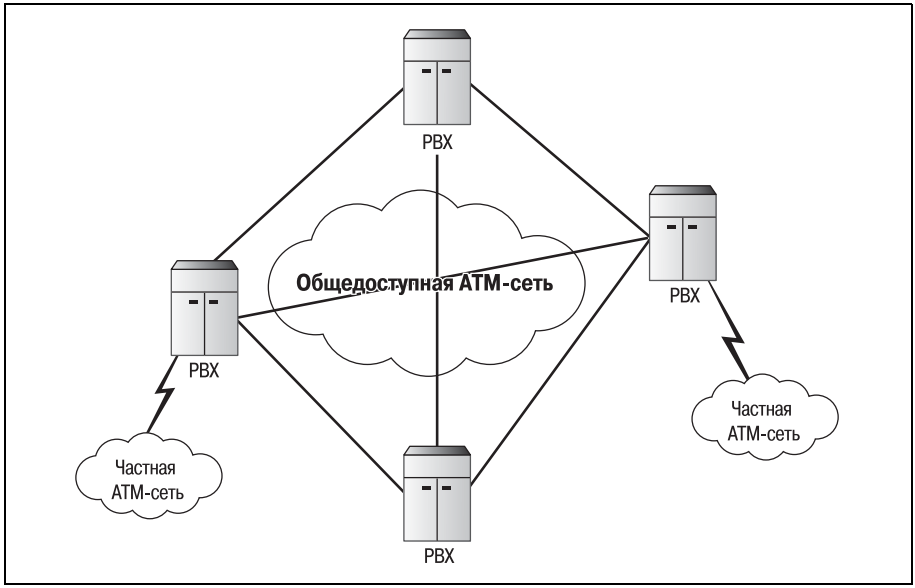

Рис. 21.4. *Сеть ATM общего пользования*

ки общедоступным коммутаторам сигналов от других коммутаторов о наличии у них данных для передачи. Как показано на рис. 21.5, для двух типов ATM-сетей (общедоступных и частных) имеются два типа протоколов: UNI и NNI.

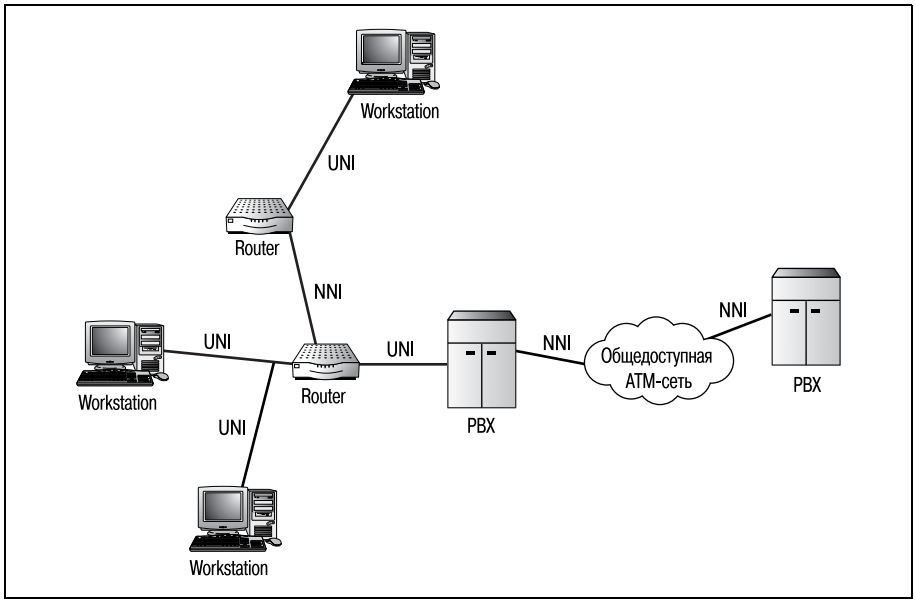

Рис. 21.5. *Применение протоколов UNI и NNI*

Протокол UNI был разработан с целью позволить ATM-коммутатору (или конечной системе) из частной сети отправлять сигнал произвольному общедоступному коммутатору о наличии данных для передачи. После этого UNI должен создать канал между двумя коммутаторами. По окончании сеанса связи UNI должен удалить канал. Частная версия этого протокола (под названием P-UNI) может использоваться для соединения конечных систем с частными ATM-коммутаторами.

С ростом сетей ATM возникла необходимость в разработке протокола, справляющегося с нагрузкой, создаваемой процессами межкоммутаторной сигнализации. Для этой цели и были разработаны протокол NNI и его аналог для частных сетей PNNI. Протокол NNI был создан по образцу UNI, чтобы обеспечить соблюдение соответствующих коммуникационных стандартов.

Протокол сигнализации UNI

UNI (User to Network Interface) – это протокол сигнализации, который разрешает устанавливать связь между конечными системами и частными коммутаторами или между частными и общедоступными коммутаторами. Протоколы сигнализации отправляют коммутаторам специальным образом отформатированные сообщения. Принимающий коммутатор в ответ предоставляет открытую линию (канал) для связи между устройствами.

UNI может подавать сигналы для открытия в среде ATM двух типов каналов: «точка-точка» и «точка-группа». Эти два типа соединений изображены на рис. 21.6 и 21.7 соответственно.

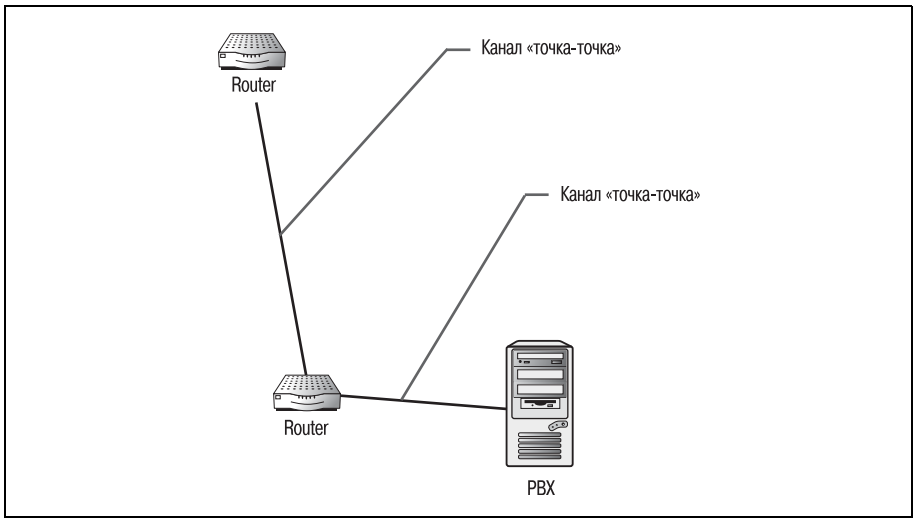

Рис. 21.6. *Канал «точка-точка»*

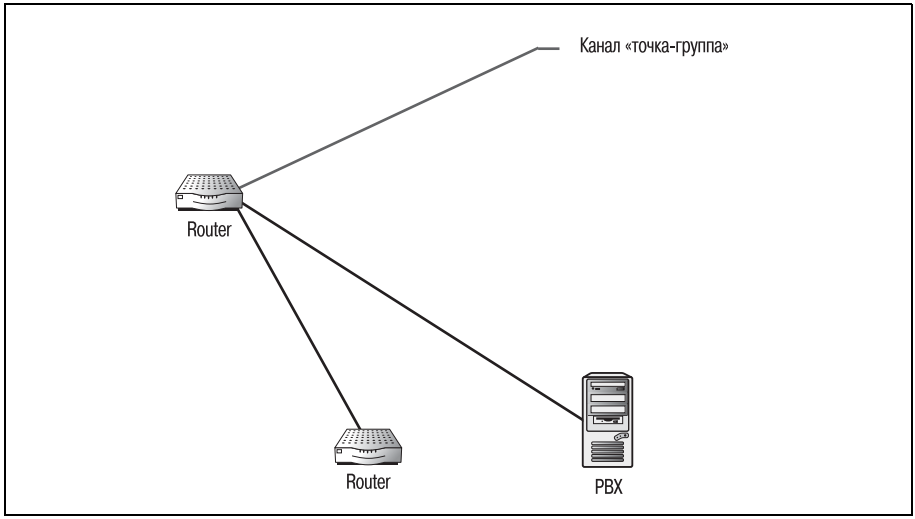

Рис. 21.7. *Канал «точка-группа»*

Когда UNI открывает канал «точка-точка», одна система связывается с другой и запрашивает соединение. После того как принимающая система примет сигнал, между двумя системами открывается канал связи. Действуя подобно прямому кабельному соединению, канал «точка-точка» обеспечивает возможность взаимодействия двух систем (точек).

Основным преимуществом канала «точка-точка» является использование асинхронных портов. ATM-соединение «точка-точка» может обеспечивать двунаправленную связь между системами. В сетевой среде такое свойство очень желательно, так как оно позволяет обеим системам прерывать сессии.

Соединение типа «точка-группа» устанавливается, когда системе необходимо открыть канал с несколькими системами. Одной из причин открытия каналов «точка-группа» может быть многоадресная рассылка.

Главный недостаток ATM-соединений типа «точка-группа» заключается в том, что они не асинхронны. Такой канал обеспечивает связь только в одном направлении (от точки ко многим точкам). Эта проблема не связана с протоколом UNI; поля ячейки ATM не предусматривают записи, необходимые для открытия асинхронного канала «точка-группа» (называемого также multipoint-to-multipoint).

Структура ячейки ATM и протокол сигнализации

Стандартная ячейка ATM имеет 5-байтный заголовок. Этот заголовок содержит информацию, необходимую любой системе для определения того, что это за ячейка, кому она предназначена и кто ее отправил. Поля заголовка зависят от типа ячейки: например, заголовок ячейки

UNI содержит поля, отличные от полей заголовка ячейки NNI. Заголовок ячейки UNI содержит следующие поля (рис. 21.8):

- Общее управление потоком (4 бита) – обычно не используется.
- Идентификатор виртуального пути (8 бит) – идентифицирует следующий переход на пути к адресату ячейки. (Ячейке может потребоваться пройти через множество коммутаторов, прежде чем она достигнет своего адресата.)
- Идентификатор виртуального канала (16 бит) – помогает текущему коммутатору идентифицировать следующий коммутатор для получения ячейки.
- Тип полезной нагрузки (3 бита) – указывает содержимое ячейки.

  Бит A – если этот бит присутствует, то полезная нагрузка содержит данные.

  Бит B – указывает на перегрузку.

  Бит C – маркер, используемый для подачи сигнала о последней ячейке последовательности.
- Приоритет при потере ячейки (1 бит) – используется коммутатором для отбрасывания ячеек, которые слишком долго добирались до места назначения.
- Контрольная сумма заголовка – используется для проверки целостности заголовка.

По мере перемещения ячейки по сети каждый коммутатор изменяет идентификаторы виртуального пути и канала, чтобы обеспечить успеш-

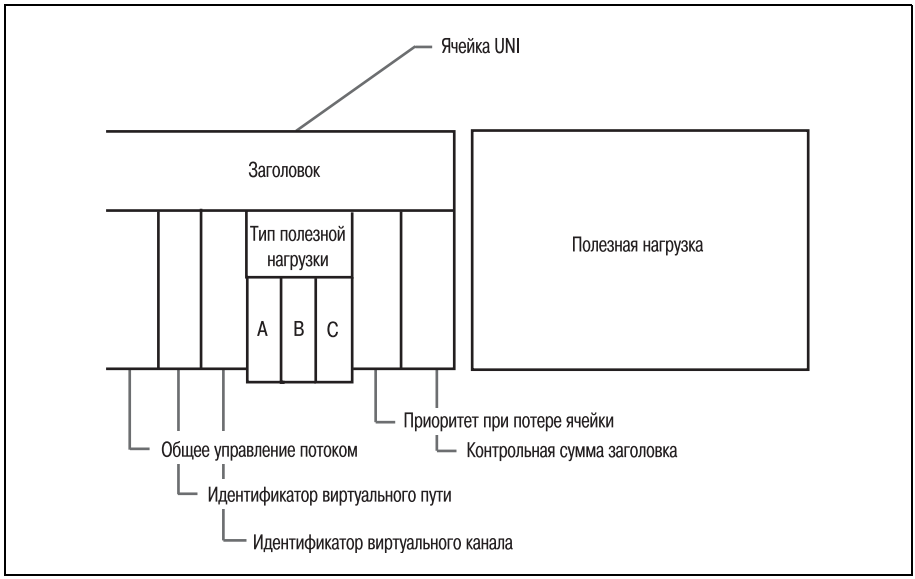

Рис. 21.8. *Поля заголовка ячейки UNI*

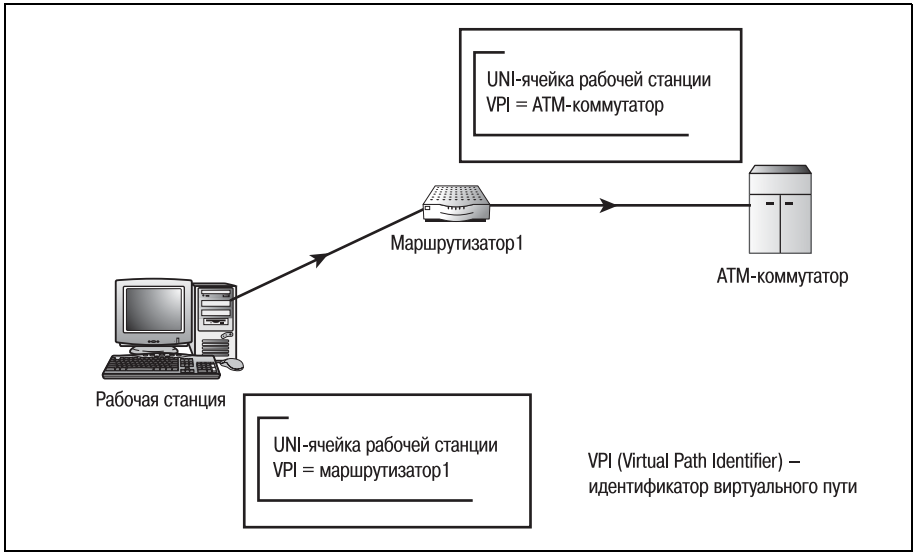

Рис. 21.9. *Маршрутизация ячейки UNI*

ную маршрутизацию ячейки. Когда ячейка проходит через коммутатор, то пройденный ею путь записывается, как показано на рис. 21.9.

Когда ячейка достигает конечного адресата, последний коммутатор принимает решение о том, принять или отвергнуть ячейку. Если коммутатор принимает ячейку, то путь, пройденный ячейкой, превращается в канал. Канал UNI изображен на рис. 21.10.

UNI – это хороший протокол для перемещения информации по сети ATM. Но когда требуется открытие канала между двумя частными

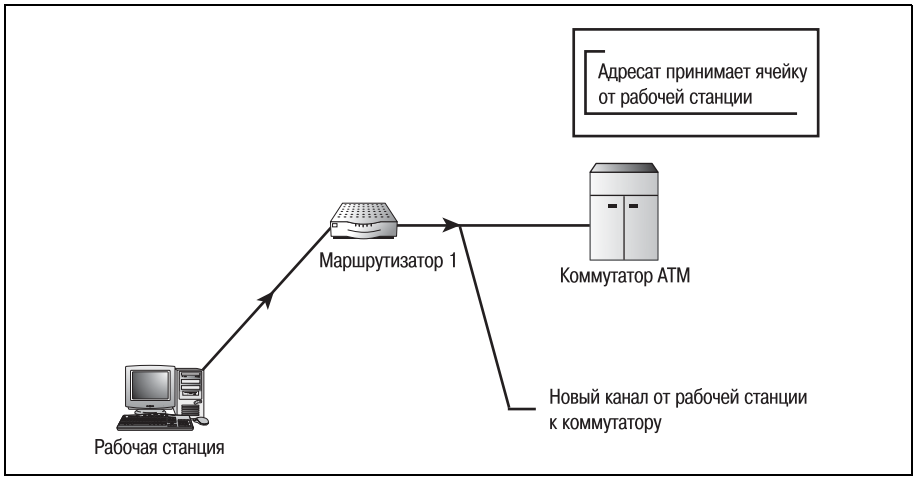

Рис. 21.10. *Канал UNI*

коммутаторами, необходим другой протокол – протокол PNNI (Private Network-to-Network Interface, интерфейс «частная сеть–сеть»).

UNI не имел таких возможностей маршрутизации и сигнализации, которые требовались для выполнения комплексной задачи создания и удаления каналов, не мог он и объявлять маршруты. Когда члены ATM Forum решили создать протокол, обеспечивающий отсутствующие у UNI возможности маршрутизации и сигнализации, они взяли за основу UNI.

Разрабатывая новый Private Network-to-Network Interface на базе UNI, они были уверены, что создают протокол на хорошей основе. В следующих разделах будет описана работа протокола PNNI и объяснено, почему он был смоделирован именно таким образом.

Иерархия PNNI

Сети PNNI иерархичны: некоторые системы или группы систем имеют приоритет перед остальными при создании оптимальных путей. В этом разделе будет рассказано об иерархической структуре PNNI и о том, как она согласуется с архитектурой ATM.

Если бы протокол PNNI не использовал для маршрутизации иерархию, то сети были бы плоскими. Все группы (кластеры) систем были бы уравнены. С точки зрения маршрутизации плоская сеть может иметь несколько проблем.

Первое, чем плоская среда маршрутизации может повредить сети, заключается в увеличении объема памяти и мощности процессора, необходимых для принятия решений о маршрутизации. Если все системы сети находятся на одном уровне, то нет возможности упростить маршрутизацию, разбив их на более мелкие группы. Маршрутизатор рассчитывает пути при помощи алгоритма маршрутизации, подставляя каждую систему в качестве переменной. Отслеживание таких переменных занимает очень большой объем памяти. При вычислении путей обработка чисел, полученных из всех этих систем, требует еще большей загрузки процессора.

Вторая проблема плоской среды маршрутизации заключается в потере скорости. Чем больше переменных маршрутизатор должен обрабатывать при принятии решения, тем больше времени это занимает. В плоской сети каждая система рассматривается маршрутизатором как возможная переменная. В больших сетях процедура маршрутизации данных требует использования настолько большого числа переменных, что сеть работает невыносимо медленно.

PNNI защищается от проблем, связанных с плоскими сетями, благодаря использованию иерархической схемы сети. PNNI помещает каждую систему в группу, затем группы объединяются в более крупные группы, то есть иерархическая структура сети может иметь неограни-

ченную глубину. Наименьшая группа, являющаяся основой для всего остального в иерархии, называется одноранговой группой.

Одноранговые группы PNNI

Сети PNNI разделены на одноранговые группы (peer group, PG). Каждая одноранговая группа имеет свой адрес в сети. Этот адрес (называемый групповым адресом, или PG-номером) является уникальным и совместно используется всеми членами одноранговой группы.

То, как именно конечные системы разбиваются на одноранговые группы, зависит от адреса коммутатора. PG определяется общим адресом ее членов, а адрес каждого члена порождается коммутатором, к которому он подключен, поэтому разбиение на одноранговые группы зависит от адресации коммутаторов.

Одноранговая группа PNNI создается исключительно в целях обеспечения маршрутизации. Внутри одноранговой группы конечные системы и их коммутаторы передают данные друг другу и другим членам группы. Это ограничивает объем трафика, передаваемого от коммутатора к коммутатору. Так что и самый младший элемент иерархии имеет свои преимущества.

Для обеспечения маршрутизации PNNI члены одноранговой группы могут иметь информацию о маршрутах только для систем, входящих в их группу. Системы некоторой определенной PG никогда не знают точного состава никакой другой одноранговой группы. Создавая группы нижнего уровня, PNNI получает возможность контролировать объем данных, которые должны обрабатывать пограничные узлы.

Примечание

Пограничный узел – это ATM-коммутатор, соединяющий одноранговые группы. Посредством использования пограничного узла информация может пересылаться от одной одноранговой группы к другой.

NSAP-адресация PNNI

Большинство частных ATM-сетей используют адресный формат NSAP (Network Dervice Access Point, точка доступа к сетевому сервису; см. главу 19).

Примечание

Другие протоколы, такие как IS-IS, также используют адресный формат NSAP или его модифицированную версию.

Двадцатибайтный NSAP-адрес ATM состоит из нескольких полей, определяемых сетью, и MAC-адреса устройства. NSAP-адрес для PNNI состоит из следующих полей (рис. 21.11):

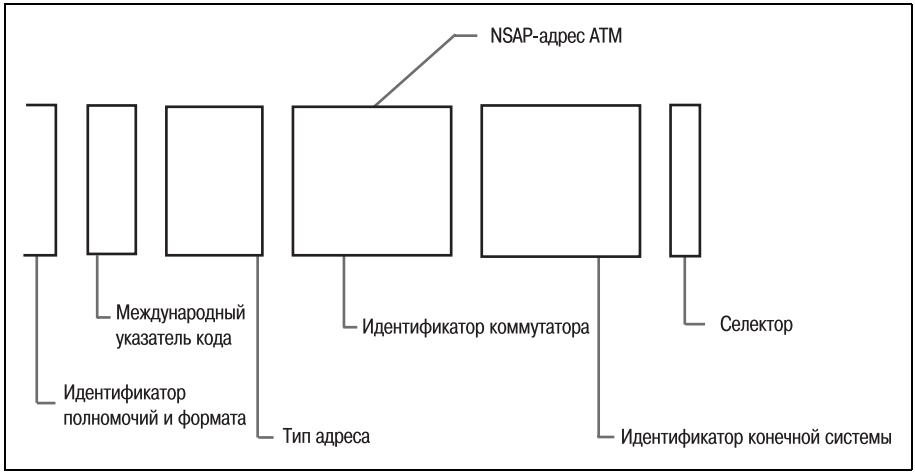

Рис. 21.11. *NSAP-адрес ATM-сети*

- AFI (Authority and Format Identifier) – идентификатор полномочий и формата. Существуют различные форматы адресов, определенные форумом ATM.[1] Данное поле имеет предопределенное значение в зависимости от формата.
- ICD (International Code Designator) – международный указатель кода. Определяет организацию, присваивающую адрес (используется для обеспечения международной уникальности).
- Тип адреса – указывает принадлежность адреса к ICD, NSAP или DCC (Data Country Code – код страны данных).
- Идентификатор коммутатора – MAC-адрес коммутатора.
- Идентификатор конечной системы – MAC-адрес конечной системы.
- Селектор – не используется.

На рис. 21.12 показан заполненный адрес ATM-коммутатора (в примере использованы AFI и ICD, относящиеся к Cisco Systems).[2]

Все системы, входящие в одноранговую группу, имеют один и тот же 12-байтный идентификатор. Этот идентификатор на самом деле представляет собой первые 12 байт 13-байтного ATM-адреса коммутатора (тринадцатый байт – это идентификатор коммутатора). Так как первые 12 байт ATM-адресов всех конечных систем одноранговой группы совпадают, то эти байты и являются номером PG.

---

[1] Для частных сетей определены форматы DCC AESA (AFI=39), ICD AESA (AFI=47), E.164 AESA (AFI=45). – *Примеч. науч. ред.*

[2] Большинство производителей оборудования ATM использует формат ICD AESA. – *Примеч. науч. ред.*

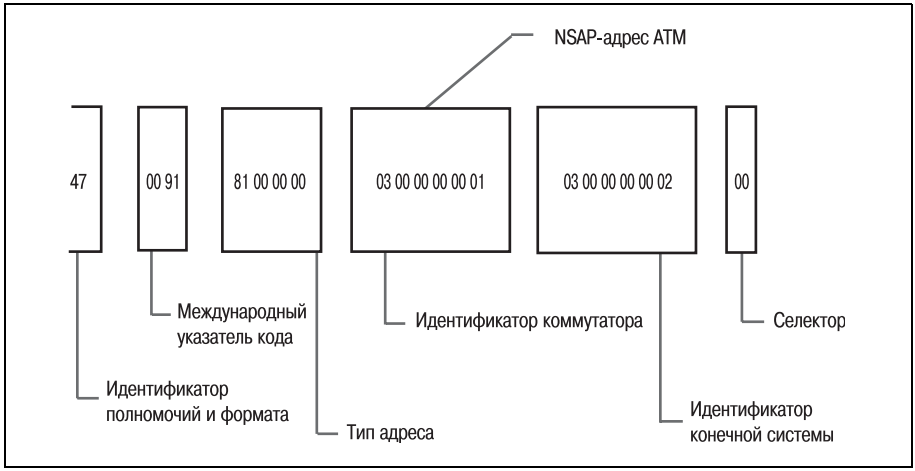

Рис. 21.12. *Адрес ATM-коммутатора*

Одна особенность формирования адресов ATM-коммутаторов связана с самим ATM-коммутатором. Согласно логике, использующейся для образования номера PG (MAC-адрес ATM-маршрутизатора – это идентификатор конечной системы для всех остальных устройств той же одноранговой группы), идентификатор конечной системы и идентификатор коммутатора (как поля адреса) идентичны для ATM-коммутатора внутри одноранговой группы.

Когда к коммутатору подключается новая конечная система, она принимает в качестве своего идентификатора маршрутизатора идентификатор конечной системы коммутатора. MAC-адрес конечной системы используется для заполнения поля ее собственного идентификатора конечной системы.

Используя для формирования ATM-адресов описанную выше логику, PNNI может динамически создавать одноранговые группы. Когда конечная система присоединяется к коммутатору, ей динамически присваивается NSAP-адрес и она принимается в одноранговую группу. Одноранговые группы являются основой иерархической модели PNNI. На рис. 21.13 изображены три одноранговые группы ATM.

Как показано на рис. 21.13, новые одноранговые группы идентифицируются PG-номерами. Так как первые 12 байт ATM-адреса всех узлов одноранговой группы совпадают, то для их представления можно использовать букву (в нашем примере это буква «A»). Число, стоящее после точки, – это тринадцатый байт ATM-адреса (ATM-коммутатора). Это число однозначно определяет коммутатор и завершает представление номера одноранговой группы. Затем к номеру группу присоединяется MAC-адрес каждого узла, идентифицирующий конкретную конечную систему.

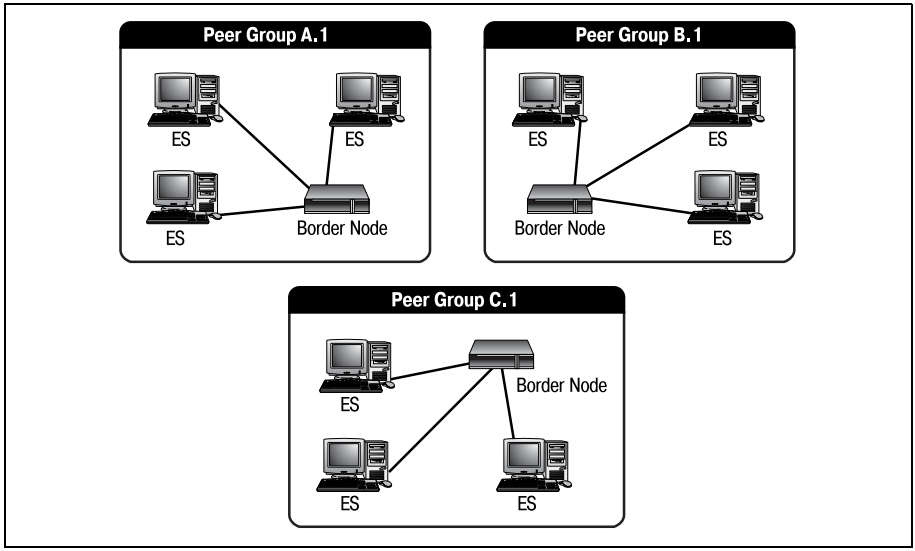

***Рис. 21.13**. Три одноранговые группы ATM*

Прежде чем можно будет осуществлять маршрутизацию между новыми одноранговыми группами, необходимо выполнить еще один шаг. Каждому коммутатору одноранговой группы необходимо назначить PNNI-уровень. Этот уровень определяет положение группы в иерархической структуре PNNI. Группа с минимальным значением уровня PNNI называется доминирующей. Доминирующая одноранговая группа имеет более высокий приоритет при маршрутизации, чем другие PG. Уровень PNNI может быть любым числовым значением в диапазоне от 0 до 104.

Одноранговая группа, для которой вы установите номер 104, будет иметь очень низкий приоритет маршрутизации по сравнению с PG уровня 0. Назначая уровни, вы можете беспрерывно объединять одноранговые группы во все большие и большие объекты. Например, на рис. 21.13 одноранговые группы A.1 и B.1 можно рассматривать как одну большую группу. Одноранговая группа с наименьшим уровнем обрабатывает решения о маршрутизации в этой большой группе.

Все время объединяя одноранговые группы в более крупные, PNNI избавляется от проблем плоской среды маршрутизации. Хотя члены одноранговой группы обладают информацией о топологии только своей группы, коммутаторы нижнего уровня имеют знания о каждой одноранговой группе, находящейся над ними в иерархической структуре.

Примечание

Топология одноранговой группы определяется положением конечных систем по отношению к коммутатору.

Чтобы свести к минимуму трафик, маршрутизируемый от коммутатора к коммутатору, каждая одноранговая группа действует как самодостаточная сеть. Если конечная система одноранговой группы A.1 (рис. 21.13) хочет отправить данные конечной системе одноранговой группы B.1, она должна выполнить следующие шаги. Коммутатор A.1, не имеющий прямых сведений о конечных системах группы B.1, определяет, что ячейка не предназначена для локальной одноранговой группы. Тогда коммутатор пересылает пакет коммутатору для одноранговой группы B.1 (и в дело вступают протоколы маршрутизации).

По мере увеличения размера группы объем памяти, необходимой для отслеживания топологии и обработки запросов на маршрутизацию, также увеличивается. Поэтому, для того чтобы обойти потенциальное ограничение архитектуры PNNI, необходимо воспользоваться иерархичностью.

Примечание

Одним из ключей к успеху сетевой структуры PNNI является ее способность масштабирования сетей практически любого размера. Иерархическая модель PNNI, использующая номера и уровни одноранговых групп, может быть адаптирована почти для любой сети.

За счет применения уровней PNNI группы A.1 и B.1 могут рассматриваться как одна более высокоуровневая одноранговая группа. Коммутатор с минимальным уровнем занимается маршрутизацией для группы высокого уровня. На рис. 21.14 изображена сеть PNNI, одноранговые группы которой разбиты на уровни.

Примечание

Для наглядного пояснения принципа организации уровней PNNI группы на рис. 21.14 представлены в виде пирамиды.

Применяя логику уровней и распространяя ее на сотни коммутаторов, можно привести сети PNNI практически к любому масштабу. После того как уровни PNNI определены, среда готова к маршрутизации. PNNI обрабатывает перемещение данных от коммутатора к коммутатору.

В начале главы уже кратко упоминалось, что PNNI фактически разделяется (функционально) на два разных протокола. Процесс маршрутизации данных в ATM-сети предъявляет специальные требования к используемым протоколам. Поэтому в дополнение к протоколу маршрутизации PNNI необходим отдельный протокол сигнализации.

После того как конечная система отправляет данные на коммутатор, протокол сигнализации PNNI оповещает принимающий коммутатор о том, что запрашивается соединение. Это первая обязанность PNNI. Затем протокол сигнализации создает канал между двумя точками. (Подробно этот процесс будет описан в разделе «Протокол сигнализации PNNI».)

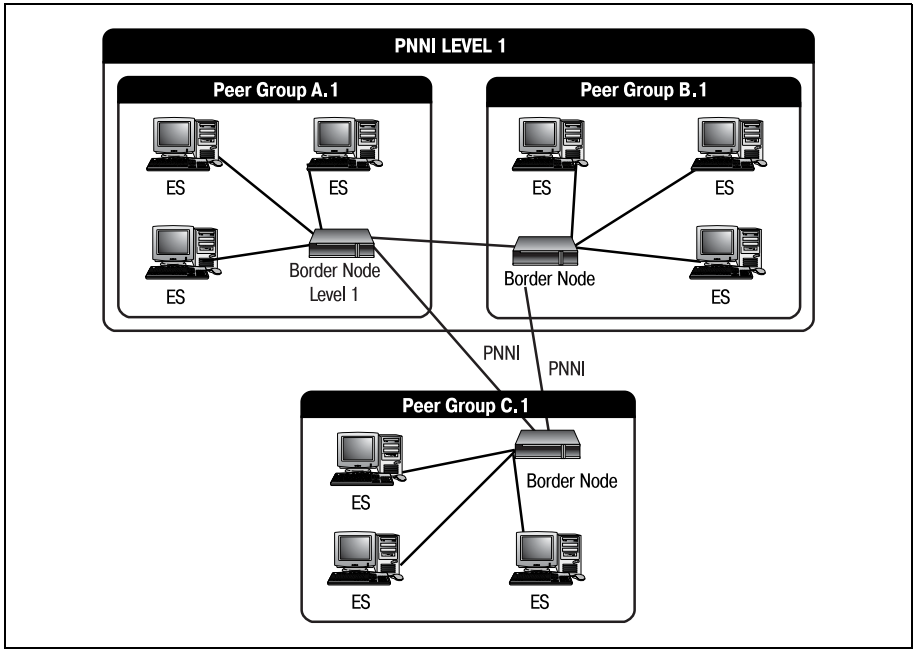

Рис. 21.14. *Уровни PNNI*

После того как канал открыт, в дело вступает вторая функция PNNI. Как будет рассказано в следующем разделе, протокол маршрутизации PNNI занимается фактическим перемещением данных от коммутатора к коммутатору. Сеть PNNI изображена на рис. 21.15, где показано, какие протоколы работают в разных областях среды ATM.

Движение потока данных (по протоколам) в ATM-сети может представляться несколько запутанным, если вы никогда не видели такого раньше. Но несмотря на то что наличие сигнальной части PNNI может показаться немного необычным, собственно маршрутизация достаточно проста.

Протокол маршрутизации PNNI

Протокол маршрутизации PNNI относится к весьма успешной категории протоколов – протоколам состояния канала. Давайте немного поговорим о протоколах состояния канала, прежде чем обсуждать технологию, лежащую в основе работы протокола маршрутизации PNNI.

Каждый коммутатор ATM-сети объявляет состояние своих каналов в сообщениях hello (приветствиях). Такие сообщения распространяются всем коммутаторам среды ATM вне зависимости от одноранговых групп. Получающая система обрабатывает информацию и создает для

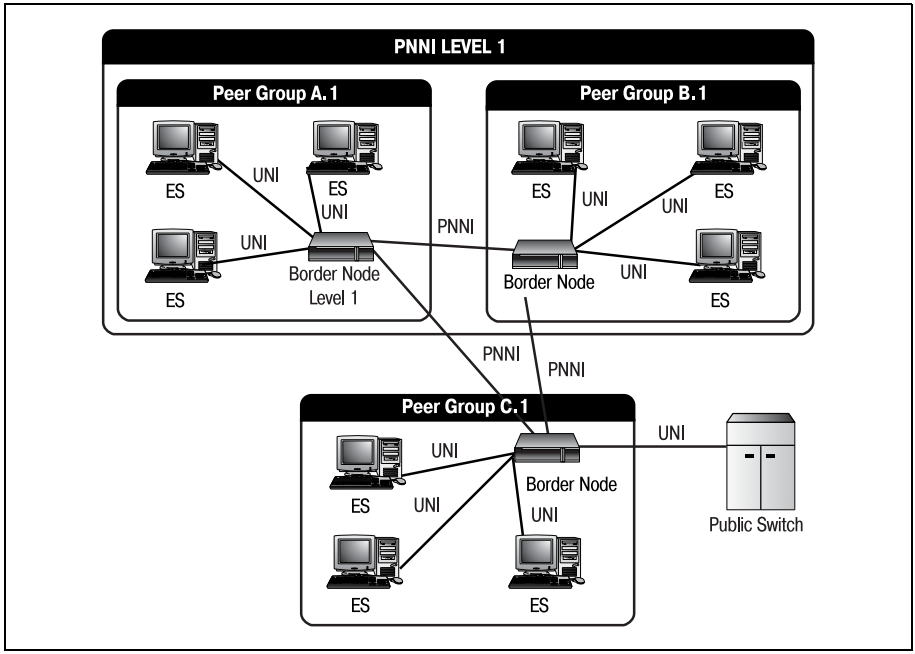

***Рис. 21.15**. Карта среды ATM, на которой отмечены используемые протоколы*

себя итоговую картину сети. На рис. 21.16 представлено прохождение сообщения hello по ATM-сети.

Коммутатор A.1 (рис. 21.16) объявляет коммутаторам B.1 и A.2 состояния всех конечных систем, непосредственно подключенных к нему, а именно, что пять связанных с ним конечных систем доступны и функционируют. В сообщение hello включены все метрики, необходимые для сопоставления отдельным системам маршрута.

Когда коммутатор A.2 получает сообщение hello от A.1, он понимает (по идентификатору одноранговой группы), что A.1 относится к его одноранговой группе. Тогда два коммутатора образуют канал, поскольку они входят в одну одноранговую группу.

Процесс отправки сообщений hello называется затоплением. В ходе этого процесса каждая система затапливает сеть сообщениями hello, объявляющими состояния ее каналов. Эти сообщения обрабатываются всеми коммутаторами сети. Затопление объявлениями состояния каналов обычно требует большой пропускной способности и очень загружает процессор.

Примечание

Рассылка сообщений hello (затопление объявлениями состояния канала) инициируется изменением среды ATM. Таким изменением может быть уничтожение или восстановление канала, изменение метрики маршрутизации или другое админист-

ративное изменение сети. Однако отдельные изменения, например, касающиеся некоторых метрик QOS (quality of service – качество обслуживания), могут быть не настолько значимы, чтобы запускать процесс затопления.

Так как объявления отправляются всеми коммутаторами, то затопления в больших сетях сильнее, чем в маленьких. Независимо от количества коммутаторов в сети применение одной меры значительно изменяет процедуру обработки затопления коммутаторами. ATM-коммутатор всегда обрабатывает сообщение hello прежде, чем приступить к обработке маршрутизируемой ячейки. Поэтому увеличение объема памяти (и, если это возможно, мощности процессора) позволит коммутатору быстрее обрабатывать сообщения, и значит, он быстрее сможет вернуться к нормальной работе. Такой процесс быстрого получения актуальной сетевой информации о маршрутах называется конвергенцией.

Конвергенция достигается, когда все коммутаторы работают с одной и той же картиной сети, то есть каждый коммутатор обработал hello-сообщения и согласился с тем, что содержимое этих сообщений описывает текущую топологию сети. (Вообще протоколы состояния канала, такие как PNNI, обладают более быстрой конвергенцией, чем другие протоколы.)

Маршрутизаторы (вне зависимости от того, какие протоколы ими используются) нуждаются в актуальной топологической информации для принятия наилучших решений. Протоколы состояния канала ис-

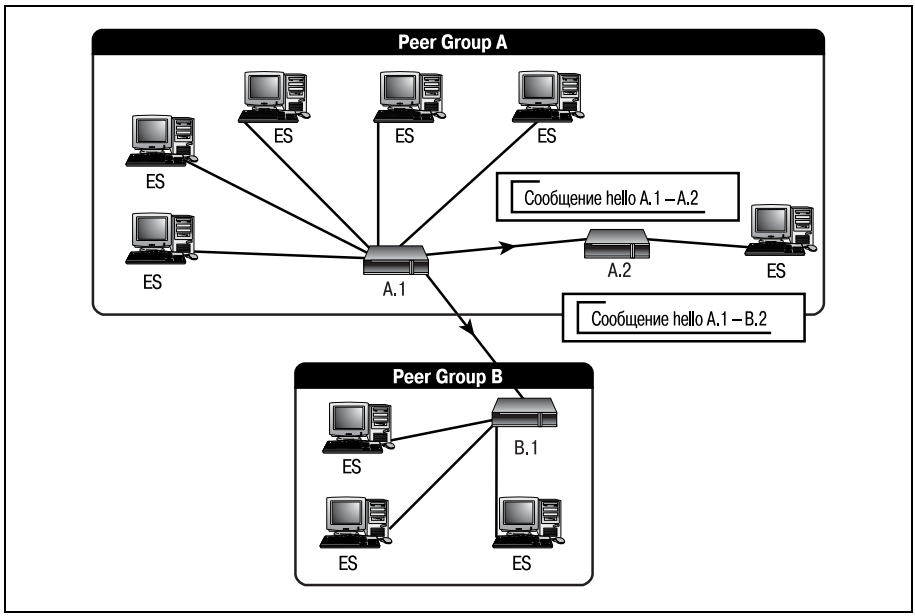

Рис. 21.16. *ATM-сообщения hello*

пользуют информацию сообщений hello (по которой строится таблица маршрутов коммутатора) для вычисления кратчайшего пути между системами. Как и другие протоколы состояния канала, PNNI работает на основе «кратчайших путей». PNNI-устройство всегда пытается выбрать кратчайший из возможных путей между двумя системами.

Выбором кратчайшего пути занимается алгоритм маршрутизации. Алгоритм маршрутизации – это формула, применяемая PNNI-устройством к метрикам из таблицы маршрутов для вычисления лучшего пути к адресату. PNNI-устройства конфигурируются с различными метриками, чтобы указать предпочтительность устройства в качестве кратчайшего пути. Для описания устройства PNNI использует следующие метрики:

- Administrative weight (AW, административный вес) – стоимость, присваиваемая администратором, абсолютно произвольное число
- Available cell rate (AvCR, доступная скорость ячеек) – пропускная способность канала
- Maximum cell transfer delay (MaxCTD, максимальная задержка передачи ячеек) – максимальная задержка, испытываемая при передаче данных по определенному каналу
- Cell loss ratio (CLR, коэффициент потери ячеек) – отношение числа потерянных ячеек к общему числу переданных ячеек
- Cell delay variation (CDV, вариация задержки) – комбинированная метрика, представляющая собой разность между максимальной и минимальной задержками передачи ячеек (MaxCTD – MinCTD)
- Maximum cell rate (MaxCR) – максимальная скорость ячеек для канала

Каждой из этих метрик динамически или статически присваивается значение. Значения используются алгоритмом маршрутизации PNNI для вычисления суммарного значения или пригодности определенного канала для достижения конечного адресата ячейки. Если комбинация метрик показывает, что текущий канал идеален для маршрутизации, то ячейки отправляются по этому каналу.

Все приведенные выше метрики, кроме административного веса, являются метриками QOS. Маршрутизация QOS (quality of service, качество обслуживания) – это стандарт, изложенный в RFC 2386.

PNNI и QOS

PNNI придерживается требований QOS для ATM-сетей. Это означает, что ATM поддерживает те же метрики для обеспечения конечных систем таким же уровнем маршрутизации, как и любая другая, удовлетворяющая QOS, архитектура. Многие протоколы были адаптированы или усовершенствованы для того, чтобы соответствовать стандартам QOS, в то время как PNNI изначально проектировался в соответствии с

QOS. Иначе говоря, никакого дополнительного программного или аппаратного обеспечения для реализации метрик маршрутизация QOS ATM не требует.

Такие метрики, как MaxCR и CLR, используются для уверенности в том, что ячейка пересылается по каналу, который в состоянии иметь с ней дело. Все метрики QOS (максимальная задержка передачи ячеек, коэффициент потери ячеек, вариация задержки и максимальная скорость ячеек) и административная метрика веса определяются статически, то есть они редко меняются без участия администратора.

QOS-метрика доступной скорости ячеек – это динамическая метрика, которая может колебаться в зависимости от трафика канала. Поскольку метрика динамическая, ее изменение обычно не вызывает обновления состояния канала. Для того чтобы инициировать рассылку обновлений по сети, изменение доступной скорости ячеек должно превысить некоторое (заранее установленное) пороговое значение.

Когда возникает необходимость переслать ячейку по сети PNNI, протокол сигнализации запрашивает минимальное (QOS) значение, необходимое каналу для успешной маршрутизации ячейки.

Примечание

Для помощи в определении доступных путей могут использоваться различные алгоритмы QOS, зависящие от того, какой коммутатор используется в среде ATM. PNNI предусматривает два алгоритма QOS: простой GCAC (Generic Call Admission Control – процесс проверки достаточности потенциальных ресурсов для поддержания соединения) и комплексный GCAC.

Простой GCAC при принятии решения учитывает только метрику доступной скорости ячеек, в то время как комплексный использует и эту метрику, и две новые метрики: изменение скорости передачи ячеек (cell rate margin) и коэффициент изменения (variance factor).

Например, протокол сигнализации PNNI может запросить определенную пропускную способность канала – тогда только те каналы, чья QOS-метрика пропускной способности соответствует запрошенной, будут рассматриваться алгоритмом как возможные пути. На рис. 21.17 представлен QOS-запрос канала.

Рисунок 21.17 иллюстрирует попытку маршрутизировать данные от A.1 к C.4. Протокол сигнализации (о котором будет рассказано в разделе «Протокол сигнализации PNNI») запрашивает для маршрутизации ячеек открытие канала с постоянной пропускной способностью не менее 56 Кбит/с. Этот запрос автоматически исключает из рассмотрения канал между B.1 и C.1.

PNNI определяет, какие пути использовать для маршрутизации данных, сравнивая метрики каждого пути при помощи алгоритма маршрутизации. Этот алгоритм исследует стоимость каждого канала и определяет, какой из них лучше использовать.

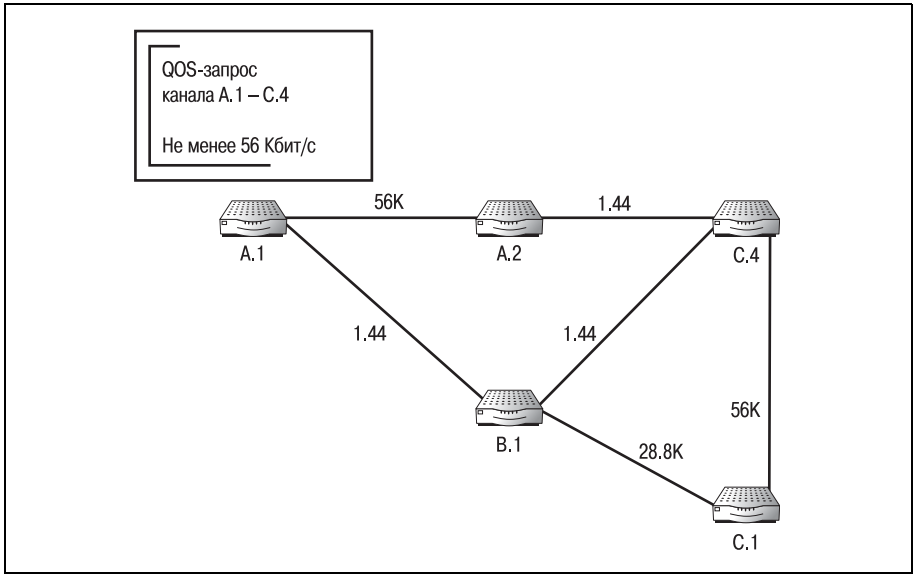

Рис. 21.17. Маршрутизация QOS

Алгоритм, предпочитаемый многими протоколами состояния канала, – это алгоритм Дейкстры. И PNNI не является исключением. PNNI может очень хорошо использовать этот алгоритм (полное описание алгоритма Дейкстры представлено в главе 17 «Конфигурирование OSPF»). После того как были учтены требования QOS, алгоритм приступает к выбору кратчайшего пути передачи данных.

Алгоритм Дейкстры использует метрики, присвоенные каждому каналу, для определения кратчайшего пути между всеми возможными точками сети одновременно. То есть алгоритм Дейкстры динамически вычисляет стоимости каждого канала сети в одно и то же время. Это делает его очень быстрым и надежным способом определения маршрутов.

Сочетание собственно QOS-маршрутизации и маршрутизации на основе кратчайших путей – это одно из тех свойств PNNI, которые делают его использование привлекательным, особенно в ATM-средах. Но протокол маршрутизации PNNI – это только половина картины. Основой архитектуры ATM являются каналы, которые необходимо создавать, чтобы затем пересылать по ним данные. За создание таких каналов отвечает протокол сигнализации PNNI.

Протокол сигнализации PNNI

Протокол сигнализации PNNI – это та часть PNNI, которая необходима для установления и закрытия соединений между устройствами. При запросе открытия канала для маршрутизации протокол сигнали-

зации учитывает различные метрики, в том числе и метрики QOS. После того как протокол сигнализации успешно создал канал между двумя точками, за работу принимается протокол маршрутизации PNNI.

Когда конечная система сети ATM хочет отправить данные, протокол сигнализации PNNI начинает процесс, связываясь с ближайшим непосредственно подключенным к конечной системе ATM-коммутатором и уведомляя его о запросе. На рис. 21.18 изображена конечная ATM-система, отправляющая запрос на маршрутизацию.

На рис. 21.18 протокол сигнализации PNNI связывается с коммутатором A.1 от имени конечной системы и сообщает ему, что конечная система просит, чтобы коммутатор открыл канал к конечной системе в одноранговой группе D.

Примечание

Хотя конечная система знает, с какими именно конечными системами одноранговой группы D она бы хотела связаться, но у нее нет топологической информации об этой группе. Она просто запрашивает канал к пограничному узлу одноранговой группы. Затем пограничный узел одноранговой группы D отправит ячейки нужной конечной системе.

Протокол сигнализации также отправляет ряд QOS-запросов. Эти запросы (или минимальные требования) передаются от одного коммутатора к другому. Когда коммутатор получает QOS-требования, он сначала находит каналы, метрики которых отвечают требованиям, а остальные каналы исключаются из рассмотрения. Затем запрос передается по оптимальному пути следующему коммутатору.

Примечание

Если ни один канал не соответствует QOS-требованиям сигнального запроса, то запрос отклоняется. Информация исходной конечной системы отбрасывается.

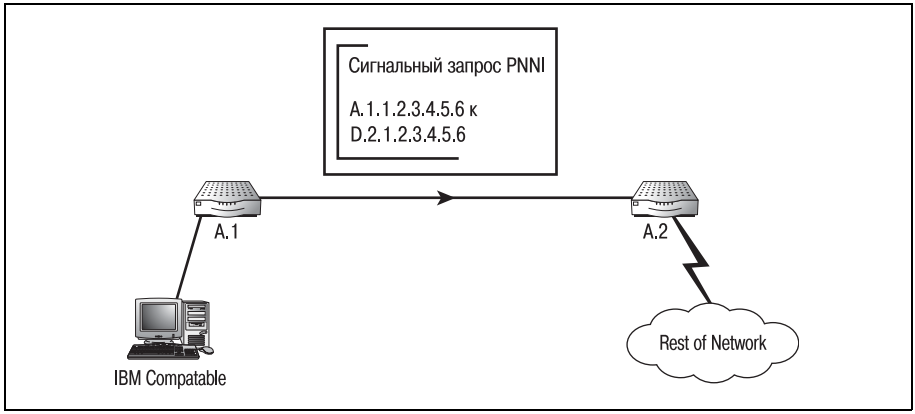

Рис. 21.18. *Конечная система, подготавливающая запрос на открытие канала*

В конце концов, сигнальный запрос достигает пограничного узла области назначения. Теперь коммутатор может или принять канал, или отказаться от него. Если канал принят, то пограничный узел передает ячейки адресату – конечной системе, и между двумя системами открывается канал. Если же пограничный узел области назначения отклоняет запрос, то данные конечной системы отбрасываются.

На рис. 21.19 изображена ATM-сеть, в которой для всех каналов проставлены метрики. Давайте посмотрим на путь протокола сигнализации от конечной системы A.1.X.X.X.X.X.X к конечной системе D.1.X.X.X.X.X.X.

На рис. 21.19 протокол сигнализации отправляет запрос коммутатору A.1. QOS-требования состоят в том, что канал должен иметь пропускную способность не ниже 56 Кбит/с. Поэтому когда A.1 получает сигнальный запрос, он оценивает все свои каналы, чтобы определить, какие из них имеют пропускную способность 56 Кбит/с. В нашем случае A.1 имеет всего один канал (B.1), и он обладает соответствующей пропускной способностью. Так что сигнальный запрос пересылается на B.1.

Коммутатор B.1 может выбрать одно из двух: отправить ячейки по каналу B.1–C.1 или же по B.1–C.2. Так как канал B.1–C.2 не удовлетворяет требованиям QOS, то этот канал не учитывается при выборе пути, и запрос пересылается C.1.

Когда C.1 получает запрос, у него также есть два варианта выбора канала. Оба канала C.1–C.2 и C.1–D.1 соответствуют требованиям QOS. Тогда коммутатор обращается к своему алгоритму маршрутизации,

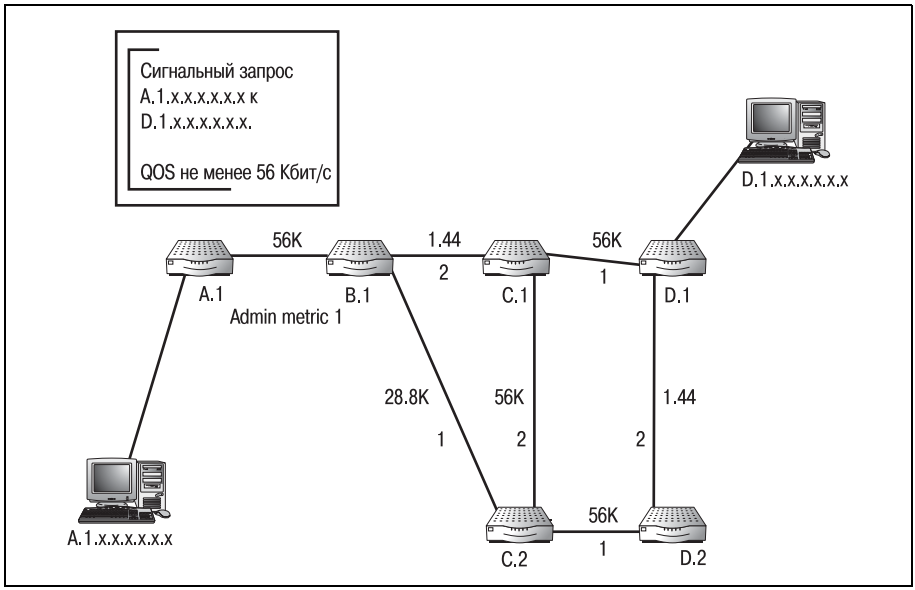

Рис. 21.19. *Пример сигнального запроса в ATM-сети*

который определяет, что кратчайшим путем является канал C.1–D.1, и сигнальный запрос пересылается D.1.

D.1 распознает нахождение конечной системы D.1.X.X.X.X.X.X в своей одноранговой группе. Теперь ATM-коммутатор D.1 может принять запрос на открытие канала, переслав его конечной системе D.1.X.X.X.X.X.X, а может отклонить запрос. Если D.1 принимает запрос, то между двумя конечными системами устанавливается соединение. Завершенный канал изображен на рис. 21.20.

Если D.1 отклоняет запрос, то коммутаторы C.1, B.1 и A.1 уведомляются об этом посредством возврата специального сообщения. Это сообщение отменяет запросы открытия канала на коммутаторах и отбрасывает любые ячейки (относящиеся к каналу), которые коммутатор мог обработать.

Когда конечная система A.1.X.X.X.X.X.X завершает свой сеанс связи с D.1.X.X.X.X.X.X, то независимо от того, был сигнал принят или нет, протокол сигнализации PNNI закрывает канал, оставляя пути свободными для маршрутизации.

Некоторые метрики QOS являются динамическими, поэтому есть вероятность того, что они изменятся в процессе передачи данных. PNNI обладает встроенным механизмом обработки таких событий под названием «crankback», который может быть приведен в действие, если метрики QOS неблагоприятно изменяются во время передачи данных.

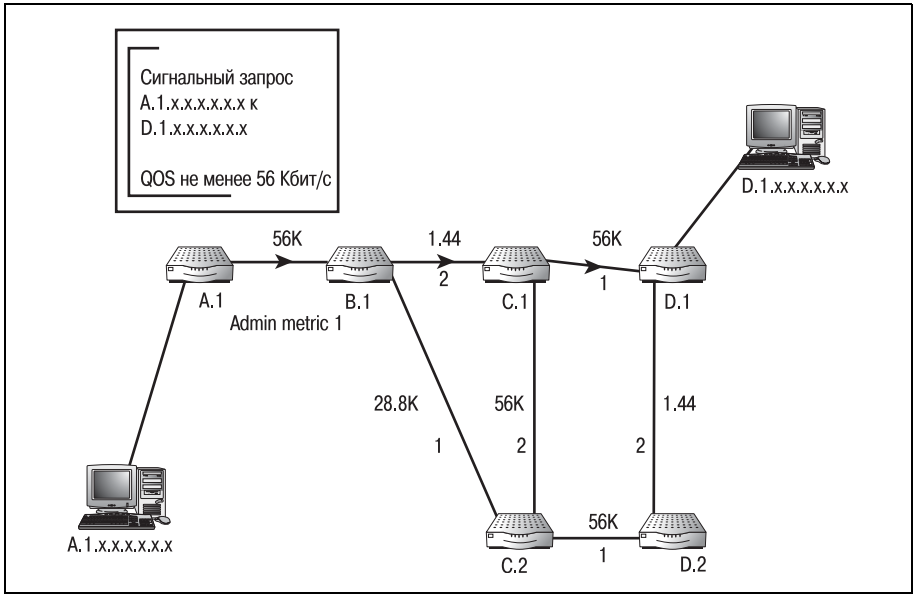

Рис. 21.20. *Завершенный ATM-канал*

Специальный механизм блокирования PNNI (crankback)

Во время существования канала такая QOS-метрика, как доступная скорость ячеек, может измениться. Если изменение QOS-метрики пагубно для канала (например, канал требует задержки в 10 миллисекунд, а величина задержки подскочила до 200 мс), то протокол PNNI выполняет специальную блокировку запроса, называемую «crankback». На рис. 21.21 видно, что в результате использования специального механизма блокирования протокол возвращает запрос последнему ATM-коммутатору, который все еще соответствует требованиям QOS. На рис. 21.22 изображен канал, для которого выполнен откат назад в результате выполнения блокировки запроса.

Канал создается заново, начиная с последнего подходящего коммутатора. Когда новый канал создан, еще раз оцениваются требования QOS.

Обсудив принципы коммутации ATM и маршрутизации PNNI, перейдем к рассмотрению команд, которые необходимы для настройки маршрутизации PNNI на коммутаторе Cisco.

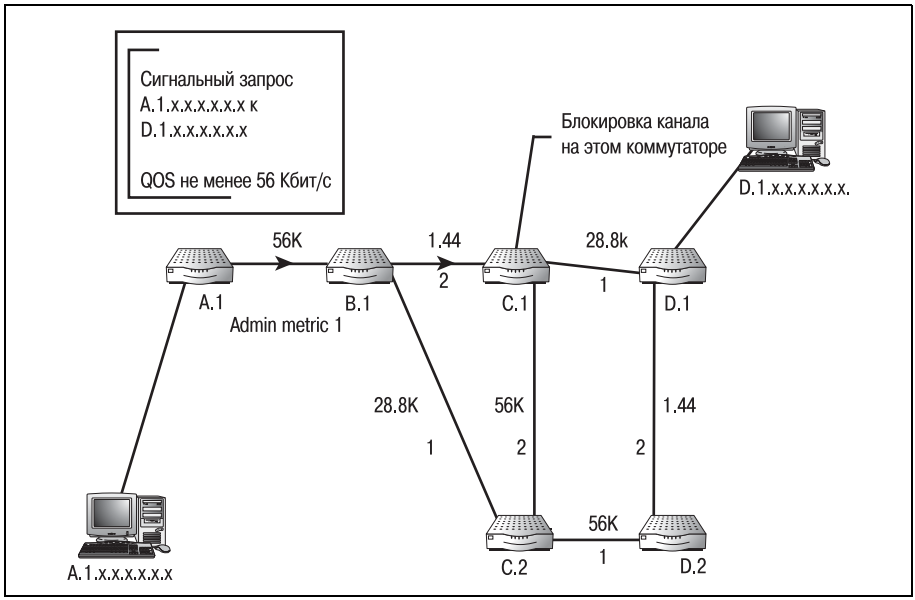

Рис. 21.21. *Завершенный ATM-канал с метриками QOS*

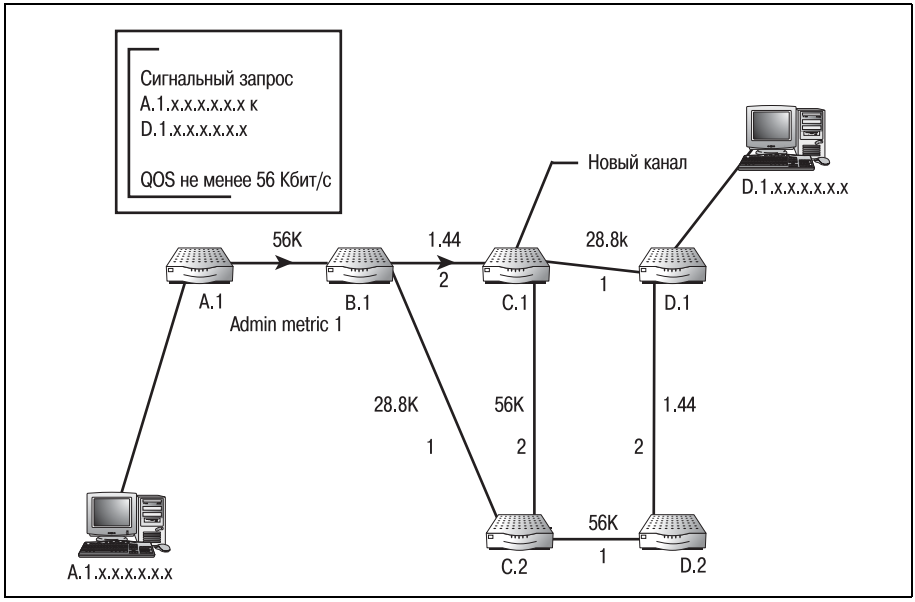

***Рис. 21.22.** Блокирование канала*

Конфигурирование PNNI

Команды настройки маршрутизации на маршрутизаторах и коммутаторах Cisco мало отличаются друг от друга. Вам должны показаться знакомыми как структура команд, так и их параметры. Сначала необходимо разрешить коммутатору маршрутизацию PNNI:

```
Switch#configure terminal
Switch(config)#atm router pnni
Switch(config-atm-router)# aesa embedded-number left-justified
```

Примечание

Последняя команда из предыдущего примера используется для разрешения кодировки PNNI версии 2. Префиксы E.164-адресов конечных систем (aesa – ATM end system address – адрес конечной системы ATM) будут преобразованы автоматически.

Команда `atm router` разрешает маршрутизацию PNNI. Вам также необходимо знать и несколько второстепенных команд. Например, для указания ATM-адреса коммутатора используйте команду `atm address`:

```
Switch#configure terminal
Switch(config)#atm address 52.000a.c001.034b.000.06a7.0001. ..
```

Три точки в конце строки указывают коммутатору, что введенный адрес является префиксом и для образования ATM-адреса к нему необходимо добавить MAC-адрес по умолчанию. Из названия коммутаторов очевидно, что маршрутизация – не основная функция этих устройств. Но коммутаторы обладают способностью использовать очень эффективный протокол маршрутизации. К сожалению, мы не имеем возможности рассказать в одной главе обо всех командах, относящихся к ATM и PNNI (а их очень много), – все же основное внимание книги сосредоточено на маршрутизации.

Целью этой главы было познакомить вас с протоколом, который используется в большом количестве сред как маршрутизаторами, так и коммутаторами Cisco. Существует множество материалов, доступных как в печатном виде, так и в Интернете, в которых вы можете прочитать об упомянутых в данной главе вопросах. Надеюсь, что изучив книгу до конца, вы приобрели те базовые знания, которые необходимы для понимания начал маршрутизации Cisco.

Примите поздравления! Изучение базовых элементов, необходимых для понимания основ функционирования маршрутизаторов Cisco, завершено. В последней главе вы также познакомились с некоторыми более сложными технологиями маршрутизации при помощи коммутаторов Cisco. Если вам необходима дополнительная информация по любой из рассмотренных тем, обращайтесь к базе знаний Cisco, доступной в Интернете по адресу *www.cisco.com*, или к другим посвященным маршрутизации книгам, которые выпущены SAMS Publishing.

A

Справочник команд Cisco

Мы приводим достаточно подробный перечень команд Cisco IOS и их параметров с описаниями. В справочник включены команды пользовательского и привилегированного режимов. Хотя определения и характеристики многих команд можно найти в самой Cisco IOS, начинающим будет полезен наш простой справочник. Для иллюстрации синтаксиса команд приведены простые примеры.

Примечание

Для получения информации, приведенной в разделе «Описание Cisco» для каждой команды, использована справочная система Cisco.

Команды пользовательского режима

access-enable

Описание Cisco

Создает временную запись в списке доступа. (Команда используется только для виртуальных каналов.)

Параметры

Параметр	Описание
host	Разрешает работу только указанного узла (host)
timeout	Максимальное время ожидания для окончания действия данной записи
<1–9999>	Тайм-аут времени ожидания для списка доступа

Параметр	Описание
timeout	Максимальное время ожидания для окончания действия данной записи
<1–9999>	Тайм-аут времени ожидания для списка доступа

Пример синтаксиса

```
>access-enable host timeout 1
```

access-profile

Описание Cisco

Применяет профиль пользователя к интерфейсу. (Команда используется только для виртуальных каналов.)

Параметры

Параметр	Описание
ignore-sanity-checks	Игнорировать все ошибки, выявленные проверкой на готовность к работе (используется для игнорирования ошибок, являющихся результатом неправильного форматирования AV-пары аутентификационной информации)
merge	Объединить старый и новый профили пользователя, удалив только списки доступа
ignore-sanity-checks	Игнорировать все ошибки, выявленные проверкой на готовность к работе (используется для игнорирования ошибок, являющихся результатом неправильного форматирования AV-пары аутентификационной информации)
replace	Удалить старую конфигурацию пользователя, заменив ее новой
ignore-sanity-checks	Игнорировать все ошибки, выявленные проверкой на готовность к работе (используется для игнорирования ошибок, являющихся результатом неправильного форматирования AV-пары аутентификационной информации)

Пример синтаксиса

```
>access-profile merge
```

connect

Описание Cisco

Открывает терминальное соединение с другим устройством. Единственный параметр – это имя или адрес внешнего устройства.

Параметры

Параметр	Описание
WORD	IP-адрес или имя узла удаленной системы

Пример синтаксиса

```
>connect RouterA
```

disable

Описание Cisco

Выключает привилегированный режим. Команда принимает в качестве необязательного параметра номер конкретного привилегированного режима.

Параметр

Параметр	Описание
(0–15)	Номер уровня

Пример синтаксиса

```
>disable 14
```

disconnect

Описание Cisco

Разрывает существующее сетевое соединение (то есть завершает удаленную сессию). Параметром этой команды может быть имя или номер соединения.

Параметры

Параметр	Описание
<1–20>	Номер активного сетевого соединения
WORD	Имя активного сетевого соединения

Пример синтаксиса

```
>disconnect 1
>disconnect RouterA
```

enable

Описание Cisco

Включает привилегированный режим. Если администратор задал несколько уровней доступа, то номер уровня может быть необязательным параметром.

Параметр	Описание
(0–15)	Номер уровня

Пример синтаксиса

```
>enable 14
```

exit

Описание Cisco

Выход из командного интерпретатора EXEC. Используется для выхода из сессии IOS; без параметров.

Пример синтаксиса

```
>exit
```

help

Описание Cisco

Описание интерактивной справочной системы; без параметров.

Пример синтаксиса

```
>help
```

lock

Описание Cisco

Команда блокирует экран удаленного терминала. Не имеет параметров.

Пример синтаксиса

```
>lock
```

login

Описание Cisco

Войти в систему под определенным именем пользователя.

Эта команда аутентифицирует пользователя маршрутизатора на специальном сервере регистрации. Если сервер регистрации недоступен, то команда не принимает параметров.

Пример синтаксиса

```
>login
```

logout

Описание Cisco

Выход из EXEC; без параметров.

Пример синтаксиса

```
>logout
```

name-connection

Описание Cisco

Присваивает имя существующему сетевому соединению. Эта команда используется для назначения имени соединению с удаленным устройством для упрощения процесса работы с несколькими сессиями. Команда не имеет параметров, но инициирует диалог на основе меню.

Пример синтаксиса

```
>name-connection 1 RouterA
```

pad

Описание Cisco

Открывает соединение X.29 PAD (packet assembler/disassembler – сборщик/разборщик пакетов). Команда используется для подключения к внешнему устройству X.25 PAD.

Параметры

Параметр	Описание
WORD	Адрес в формате X.121 или имя удаленной системы

Пример синтаксиса

```
>pad Remote
```

ping

Описание Cisco

Отправляет запросы ожидания отклика (эхо-пакеты).

Параметры

Параметр	Описание
WORD	Адрес или имя адресата ping
ip	IP-эхо
WORD	Адрес или имя адресата ping
ipx	Novell/IPX-эхо
WORD	Адрес или имя адресата ping
tag	Инкапсулированный IP эхо-пакет с тегами
WORD	Адрес или имя адресата ping

Пример синтаксиса

```
>ping 10.25.123.3
>ping ip 10.25.123.3
```

ppp

Описание Cisco

Запускает протокол IETF PPP (Point-to-Point Protocol – протокол «точка-точка»).

Параметр

Параметр	Описание
negotiate	Использовать IP-адрес, установленный PPP

Пример синтаксиса

```
>ppp negotiate
```

resume

Описание Cisco

Возобновить активное сетевое соединение. Эта команда имеет несколько опций (любая из них может быть использована), которые вводятся при помощи символа косой черты /.

Опции

Параметр	Описание
/debug	Выводить изменения параметров и сообщения
/echo	Выполнять локальное эхо
/line	Разрешить строковый режим Telnet
/next	Переход к следующему сетевому соединению
/nodebug	Не выводить изменения параметров и сообщения
/noecho	Запретить локальное эхо
/noline	Запретить строковый режим Telnet
/nostream	Запретить потоковую обработку
/set	Установить опции соединения X3
/stream	Разрешить потоковую обработку

Параметры

Параметр	Описание
<1–20>	Номер активного сетевого соединения
WORD	Имя активного сетевого соединения или опции соединения

Пример синтаксиса

```
>resume /noecho RouterA
```

rlogin

Описание Cisco

Открывает rlogin-соединение.

Параметр

Параметр	Описание
WORD	IP-адрес или имя узла удаленной системы

Пример синтаксиса

```
>rlogin 10.25.123.3

>rlogin RouterA
```

show

Описание Cisco

Выводит информацию о работающей системе. Команда `show` – это команда, предоставляющая всесторонние данные и используемая для просмотра и отслеживания статистики маршрутизатора.

Параметры

Параметр	Описание
`clock`	Выводит системное время
`detail`	Выводит подробную информацию
`dialer`	Выводит параметры наборного устройства и статистику по нему
`maps`	Выводит карты наборного устройства
`flash:`	Выводит информацию о файловой системе `flash:`
`all`	Выводит всю возможную информацию о флэш-памяти
`chips`	Выводит информацию о микросхемах флэш-памяти
`detailed`	Выводит подробное содержание каталога флэш-памяти
`err`	Показывает попытки стирания и перезаписи флэш-памяти
`summary`	Выводит сводную информацию о разбиении флэш-памяти
`history`	Выводит историю команд сессии

Параметр	Описание
hosts	Выводит доменное имя IP, режим поиска, серверы имен и host-таблицу
WORD	Имя конкретного узла
location	Выводит расположение системы
Modemcap	Показывает базу данных Modem Capabilities (характеристики модема)
WORD	Запись, относящаяся к определенному модему
ppp	Выводит параметры и статистику протокола PPP
bap	Выводит параметры и статистику PPP-протокола BAP (Bandwidth Allocation Protocol – протокол распределения (назначения) пропускной способности)
group	Информация о группе BAP
WORD	Имя группы
queues	Информация об очередях BAP
multilink	Выводит информацию о пучке каналов Multilink PPP
queues	Выводит очередь запросов PPP
rmon	Показывает статистику RAM Monitor (RMON)
alarms	Выводит таблицу предупреждений RMON
events	Выводит таблицу событий RMON
events	Выводит таблицу событий RMON
alarms	Выводит таблицу предупреждений RMON
rtr	Показывает данные о RTR (Response Time Reporter – генератор отчетов о времени реакции системы)
application	Приложение RTR
full	Полный вывод
tabular	Компактный вывод
collection-statistics	Наборы статистики RTR
<1–2147483647>	Номер записи
full	Полный вывод (для выбранной записи)
tabular	Компактный вывод (для выбранной записи)
full	Полный вывод (для всех наборов)
tabular	Компактный вывод (для всех наборов)
configuration	Конфигурация RTR
<1–2147483647>	Номер записи
full	Полный вывод (для выбранной записи)
tabular	Компактный вывод (для выбранной записи)

Параметр	Описание
full	Полный вывод (все конфигурации)
tabular	Компактный вывод (все конфигурации)
distributions-statistics	Выводит статистику распределений RTR
<1–2147483647>	Номер записи
full	Полный вывод (для выбранной записи)
tabular	Компактный вывод (для выбранной записи)
full	Полный вывод (все распределения)
tabular	Компактный вывод (все распределения)
history	История RTR
<1–2147483647>	Номер записи
full	Полный вывод (для выбранной записи)
tabular	Компактный вывод (для выбранной записи)
full	Полный вывод (все исторические записи)
tabular	Компактный вывод (все исторические записи)
operational-state	Рабочее состояние RTR
<1–2147483647>	Номер записи
full	Полный вывод (для выбранной записи)
tabular	Компактный вывод (для выбранной записи)
full	Полный вывод (все состояния)
tabular	Компактный вывод (все состояния)
reaction-trigger	Триггер реакций RTR
<1–2147483647>	Номер записи
full	Полный вывод (для выбранной записи)
tabular	Компактный вывод (для выбранной записи)
full	Полный вывод (все реакции)
tabular	Компактный вывод (все реакции)
totals-statistics	Итоговые статистические данные RTR
<1–2147483647>	Номер записи
full	Полный вывод (для выбранной записи)
tabular	Компактный вывод (для выбранной записи)
full	Полный вывод (все итоги)
tabular	Компактный вывод (все итоги)
sessions	Показывает информацию о Telnet-соединениях
snmp	Показывает статистику протокола SNMP (Simple Network Management Protocol – простой протокол сетевого управления)

Параметр	Описание
tacacs	Показывает статистику сервера TACACS+ (Terminal Access Controller Access Control System)
terminal	Выводит параметры конфигурации терминала
traffic-shape	Показывает параметры формирования трафика
Ethernet	IEEE 802.3
<0–1>	Номер интерфейса Ethernet
Null	Null-интерфейс
<0–0>	Номер Null-интерфейса
queue	Выводит содержимое очереди формирования трафика
Ethernet	IEEE 802.3
<0–1>	Номер интерфейса Ethernet
Null	Null-интерфейс
<0–0>	Номер Null-интерфейса
statistics	Статистика формирования трафика
Ethernet	IEEE 802.3
<0–1>	Номер интерфейса Ethernet
Null	Null-интерфейс
<0–0>	Номер Null-интерфейса
users	Выводит информацию о терминальных линиях
all	Включать информацию о неактивных портах
wide	Использовать широкий формат (включая неактивные порты)
wide	Использовать широкий формат
version	Выводит состояние аппаратного и программного обеспечения системы

Пример синтаксиса

```
>show system detailed
```

slip

Описание Cisco

Запускает протокол SLIP (Serial-Line IP).

Пример синтаксиса

```
>slip
```

systat

Описание Cisco

Выводит информацию о терминальных линиях.

Параметр

Параметр	Описание
all	Включать данные о неактивных портах

Пример синтаксиса

```
>systat all
```

telnet

Описание Cisco

Открывает Telnet-соединение с удаленным маршрутизатором или другим Telnet-устройством.

Параметр

Параметр	Описание
WORD	IP-адрес или имя узла удаленной системы

Пример синтаксиса

```
>telnet 10.25.123.3

>telnet RouterA
```

terminal

Описание Cisco

Устанавливает параметры терминальной линии.

Параметры

Параметр	Описание
autohangup	Автоматически отсоединяться при закрытии последнего соединения
session-timeout	Автоматически отсоединяться по тайм-ауту текущей сессии
data-character-bits	Размер обрабатываемых символов
<7–8>	Количество бит на символ
databits	Устанавливает количество двоичных разрядов в символе
<5–8>	Количество двоичных разрядов

Параметр	Описание
default	Устанавливает для команды значения параметров по умолчанию
Принимает любые корректные параметры	
dispatch-character	Определяет символ отправки пакетов
CHAR или <0–127>	Символ отправки пакетов или его десятичное значение
dispatch-timeout	Устанавливает таймер диспетчеризации
<0–4294967294>	Значение таймера диспетчеризации в миллисекундах
domain-lookup	Разрешает поиск по DNS в командах show
download	Включает для строки режим «загрузки»
editing	Разрешает редактирование командной строки
escape-character	Изменяет escape-символ для текущей строки
BREAK	Выход по BREAK
CHAR или <0–255>	Escape-символ или его десятичный ASCII-эквивалент
DEFAULT	Использовать escape-символ, заданный по умолчанию
NONE	Полностью запретить escape-символы
soft	Устанавливает «мягкий» escape-символ для строки
BREAK	Выход по BREAK
CHAR или <0–255>	Escape-символ или его десятичный ASCII-эквивалент
DEFAULT	Использовать escape-символ, заданный по умолчанию
NONE	Полностью запретить escape-символы
exec-character-bits	Размер символов в командном интерпретаторе
<7–8>	Количество бит на символ
flowcontrol	Устанавливает режим управления потоком
NONE	Установить режим без управления потоком
hardware	Установить аппаратное управление потоком
in	Принимать сигналы управления потоком от присоединенного устройства
out	Посылать сигналы управления потоком присоединенному устройству
software	Установить программное управление потоком
in	Принимать сигналы управления потоком от присоединенного устройства

Параметр	Описание
lock	Игнорировать запросы на изменение режима управления потоком
out	Посылать сигналы управления потоком присоединенному устройству
full-help	Предоставить помощь непривилегированному пользователю
help	Описание интерактивной системы помощи
history	Разрешает и настраивает функцию истории команд
size	Установить размер буфера истории
<0–256>	Размер буфера истории
hold-character	Установить символ удерживания
CHAR или <0–255>	Символ удерживания или его десятичное значение
international	Включить поддержку международных 8-битовых символов
ip	Параметры IP
netmask-format	Изменить представление маски сети
bit-count	Выводить маску сети как количество значащих битов
decimal	Выводить маску в десятичном формате с точкой
hexadecimal	Выводить маску сети в шестнадцатеричном формате
tcp	Параметры TCP
input-coalesce-threshold	Установить порог объединения пакетов (по умолчанию – 20)
length	Устанавливает количество строк на экране
<0–512>	Количество строк на экране (0 для непрерывного вывода)
no	Инвертирует команду или устанавливает для нее параметры по умолчанию
notify	Информирует пользователя о наличии вывода в параллельных сессиях
padding	Устанавливает символ-заполнитель
CHAR или <0–127>	Символ для заполнения
parity	Настраивает контроль терминала по четности
even	Проверка на четность
mark	Контроль по единичному биту четности
none	Нет контроля по четности

Параметр	Описание
odd	Проверка на нечетность
space	Контроль по нулевому биту четности
rxspeed	Устанавливает скорость приема
<0–4294967295>	Скорость приема
special-character-bits	Размер специальных символов
<7–8>	Количество бит на символ
speed	Устанавливает скорости передачи и приема
<0–4294967295>	Скорости передачи и приема
start-character	Устанавливает символ начала
CHAR или <0–255>	Символ начала или его десятичный эквивалент
stop-character	Устанавливает символ остановки
CHAR или <0–255>	Символ остановки или его десятичный эквивалент
stopbits	Устанавливает стоп-биты для асинхронной линии
1	Один стоп-бит
1.5	Полтора стоп-бита
2	Два стоп-бита
telnet	Установка специальных параметров протокола Telnet
break-on-ip	При получении сигнала interrupt отправляет break
ip-on-break	При получении сигнала break отправляет interrupt
refuse-negotiations	Запретить согласование опций Telnet Remote Echo и Suppress Go Ahead
speed	Определяет быстродействие линии
<1–4294967295>	Скорость по умолчанию
sync-on-break	Отправляет Telnet-сигнал synchronize после получения Telnet-сигнала break
transparent	Отправляет CR как CR, за которым следует NULL, а не как CR, за которым следует LF
terminal-type	Задает тип терминала
WORD	Тип терминала
transport	Определяет транспортные протоколы для линии
all	Все протоколы
none	Ни одного протокола
pad	X.3 PAD (packet assembler/disassembler – сборщик/разборщик пакетов)
rlogin	Unix-протокол rlogin

Параметр	Описание
telnet	Протокол TCP/IP Telnet
txspeed	Устанавливает скорость передачи
<0–4294967295>	Скорость передачи
width	Определяет длину строки, отображаемой на экране терминала
<0–512>	Количество символов в экранной строке

Пример синтаксиса

```
>terminal autohangup session-timeout
```

traceroute

Описание Cisco

Прослеживает маршрут к указанной цели.

Параметры

Параметр	Описание
WORD	Отследить маршрут к узлу, заданному адресом или именем
appletalk	Трассировка AppleTalk
WORD	Отследить маршрут к узлу, заданному адресом или именем
clns	Трассировка ISO Connectionless Network Service (CLNS)
WORD	Отследить маршрут к узлу, заданному адресом или именем
ip	Трассировка IP
WORD	Отследить маршрут к узлу, заданному адресом или именем
ipx	Трассировка IPX
WORD	Отследить маршрут к узлу, заданному адресом или именем
oldvines	Трассировка VINES (Virtual Integrated Network Service), (Cisco)
WORD	Отследить маршрут к узлу, заданному адресом или именем
vines	Трассировка VINES (Banyan)
WORD	Отследить маршрут к узлу, заданному адресом или именем

Пример синтаксиса

```
>traceroute RouterA

>traceroute 10.25.123.3

>traceroute ip 10.25.123.3
```

tunnel

Описание Cisco

Открывает туннельное соединение.

Параметр

Параметр	Описание
WORD	Адрес или доменное имя удаленной системы

Пример синтаксиса

```
>tunnel RouterA
```

where

Описание Cisco

Выводит список активных соединений.

Пример синтаксиса

```
>where
```

x28

Описание Cisco

Войти в режим X.28 PAD (packet assembler/disassembler – сборщик/разборщик пакетов). Может быть использована произвольная комбинация параметров X.28.

Параметры

Параметр	Описание
debug	Включить вывод отладочных сообщений режима X.28 PAD
escape	Установить строку для выхода из режима X.28 PAD
noescape	Никогда не выходить из режима X.28 (используйте с осторожностью!)
nuicud	Все звонки с NUI (Network User ID) относятся на счет звонящего, при этом NUI помещается в пользовательские данные звонка
profile	Использовать предопределенный профиль X.3
reverse	По умолчанию все звонки относятся на счет адресата (reverse charge)
verbose	Включить подробные сообщения для режима X.28

Пример синтаксиса

```
>x.28 debug nuicud

>x.28 escape nuicud debug
```

x3

Описание Cisco

Устанавливает параметры X.3 для PAD.

Параметр

Параметр	Описание
<0–22>:<0–255>	Параметры и значения X.3 для PAD

Пример синтаксиса

```
>X3 5:225
```

Команды привилегированного режима

Ниже приведены некоторые команды привилегированного режима и описано их использование. Существует много других команд, выполняющихся в привилегированном режиме, но здесь перечислены только те из них, которые отсутствуют в пользовательском режиме.

Примечание

Помните, что эти команды, все или частично, могут быть заблокированы для некоторых пользователей путем установки уровня доступа привилегированного режима.

access-template

Описание Cisco

Создает временную запись в списке доступа. Эта команда создает временный список доступа, который пользователи могут вызвать на ограниченное время.

Параметры

Параметр	Описание
<100–199>	Расширенный список доступа IP
WORD	Имя временного списка доступа
A.B.C.D	Адрес отправителя
A.B.C.D	Шаблон адреса отправителя
A.B.C.D	Адрес получателя
A.B.C.D	Шаблон адреса получателя
timeout	Максимальное время ожидания для данной записи
<1–9999>	Значение тайм-аута списка доступа
any	Любой получатель

Параметр	Описание
host	Единственный получатель
Hostname или *A.B.C.D*	Адрес отправителя
A.B.C.D	Адрес получателя
A.B.C.D	Шаблон адреса получателя
timeout	Максимальное время ожидания для данной записи
<1–9999>	Значение тайм-аута списка доступа
any	Любой отправитель
A.B.C.D	Адрес получателя
A.B.C.D	Шаблон адреса получателя
timeout	Максимальное время ожидания для данной записи
<1–9999>	Значение тайм-аута списка доступа
host	Единственный отправитель
Hostname или *A.B.C.D*	Адрес отправителя
A.B.C.D	Адрес получателя
A.B.C.D	Шаблон адреса получателя
timeout	Максимальное время ожидания для данной записи
<1–9999>	Значение тайм-аута списка доступа
<2000–2699>	Расширенный список доступа IP (дополнительный диапазон)
Принимает те же параметры, что и расширенный список доступа IP	
WORD	Имя списка доступа

Пример синтаксиса

```
#access-template 100 test 10.25.123.3 0.255.255.255 10.35.124.4 0.255.255.255 timeout 1
```

cd

Описание Cisco

Изменяет текущий каталог.

Параметры

Параметр	Описание
flash:	Название каталога
flh:	Название каталога
null:	Название каталога
nvram:	Название каталога
system:	Название каталога

Пример синтаксиса

```
#cd flash: files
```

clear

Описание Cisco

Функция очистки.

Параметры

Параметр	Описание
access-list	Удаляет статистическую информацию списка доступа
counters	Очистить счетчики списка доступа
<0–199>	Номер списка доступа
WORD	Имя списка доступа
access-template	Удаляет шаблон доступа
<100–199>	Расширенный список доступа IP
Список параметров access-list ***см. в*** access-template	
<2000–2699>	Расширенный список доступа IP (дополнительный диапазон)
Список параметров access-list ***см. в*** access-template	
arp-cache	Очищает кэш ARP (Address Resolution Protocol)
bridge	Очищает кэш пересылки моста
<1–255>	Номер группы моста
cdp	Удаляет информацию CDP (Cisco Discovery Protocol)
counters	Очистить счетчики CDP
table	Очистить таблицы CDP
counters	Сбрасывает счетчики на одном или всех интерфейсах
Ethernet	IEEE 802.3
<0–1>	Номер интерфейса Ethernet
line	Терминальная линия
<0–5>	Номер первой линии
<1–5>	Номер последней линии
console	Основная терминальная линия
<0–0>	Номер первой линии
vty	Виртуальный терминал
<0–4>	Номер первой линии

Параметр	Описание
<1–4>	Номер последней линии
Null	Null-интерфейс
<0–0>	Номер Null-интерфейса
dialer	Удаляет статистику наборного устройства
frame-relay-inarp	Удаляет обратные ARP-записи из таблицы соответствий
host	Удаляет записи из host-таблицы
WORD	Удаляемое имя (* для всех записей)
interface	Сбрасывает аппаратную логику интерфейса
Ethernet	IEEE 802.3
<0–1>	Номер интерфейса Ethernet
Null	Null-интерфейс
<0–0>	Номер Null-интерфейса
ip	IP
access-list	Удалить статистическую информацию списка доступа
Список параметров `access-list` ***см. в*** `access-template`	
access-template	Шаблон доступа
Список параметров `access-list` ***см. в*** `access-template`	
accounting	Очистить учетную базу данных IP
checkpoint	Очистить учетную базу контрольных точек IP
cache	Удалить записи кэш-таблицы
A.B.C.D	Префикс адреса
A.B.C.D	Маска префикса
eigrp	Очищает IP-EIGRP
<1–65535>	Номер автономной системы
neighbors	Удалить соседей IP-EIGRP
A.B.C.D	Адрес соседа IP-EIGRP
Ethernet	IEEE 802.3
<0–1>	Номер интерфейса Ethernet
Null	Null-интерфейс
<0–0>	Номер Null-интерфейса
neighbors	Удалить соседей IP-EIGRP
A.B.C.D	Адрес соседа IP-EIGRP
Ethernet	IEEE 802.3

Параметр	Описание
<0–1>	Номер интерфейса Ethernet
Null	Null-интерфейс
<0–0>	Номер Null-интерфейса
nat	Очищает NAT (Network Address Translation)
statistics	Удалить статистику трансляции
translation	Удалить динамическую трансляцию
nhrp	Очищает кэш протокола NHRP (Clear Next Hop Resolution Protocol)
A.B.C.D	Удаляемая сеть назначения
A.B.C.D	Маска сети назначения
ospf	Удаляет команды OSPF
counters	Счетчики OSPF
neighbor	Статистика по соседу для интерфейса или идентификатор соседа
Ethernet	IEEE 802.3
<0–1>	Номер интерфейса Ethernet
Hostname или *A.B.C.D*	Идентификатор соседа
Null	Null-интерфейс
<0–0>	Номер Null-интерфейса
redistribution	Перераспределение маршрутов
prefix-list	Очищает список префиксов
WORD	Имя списка префиксов
redirect	Очищает кэш перенаправления
route	Удаляет записи таблицы маршрутов
*	Удалить все маршруты
A.B.C.D	Сеть назначения удаляемого маршрута
rtp	Удаляет статистику сжатия заголовков RTP/UDP/IP
header-compression	Статистика сжатия заголовков RTP/UDP/IP
Ethernet	IEEE 802.3
<0–1>	Номер интерфейса Ethernet
Null	Null-интерфейс
<0–0>	Номер Null-интерфейса
ipx	Удаляет информацию о Novell/IPX
accounting	Очистить учетную базу данных IPX

Параметр	Описание
checkpoint	Очистить учетную базу контрольных точек IPX
cache	Очистить кэш быстрой коммутации IPX
eigrp	Очистить Novell-EIGRP
route	Удалить запись таблицы маршрутов IPX
neighbors	Удалить соседей Novell-EIGRP
Ethernet	IEEE 802.3
<0–1>	Номер интерфейса Ethernet
N.H.H.H	Адрес соседа Novell-EIGRP
Null	Null-интерфейс
<0–0>	Номер Null-интерфейса
spx-spoof	Очистить ложную (spoof) таблицу IPX SPX
line	Переустанавливает терминальную линию
<0–5>	Номер линии
console	Основная терминальная линия
<0–0>	Номер линии
vty	Виртуальный терминал
<0–4>	Номер линии
logging	Очищает буфер журнала
snapshot	Сбрасывает Snaphot-таймеры
quiet-time	Удаляет время молчания, вводит время активности (со стороны клиента)
Ethernet	IEEE 802.3
<0–1>	Номер интерфейса Ethernet
Null	Null-интерфейс
<0–0>	Номер Null-интерфейса
tcp	Удаляет TCP-соединение или статистику
line	Линия TTY
<0–0>	Номер линии
local	Адрес/порт локального узла
Hostname или *A.B.C.D*	Имя или IP-адрес локального узла
<1–65535>	Локальный порт TCP
remote	Адрес/порт удаленного узла
Hostname или *A.B.C.D*	Имя или IP-адрес удаленного узла
<1–65535>	Удаленный порт TCP
statistics	Статистика протокола TCP

Параметр	Описание
tcb	Адрес TCB
<0x0–0xFFFFFFFF>	Адрес TCB
x25	Удаляет цепи X.25
Ethernet	IEEE 802.3
<0–1>	Номер интерфейса Ethernet
H.H.H	MAC-адрес узла CMNS
xot	Удаляет виртуальную цепь XOT (X.25-Over-TCP)
remote	Показать виртуальные цепи XOT к удаленному узлу
A.B.C.D	IP-адрес узла
<1–65535>	Номер порта
local	Показать виртуальные цепи XOT локального узла
A.B.C.D	IP-адрес узла
<1–65535>	Номер порта

Пример синтаксиса

```
#clear counters
```

clock

Описание Cisco

Управляет системными часами.

Параметры

Параметр	Описание
set	Устанавливает время и дату
hh:mm:ss	Текущее время
<1–31>	День месяца
MONTH	Месяц
<1993–2035>	Год
MONTH	Месяц
<1993–2035>	Год

Пример синтаксиса

```
#clock set 12:00:00 1 January 2000
```

copy

Описание Cisco

Копирует файл.

Опция

Параметр	Описание
/erase	Очистить файловую систему назначения

Параметры «откуда»

Параметр	Описание
flash:	Копировать из файловой системы flash:
flh:	Копировать из файловой системы flh:
ftp:	Копировать из файловой системы ftp:
null:	Копировать из файловой системы null:
nvram:	Копировать из файловой системы nvram:
rcp:	Копировать из файловой системы rcp:
running-config	Копировать из текущей системной конфигурации
startup-config	Копировать из стартовой конфигурации
system:	Копировать из файловой системы system:
tftp:	Копировать из файловой системы tftp:

Параметры «куда»

Параметр	Описание
flash:	Копировать в файловую систему flash:
flh:	Копировать в файловую систему flh:
ftp:	Копировать в файловую систему ftp:
null:	Копировать в файловую систему null:
nvram:	Копировать в файловую систему nvram:
rcp:	Копировать в файловую систему rcp:
running-config	Копировать в текущую системную конфигурацию
startup-config	Копировать в стартовую конфигурацию
system:	Копировать в файловую систему system:
tftp:	Копировать в файловую систему tftp:

Пример синтаксиса

```
#copy running-config startup-config
```

debug

Описание Cisco

Отладочные функции (см. также `undebug`).

Параметры

Параметр	Описание
`aaa`	AAA (Authentication, Authorization, Accounting – аутентификация, авторизация и учет)
`access-expression`	Булево выражение для доступа
`all`	Включить всю отладку
`arp`	IP ARP (Address Resolution Protocol) и HP Probe transactions
`async`	Информация асинхронного интерфейса
`callback`	Активность обратных вызовов
`cdp`	Информация протокола CDP (Certificate Discovery Protocol)
`chat`	Chat scripts activity
`compress`	Трафик COMPRESS
`condition`	Условие
`confmodem`	Конфигурационная база данных модема
`custom-queue`	Произвольные выходные очереди (custom queuing)
`dhcp`	Активность клиента DHCP (Dynamic Host Configuration Protocol)
`dialer`	Набор по требованию
`dnsix`	Информация DNSIX (Department of Defense Intelligence Information System Network Security for Information Exchange)
`domain`	Служба имен доменов
`dxi`	Информация ATM-DXI (Asynchronous Transfer Mode-Data Exchange Interface)
`eigrp`	Информация EIGRP (Enhanced Interior Gateway Routing Protocol)
`entry`	Элементы входной очереди
`ethernet-interface`	События сетевого интерфейса Ethernet
`frame-relay`	Frame Relay
`interface`	Интерфейс
`ip`	Информация IP
`ipx`	Информация Novell/IPX
`lapb`	Транзакции LAPB (Link Access Protocol Balanced)
`list`	Устанавливает интерфейс, список доступа или и то и другое для следующей отладочной команды

Параметр	Описание
llc2	Информация LLC2 Type II
modem	Активация управляющего/рабочего режима модема
nhrp	Протокол NHRP (Next Hop Resolution Protocol)
packet	Журнал неизвестных пакетов
pad	Протокол X.25 PAD (packet assembler/disassembler)
ppp	Информация PPP (Point to Point Protocol)
priority	Приоритет постановки в выходную очередь
radius	Протокол RADIUS (Remote Authentication Dial-In User Service)
rtr	Информация монитора RTR (Response Time Reporter)
serial	Информация последовательного интерфейса
smf	Программный фильтр MAC
snapshot	Snapshot-активность
snmp	Информация SNMP (Simple Network Messaging Protocol)
sntp	Информация SNTP (Simple Network Time Protocol)
spantree	Информация связующего дерева (Spanning tree)
standby	Протокол горячего резервирования
tacacs	Аутентификация и авторизация TACACS (Terminal Access Controller Access Control System)
tbridge	Прозрачный мост
telnet	Входящие соединения Telnet
tftp	Отладка TFTP (Trivial File Transfer Protocol)
tunnel	Общий туннельный интерфейс
vprofile	Информация виртуального профиля
vtemplate	Информация виртуального терминала
x25	Информация X.25, CMNS (Connection Mode Network Service) и XOT (X.25 по TCP)
x28	Режим X.28

delete

Описание Cisco

Удаляет файл.

Параметр

Параметр	Описание
flash:	Файл, подлежащий удалению

Пример синтаксиса

```
#delete flash: aa3424.bin
```

dir

Описание Cisco

Список файлов файловой системы.

Параметры

Параметр	Описание
/all	Вывести все файлы
flash:	Каталог или имя файла
flh:	Каталог или имя файла
null:	Каталог или имя файла
nvram:	Каталог или имя файла
system:	Каталог или имя файла

Пример синтаксиса

```
#dir /all
```

erase

Описание Cisco

Удаляет файловую систему.

Параметры

Параметр	Описание
flash:	Файловая система, подлежащая удалению
nvram:	Файловая система, подлежащая удалению
startup-config	Удалить содержимое конфигурационной памяти

Пример синтаксиса

```
#erase startup-config
```

more

Описание Cisco

Показывает содержимое файла.

Опции

Параметр	Описание
/ascii	Показать двоичный файл в ASCII-кодировке

Параметр	Описание
/binary	Переключить отображение в шестнадцатеричный/текстовый формат
/ebcdic	Показать двоичный файл в EBCDIC-кодировке

Параметры

Параметр	Описание
flash:	Отображаемый файл
flh:	Отображаемый файл
ftp:	Отображаемый файл
null:	Отображаемый файл
nvram:	Отображаемый файл
rcp:	Отображаемый файл
system:	Отображаемый файл
tftp:	Отображаемый файл

Пример синтаксиса

```
#more /ascii system:
```

pwd

Описание Cisco

Показывает текущий каталог.

Пример синтаксиса

```
#pwd
```

reload

Описание Cisco

Выполняет останов и холодный рестарт.

Параметры

Параметр	Описание
LINE	Причина перезагрузки
at	Перезагрузить в указанное время/дату
hh:mm	Время перезагрузки (чч:мм)
<1–31>	День месяца
LINE	Причина перезагрузки
MONTH	Месяц

Параметр	Описание
cancel	Отменить запланированную перезагрузку
in	Перезагрузить через заданный интервал времени
mmm или hhh:mm	Задержка перед перезагрузкой
LINE	Причина перезагрузки

Пример синтаксиса

```
#reload in 30 for maintenance
```

rsh

Описание Cisco

Выполняет удаленную команду.

Параметры

Параметр	Описание
WORD	IP-адрес или имя rsh-сервера
/user	Имя удаленного пользователя
LINE	Команда, которая должна быть выполнена удаленно

Пример синтаксиса

```
#rsh RouterA reload
```

setup

Описание Cisco

Позволяет реконфигурировать маршрутизатор.

Пример синтаксиса

```
#setup
```

test

Описание Cisco

Проверяет подсистемы, память и интерфейсы.

Параметры

Параметр	Описание
eigrp	Тестовые команды IPX EIGRP
<1–65535>	Номер АС
1local	Статус соседей 1
<1–FFFFFFFE>	IPX-адрес

Параметр	Описание
1successor	Статус соседей 3
<1–FFFFFFFE>	IPX-адрес
2local	Статусы соседей 1–2
<1–FFFFFFFE>	IPX-адрес
2successor	Статусы соседей 3–2
<1–FFFFFFFE>	IPX-адрес
3local	Статусы соседей 1–0
<1–FFFFFFFE>	IPX-адрес
4local	Статусы соседей 1–0–2
<1–FFFFFFFE>	IPX-адрес
5local	Статусы соседей 1–0–FC fail–1
<1–FFFFFFFE>	IPX-адрес
6local	Статусы соседей 1–2–FC fail–3
<1–FFFFFFFE>	IPX-адрес
ack	Переключает быстрое подтверждение получения EIGRP
delete	Удаляет фальшивую запись топологической таблицы
ifs	IFS TEST-код
appn	APPN-команды
read	Читать APPN-файл
Hostname или *A.B.C.D*	Адрес назначения
LINE	Только имя файла
write	Записать APPN-файл
Hostname или *A.B.C.D*	Адрес назначения
LINE	Только имя файла
boot	Разобрать командную строку команды начальной загрузки boot
LINE	Остаток команды начальной загрузки boot
defaults	Показать загрузочные файлы по умолчанию
show	Показывает команды тестирования
hidden	Включает режим отображения скрытых файловых систем и файлов
slot	Создает дамп для слотов в случае отказа
<0–32>	Номер слота
WORD	URL дампа слота

Параметр	Описание
WORD	URL дампа слота

Пример синтаксиса

```
#test ifs appn read RouterA test_file
```

verify

Описание Cisco

Проверяет файл.

Параметр

Параметр	Описание
flash:	Файл для проверки

Пример синтаксиса

```
#verify flash: aaa8764.bin
```

write

Описание Cisco

Записывает рабочую конфигурацию в память, на сетевой сервер или терминал.

Параметры

Параметр	Описание
erase	Стереть энергонезависимую (флэш) память
memory	Запись в энергонезависимую память
network	Запись на сетевой TFTP-сервер
flash:	URL файла назначения
ftp:	URL файла назначения
null:	URL файла назначения
nvram:	URL файла назначения
rcp:	URL файла назначения
system:	URL файла назначения
tftp:	URL файла назначения
terminal	Запись на терминал

Пример синтаксиса

```
#write memory
```

Алфавитный указатель

Специальные символы

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

А

Б

В

Г

Д

З

И

К

Л

М

Н

О

П

P

C

Т

У

Ф

Ш

Э