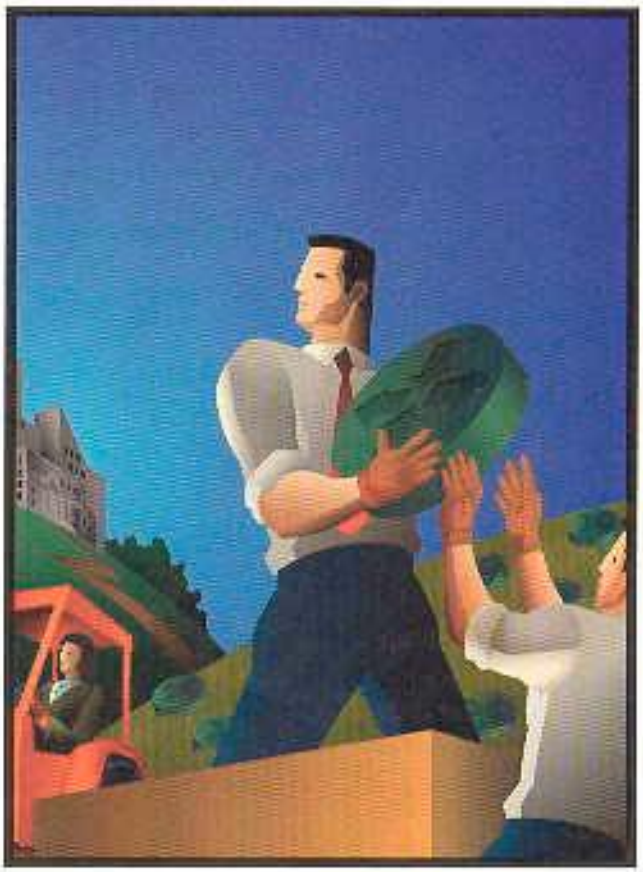
Аллан Леонард
Брюс Пински,
CCIE #1045



Конфигурирование маршрутизаторов Cisco

2-е издание

Практическое руководство по конфигурированию ОС IOS®

Cisco Systems
Cisco Press

ББК 32.973.26-018.2.75
Л42 УДК 681.3.07

Издательский дом "Вильяме"

Зав. редакцией *СИ. Тригуб* Перевод с английского и редакция *А.А. Голубченко*

По общим вопросам обращайтесь в Издательский дом "Вильяме" по адресу: info@williamspublishing.com, http://www.williamspublishing.com

Леинванд, Аллан, **Пински,** Брюс.

Л42 Конфигурирование маршрутизаторов Cisco, 2-е изд. : Пер. с англ. — М. : Издательский дом "Вильяме", 2001. — 368 с. : ил. — Парад, тит. англ

ISBN 5-8459-0219-3 (рус.)

Эта книга, написанная специалистами-практиками, посвящена изучению основ конфигурирования операционной системы IOS, под управлением которой работают все устройства межсетевого взаимодействия компании Cisco Systems, Inc. Материал подается на примере типовой сети вымышленной компании ZIP. Книга будет полезна специалистам, работающим в области сетевых технологий, которые впервые решают задачу построения работоспособной сети, использующей мосты, коммутаторы и маршрутизаторы компании Cisco.

БК 32.973.26-018.2.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства Cisco Press.

Authorized translation from the English language edition published by Cisco Press, Copyright © 2001 All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Russian language edition published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2001

ISBN 5-8459-0219-3 (рус)
ISBN 1-57870-241-0 (англ.)

© Издательский дом "Вильяме", 2001
© Cisco Press 2001

Оглавление

Содержание

Глава 3. Основы интерфейсов устройств Cisco

Базовое конфигурирование интерфейсов
- Команда show interfaces
- Команда encapsulation
- Команда shutdown
- Команда description

Технологии локальных сетей
- Технологии локальной сети Ethernet и IEEE802.3
- Технология Fast Ethernet
- Подкоманды конфигурирования интерфейсов Fast Ethernet и Ethernet
- Технология Gigabit Ethernet
- Технология Token Ring
- Технология FDDI

Технологии глобальных сетей и технологии взаимодействия по коммутируемым каналам связи
- Протокол HDLC
- Протокол Point-to Point
- Протокол X.25
- Подкоманды конфигурирования интерфейса X 25
- Протокол Frame Relay
- Подкоманды конфигурирования интерфейса Frame Relay
- Технология Asynchronous Transfer Mode
- Подкоманды конфигурирования интерфейсов ATM
- Технология Digital Subscriber Line
- Технология ISDN
- Подкоманды конфигурирования интерфейса ISDN

Резюме

Дополнительная литература

Глава 4. Основы TCP/IP

TCP/1P-адресация
- Структура адреса

Конфигурирование IP-адресов
- Конфигурирование интерфейса локальной сети
- Конфигурирование интерфейса глобальной сети
- Проверка конфигурации IP-адресов

Конфигурирование IP-маршрутизации
- Команды конфигурирования IP-маршрутизации
- Проверка конфигурации IP-маршрутизации
- Конфигурирование протоколов IP-маршрутизации
- Конфигурирование протокола маршрутной информации Routing Information Protocol
- Конфигурирование протокола внутренней маршрутизации между шлюзами компании Cisco Systems Interior Gateway Routing Protocol
- Конфигурирование открытого протокола выбора первым наикратчайшего пути Open Shortest Path First Protocol
- Конфигурирование усовершенствованного IP-протокола IGRP компании Cisco
- Конфигурирование протокола пограничной маршрутизации Border

Глава 6. Основы IPX

Система адресации и структура адреса в протоколе IPX

Конфигурирование IPX-адресов
- Конфигурирование интерфейсов локальной сети
- Конфигурирование интерфейсов глобальной сети
- Проверка конфигурации IPX-адресов

Конфигурирование IPX-маршрутизации
- Команды конфигурирования IPX-маршрутизации
- Конфигурирование статической маршрутизации
- Проверка конфигурации IPX-маршрутизации

Конфигурирование протоколов маршрутизации, работающих с протоколом IPX
- Протокол SAP
- Фильтры сообщений протокола SAP
- Конфигурирование протокола IPX RIP
- Конфигурирование протокола NLSP
- Конфигурирование протокола IPX EIGRP

Конфигурирование фильтрации в протоколе IPX с применением списков доступа
- Задание списков доступа
- Наложение списков доступа

Конфигурирование основных служб удаленного доступа по коммутируемым каналам связи протокола IPX

Верификация взаимодействия в сети с протоколом IPX и устранение неполадок

Конфигурирование переадресации IPX-пакетов типа

Резюме

Дополнительная литература

Глава 7. Основы администрирования и управления

Основы управления доступом
- Подключение к виртуальному терминалу с использованием протокола
- Telnet и оболочки SSH
- Активация SSH-сервера
- Проверка конфигурации протокола SSH
- Защита порта консоли и виртуальных терминалов
- Активация AAA-служб
- Протокол RADIUS
- Протокол TACACS+
- Сравнение протоколов RADIUS и TACACS+

Основы предотвращения атак
- TCP-перехват
- Одноадресная пересылка по обратному пути

Основы управления сетью

Основы управления временем
- Конфигурирование даты и времени вручную
- Протокол сетевого времени
- Простой протокол сетевого времени

Резюме

Об авторах

Аллан Леинванд (Allan Leinwand) — главный технолог и вице-президент по конструированию компании Telegis Networks, Inc. Ранее, занимая аналогичный пост в компании Digital Island, Inc., отвечал за техническое развитие глобальной сети компании и стратегию распространения информации. До этого он работал менеджером по техническому консалтингу в компании Cisco Systems, Inc. и решал вопросы, связанные с проектированием глобальных сетей для клиентов компании. Свою степень бакалавра по вычислительной технике Аллан получил в 1988 году в Университете штата Колорадо (г. Боулдер) и с того времени работает в области технологий межсетевого взаимодействия крупных корпораций. Он также ведет курс компьютерных сетей для аспирантов в Университете г. Беркли (штат Калифорния). Леинванд опубликовал множество статей, посвященных управлению и проектированию сетей. Кроме того, он является одним из соавторов выпущенной издательством Addison-Wesley монографии Network Management: Practical Perspective, Second Edition.

Брюс Пински (Bruce Pinsky) — сертифицированный Cisco эксперт по межсетевому взаимодействию (сертификат № 1045), вице-президент компании Telegis Networks, Inc. по конструированию продуктов и сетевым инфраструктурам. Ранее, будучи ведущим специалистом по информатизации, вице-президентом по техническим вопросам и старшим специалистом по стратегии развития сетей в компании Digital Island, Inc., отвечал за развитие и внедрение корпоративных инфраструктурных технологий и перспективные исследования в области технологий. До компании Digital Island Брюс работал старшим технологом по поддержке межсетевых комплексов компании Cisco и решал возникавшие у заказчиков сложные технические проблемы. До и после получения степени бакалавра по вычислительной технике в Калифорнийском государственном университете (г. Хайвард) в 1988 году он занимался технологиями межсетевых взаимодействий и системной интеграцией для больших корпораций и консалтинговых фирм. Являясь одним из первых сертифицированных Cisco экспертов по сетевым комплексам, Брюс обладает большим опытом в таких областях, как поиск и устранение неисправностей в сетях, анализ протоколов, проектирование и конфигурирование сетей, а также в области серверных операционных систем и операционных систем рабочих станций. Он регулярно читает курсы по конфигурированию, проектированию, поиску и устранению неисправностей в сетях, а также является одним из авторов запатентованной технологии маршрутизации.

О технических рецензентах

Генри Бенджамин (Henry Benjamin) — обладатель сертификатов Cisco по сетевым комплексам, сетям и проектированию, бакалавр технических наук, является сертифицированным Cisco экспертом в области сетевых комплексов и инженером по информационным сетям компании Cisco. У него более чем десятилетний опыт работы с сетями на основе устройств компании Cisco, включая планирование, проектирование и реализацию больших IP-сетей, использующих протоколы Динамической маршрутизации IGRP, EIGRP и OSPF. В последний год Генри сосредоточил свои усилия на проектировании архитектуры и реализации внутренних сетей компании Cisco в Австралии и в азиатско-тихоокеанском регионе. Бенджамин написал книгу, посвященную сдаче письменного экзамена на звание сертифицированного Cisco эксперта по сетевым комплексам. Генри имеет степень бакалавра технических наук Сиднейского университета.

Кевин Бюргесс (Kevin Burgess) в течение 10 лет занимается вопросами проектирования, анализа и технического сопровождения сетей. Последние пять лет он в качестве инженера по сетям компании EDS принимал участие в различных проектах на территории Канады. Кевин является обладателем ряда сертификатов от компаний Novell и Cisco и в настоящее время работает над получением звания сертифицированного Cisco эксперта по сетевым комплексам.

Андре Пари-Хафф (Andre Paree-Huf!) — сертифицированный специалист по ряду программ, занимается вычислительной техникой более 8 лет. В настоящее время работает в Североамериканском центре поддержки клиентов компании Compaq Computer Corporation в Колорадо-Спрингс (штат Колорадо) инженером III категории по технической поддержке сетей. Андре занимается поиском и устранением неисправностей в сетевой аппаратуре, специализируясь на 2 и 3 уровнях модели OSI. Он является соавтором четырех технических руководств, посвященных сетям, и был техническим редактором многих других изданий. В настоящее время Андре готовится получить звание сертифицированного Cisco эксперта по сетевым комплексам.

Дейв Самтер (Dave Sumter) — сертифицированный Cisco эксперт по сетевым комплексам (сертификат № 4942). В сетевой промышленности Дейв работает около пяти лет, а последние три года полностью сосредоточен на решениях компании Cisco. Работает в компании Cisco Systems, Inc. в ЮАР. В настоящее время его обязанности связаны с проектированием крупномасштабных кампусов и глобальных сетей для корпоративных и правительственных клиентов в ЮАР. Кроме того, Самтер постоянно обучает партнеров компании и принимает экзамены у кандидатов на звание сертифицированного Cisco эксперта по сетевым комплексам в Лаборатории маршрутизации и коммутирования центра подготовки экспертов по сетевым комплексам в ЮАР.

Майкл Труетт (Michael Truett) — сертифицированный Cisco специалист в области сетей, инженер по сетям в крупной организации, специализирующейся на передаче речи по протоколу IP (VoIP). В настоящее время работает над получением звания сертифицированного Cisco специалиста по проектированию и эксперта по сетевым комплексам. Майкл занимается проектированием, поиском и устранением неполадок в крупных сетях, работающих в различных средах, включая Frame Relay и спутниковые каналы связи. В свободное время Майкл также ведет несколько классов по маршрутизаторам и коммутаторам компании Cisco.

Посвящения

Дллан Леинванд хотел бы посвятить эту книгу своей семье и друзьям, которые оказывали ему постоянную поддержку, давали советы, ободряли и помогали лучше разобраться в технических вопросах.

Брюс Пински хотел бы поблагодарить всех своих друзей и семью, которые помогли сделать эту книгу реальностью Особая признательность — жене Пауле и сыновьям Эрику и Кайлу за их неустанную поддержку на протяжении многих ночей и выходных дней, отданных завершению этой книги.

Введение

Компания Cisco Systems, Inc. является ведущим в мире поставщиком аппаратного и программного обеспечения для межсетевого взаимодействия. Cisco ежегодно устанавливает более 100 000 устройств, которые работают как в частных сетях, так и в сетях общего пользования. На момент написания книги эти устройства обслуживали более 80 процентов трафика сети Internet. Настоящая книга призвана помочь новым пользователям продуктов компании Cisco освоить основы администрирования своих устройств межсетевого взаимодействия.

В состав этих устройств входит разработанная компанией специальная операционная система — межсетевая операционная система Cisco (Cisco Internetwork Operating System — IOS). ОС IOS представляет собой сложную операционную систему реального времени, состоящую из нескольких подсистем и имеющую десятки тысяч возможных параметров конфигурирования. Используя приводимые в хронологическом порядке простые описания и практические примеры, авторы основное внимание уделяют ОС IOS с точки зрения конфигурирования, эксплуатации и сопровождения устройств межсетевого взаимодействия. Кроме общих аспектов ОС IOS, здесь также рассматриваются три наиболее популярных сетевых протокола: протокол управления передачей данных/межсетевой протокол TCP/IP (Transmission Control Protocol/Internet Protocol), протокол межсетевого обмена пакетами IPX компании Novell (Internetwork Packet Exchange) и протокол AppleTalk компании Apple Computer, Inc.

Цели книги

Основная цель данной книги состоит в том, чтобы сделать ОС IOS для пользователей-новичков легкой в конфигурировании, работе и сопровождении. Поставляемая с каждым изделием компании Cisco документация на ОС IOS занимает несколько компакт-дисков и дает полное представление о каждой команде со всеми соответствующими опциями. Однако зачастую документация пугает и запутывает людей, когда они пытаются с ее помощью сконфигурировать изделие на выполнение базовых функций межсетевого взаимодействия.

Настоящая книга, объясняя общеупотребительные команды и опции ОС IOS, является дополнением к имеющейся технической документации. С помощью многочисленных примеров, иллюстраций и выводимых ОС IOS сообщений в ней поясняется применение ОС IOS для различных пользователей и конфигураций межсетевого взаимодействия. Весь материал книги построен на рассмотрении примера создания сетевого комплекса вымышленной компании Zoom Integrated Products (ZIP). Все вводимые команды конфигурирования и стратегии иллюстрируются конкретным примером устройства или топологии сети компании ZIP.

На кого рассчитана эта книга

Данная книга предназначается любому начинающему пользователю ОС IOS. Однако благодаря множеству примеров и советов по использованию общих функций ОС IOS и более искушенный пользователь найдет в ней много ценного.

Предполагается, что читатель имеет некоторое общее представление о различных типах оборудования для межсетевого взаимодействия — о концентраторах, мостах, коммутаторах и маршрутизаторах. Детальное рассмотрение работы оборудования этих типов выходит за рамки настоящей книги, но краткое их описание в преломлении к ОС 1OS приводится. Для читателей, желающих подробно изучить протоколы TCP/IP, AppleTalk и IPX, в конце каждой главы даются ссылки на соответствующие издания. Другими словами, вместо того, чтобы дублировать существующую литературу по конкретному оборудованию для межсетевого взаимодействия и протоколам, настоящая книга сконцентрирована на вопросах применения этих технологий в изделиях, которые работают с ОС IOS компании Cisco.

Структура книги

Глава 1, "Введение в технологии межсетевого взаимодействия", посвящена описанию эталонной модели взаимодействия открытых систем (модель OSI) и кратком} описанию общих типов межсетевых устройств, которыми (в контексте данной книги' являются мосты, коммутаторы и маршрутизаторы. Завершается глава описанием взятого в качестве примера сетевого комплекса вымышленной компании Zoom Integratec Products (ZIP).

В главе 2, "Основы конфигурирования устройств", приводится базовая информация, которую необходимо знать об устройстве компании Cisco, начиная с его конфигурирования по извлечении из упаковки. Рассматриваются доступ к порт; консоли, базовое конфигурирование с терминала, режим задания начальных установок ОС IOS, средства контекстной помощи, привилегированный режим и структура команд конфигурирования. В этой главе также объясняются некоторые физические характеристики устройств, например, обращение к оперативной памяти, со хранение информации о конфигурации в энергонезависимой памяти и пересылка образа ОС IOS во флэш-память.

В главе 3, "Основы интерфейсов устройств Cisco", приводятся необходимые данные о типах сетевых интерфейсов, которыми обладают устройства компании Ciscc Глава знакомит с каждым из следующих типов интерфейсов: Ethernet, Fast Ethernet Gigabit Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI), High-Level Data Link Control (HDLC), Point-to-Point Protocol (PPP), X.25, Frame Relay, Asynchronou Transfer Mode (ATM), Digital Subscriber Loop (DSL) и Integrated Services Digital Network (ISDN). Глава также содержит объяснения применения команд ОС IOS для проверю статуса и работоспособности интерфейса.

В главе 4, "Основы TCP/IP", объясняются главные концептуальные особенности протокола IP: образование подсетей и маршрутизация. В главе также показываете; как использовать ОС IOS для конфигурирования IP-адресов, IP-маршрутов, протоке лов IP-маршрутизации (RIP, IGRP, OSPF, EIGRP и BGP4), средств защиты IP-сети удаленного IP-доступа по коммутируемым каналам связи. Кроме того, приводите описание других деталей конфигурирования ОС IOS для работы с протоколом II включая конфигурирование службы имен доменов (DNS), переадресацию Широковещательных IP-пакетов, DHCP-службы и резервное дублирование.

Глава 5, "Основы AppleTalk", начинается с обзора группы протоколов, работающих в рамках протокола AppleTalk. После этого в главе рассматривается конфигурирование в ОС IOS диапазонов кабелей, зон, протоколов маршрутизации (RTMP), средств защиты сети AppleTalk и удаленного AppleTalk-доступа по коммутируемым м каналам связи.

В начале главы 6, "Основы IPX", дается обзор элементов протокола IPX компании номеров сетей, протокола объявления служб Service Advertising (SAP) и маршрутизации. Затем рассматривается использование ОС IOS для конфигурирования IPX-адресов, различных методов инкапсуляции в локальных сетях, маршрутов, протоколов маршрутизации (RIP, NLSP и EIGRP), средств защиты в IPX-сетях и удаленного IPX-доступа по коммутируемым каналам связи.

Глава 7, "Основы администрирования и управления", объясняет другие конфигурируемые элементы ОС IOS, о которых пользователю необходимо знать. Эти элементы включают управление доступом, использование защищенной оболочки Secure Shell для доступа к устройству, работающему под управлением ОС IOS, протоколирование сообщений, протоколы управления сетью, управление часами/календарем. В главе также показывается, как конфигурировать простой протокол управления сетью Network Management Protocol, систему управления доступом на основе контроллера доступа к терминалу Terminal Access Controller System TACACS+), службу удаленной аутентификации пользователей по телефонным линиям Remote Authentication Dial-In User Service (RADIUS) и протокол сетевого времени Network Time Protocol (NTP).

В главе 8, "Полная конфигурация ОС IOS для сети компании ZIP", приводится полная конфигурация ОС IOS для всей взятой в качестве примера сети компании ZIP. Эта глава сводит воедино все примеры конфигурирования, использованные в тексте книги.

Особенности и элементы книги

В этой книге для того, чтобы подчеркнуть ключевые концептуальные моменты, используется ряд элементов и условных обозначений. Так, для иллюстрации конфигурирования применяется единый пример сети на протяжении всей книги. Для удобства схема сети компании ZIP размещена на внутренней (второй) странице обложки.

Естественно, что код конфигурации является центральным элементом этой книги. Фрагменты кода для облегчения идентификации представлены отличающимся шрифтом. Вводимые пользователем элементы кода и его отдельные термины выделяются жирным шрифтом, кроме того, в книге используются следующие элементы.

Примечания. Комментарии на полях страницы, которые связаны с обсуждаемой темой, но могут быть пропущены без ущерба для понимания сути вопроса и целостности материала.

Советы. Комментарии на полях страницы, в которых описывается эффективный, короткий или оптимальный способ использования технологии.

Дополнительные справки. Тексты на полях страницы, указывающие местонахождение источников дополнительной информации по обсуждаемой теме.

Сводные таблицы команд. Справочный материал и повторение наиболее важных новых команд и синтаксиса, введенных в главе, размещаются в конце соответствующих глав.

Краткая история компании Cisco Systems

У истоков компании Cisco Systems стояли Лен и Сэнди Босак (Len and Sandy Bosack), муж и жена, работавшие на различных факультетах Станфордского университета. Им надо было сделать так, чтобы их компьютерные системы могли общаться друг с другом. Решая эту проблему, они построили устройство, названное шлюзовым сервером. Шлюзовой сервер помогал машинам двух факультетов Станфордского университета обмениваться данными с помощью межсетевого протокола Internet Protocol (IP).

Вскоре после этого достижения они решили испытать счастье и создать коммерческий вариант шлюзового сервера. Первым помещением компании Cisco, где осуществлялись разработка и производство, была гостиная Босаков. В 1984 году была основана компания cisco Systems, Inc., и новая эра в межсетевом взаимодействии началась.

Обратите внимание на строчную букву "c" в первоначальном названии компании; существует множество слухов и объяснений ее происхождения. Она интерпретировалась как попытка смутить редакторов, когда те начинают предложение с названия компании, и как ошибка юристов, предлагавших проект названия. В качестве причины назывались и порванный лист бумаги, на котором изначально стояло название San Francisco Systems, и просто стремление выделиться. Мы не будем открывать здесь правду, поскольку предпочитаем сохранить тайну — пусть каждый выберет себе вариант по вкусу. В 1992 году название компании было официально изменено на Cisco Systems, Inc. Переход к прописной "C" был встречен ортодоксами cisco с некоторыми колебаниями, однако сегодня большинством используется название Cisco Systems, Inc., исключая, может быть, несгибаемых инженеров времен cisco Systems.

Первым шлюзовым продуктом от Cisco был так называемый новейший шлюзовой сервер (Advanced Gateway Server — AGS), за которым вскоре последовали шлюзовой сервер среднего диапазона (Mid-Range Gateway Server — MGS), малогабаритный шлюзовой сервер (Compact Gateway Server — MGS), интегрированный шлюзовой сервер (Integrated Gateway Server — IGS) и новейший шлюзовой сервер плюс (AGS+). Теперь эти продукты — история компании. Следующее поколение продуктов начало появляться в 1993 году, начиная с маршрутизаторов серии Cisco 4000, за которыми быстро последовали серии маршрутизаторов Cisco 7000, 2000 и 3000. Семейство продуктов Cisco продолжает развиваться и сегодня. Следуя традиции использовать номера продуктов вместо названий, появились маршрутизаторы Cisco 12000 и коммутаторы Catalyst 6500.

С середины 1990-х Cisco расширила ассортимент, включив в него, кроме маршрутизаторов, и другие продукты для межсетевого взаимодействия: коммутаторы для локальных сетей, ATM-коммутаторы, продукты для глобальных сетей, средства связи с мэйнфреймами IBM и многое другое.

Из-за разнообразия продуктов компании Cisco, наследуемой сложности ОС IOS и широкого роста количества реализаций сетевых комплексов сетевые проектировщики и менеджеры начали захлебываться в потоке информации, необходимой для конфигурирования сети с устройствами Cisco. Данная книга призвана выделить из огромного количества имеющейся информации и документации главные положения. Авторы хотели сделать превосходные продукты Cisco столь же доступными новичкам, сколь они доступны ветеранам использования ОС IOS.

Глава 1

Введение в технологии межсетевого взаимодействия

Ключевые темы этой главы

Эталонная модель взаимодействия открытых систем (модель OSI). Обзор семи уровней задач, обеспечивающих работоспособность коммуникационных систем.

Типы устройств межсетевого взаимодействия. Основные устройства сетевых комплексов: мосты, коммутаторы, маршрутизаторы и серверы доступа.

Пример сетевого комплекса. Топология конкретного сетевого комплекса, используемого в данной книге в качестве примера.

Данная глава призвана помочь начать изучение основ технологий межсетевого взаимодействия Изучение и понимание этих основ является первым шагом в освоении операционной системы Internetwork Operating Systems компании Cisco (ОС IOS). ОС IOS обеспечивает тот уровень "интеллекта", который необходим продуктам компании Cisco для решения различных задач межсетевого взаимодействия. ОС IOS — это операционная система со своим интерфейсом пользователя, набором команд, собственным синтаксисом конфигурирования и т д. Для устройств компании Cisco ОС IOS является тем же, чем для IBM-совместимых компьютеров является ОС Windows 2000. Она работает на всех продуктах компании Cisco, речь о которых пойдет в этой книге.

Для того чтобы разобраться в сложных деталях ОС IOS, необходимо четко усвоить материал данной главы, посвященный описанию принципов межсетевого взаимодействия. Термин *межсетевое взаимодействие* используется для описания группы протоколов и устройств, которые взаимодействуют в сетях передачи данных. Настоящая глава позволит понять основы предмета. Это не означает, что в ней представлены все данные для изучения межсетевых взаимодействий (объем информации, необходимый для полного освещения темы, настолько велик, что для него потребовалось бы несколько книг). Вниманию тех читателей, которые хотят больше узнать о технологии межсетевого взаимодействия, в конце настоящей главы представлен раздел "Дополнительная литература".

Изучив данную главу, читатель уже будет на "ты" с моделью OSI и получит общее представление о функционировании мостов, коммутаторов, маршрутизаторов и серверов доступа. В главе 2, "Основы конфигурирования устройств", приводятся основные сведения по конфигурированию устройств компании Cisco.

Эталонная модель взаимодействия открытых систем

Эталонная модель взаимодействия открытых систем (Open System Interconnection reference model) является главным принципом взаимодействия в сетях, который необходимо усвоить, чтобы понять, как работают устройства компании Cisco. Модель OSI — это семиуровневая архитектурная модель, разработанная Международной организацией по стандартизации (International Organization for Standardization — ISO) и Международным телекоммуникационным союзом (International Telecommunications Union-Telecommunications — ITU-T). Эта модель является универсальным средством описания принципов функционирования сетей, облегчающим изучение и понимание их конечными пользователями. Эталонная модель позволяет придать структурированный вид различным по сложности функциям в коммуникационном программном обеспечении. Для создания коммуникационного программного обеспечения следует гарантировать взаимодействие разнотипных приложений в условиях различных стратегий передачи данных, а также учет свойств физических сетей. Без этой модели, без стандартной структуры написание, изменение и последующая поддержка коммуникационного программного обеспечения могут быть чрезвычайно затруднены.

Примечание

ISO — это международная организация по сотрудничеству в сфере технологических разработок, в частности, в телекоммуникационной области. ITU-T является всемирной организацией, которая разрабатывает международные стандарты для всех видов связи — как цифровой, так и аналоговой. ITU-T отвечает за телекоммуникационные стандарты.

Модель OSI состоит из семи различных уровней. Каждый уровень отвечает за отдельный участок в работе коммуникационных систем. Уровень выполняет свою задачу в соответствии с набором правил, называемым *протоколом.* Кроме того, каждый уровень модели предоставляет набор служб для других уровней. Ниже описаны семь уровней модели OSI: уровень приложений, уровень представлений, сеансовый, транспортный, сетевой, канальный и физический уровни. Структура уровней показана на рис. 1.1. В последующих разделах кратко описаны все семь уровней, начиная с уровня приложений.

Рис. 1.1. Эталонная модель OSI содержит семь уровней

Уровень приложений

Уровень приложений предоставляет интерфейс с коммуникационной системой, с которым непосредственно работает пользователь. Сегодня в сетевой среде используется множество широко распространенных приложений: электронная почта, Web-броузеры, программы передачи файлов по протоколу FTP и др. В качестве примера работы на седьмом уровне модели OSI можно привести процесс загрузки Web-броузером файла из сервера сети Internet. Web-броузер и сервер Internet являются одноранговыми сторонами уровня приложений, которые непосредственно взаимодействуют друг с другом в процессе извлечения и передачи документа. Взаимодействующие стороны не подозревают о наличии остальных шести уровней модели OSI, которые участвуют в процессе обмена информацией.

Уровень представлений

Задача уровня представлений состоит в преобразовании синтаксиса данных, передаваемых между двумя общающимися приложениями. Уровень представлений обеспечивает механизм для передачи между приложениями данных в нужном виде. Многие пользователи полагают, что примером уровня представлений может служить среда рабочего стола компьютера, например, одинаковый вид и взаимодействие всех приложений в компьютерах компании Apple Computer, Inc. В действительности, это не имеет никакого отношения к уровню представлений, а просто ряд приложений используют общий интерфейс программирования. В качестве примера уровня представлений можно назвать часто используемую сегодня систему обозначений для описания абстрактного синтаксиса (Abstract Syntax Notation One — ASN.l). Эта система применяется в таких протоколах, как простой протокол управления сетью (Simple Network Management Protocol — SNMP), в котором с ее помощью создается представление структуры объектов в базах данных управления сетью.

Сеансовый уровень

Сеансовый уровень позволяет двум приложениям синхронизировать связь и обмениваться данными. На этом уровне обмен между двумя системами делится на диалоговые блоки и обеспечиваются основные и вспомогательные точки синхронизации этого обмена. Например, для того, чтобы транзакция в большой распределенной базе данных выполнялась с одинаковой скоростью во всех системах, могут использоваться протоколы сеансового уровня.

Транспортный уровень

Четвертый, транспортный уровень, отвечает за передачу данных между сеансовыми уровнями. Существует множество классов протоколов транспортного уровня, начиная от протоколов, предоставляющих только основные транспортные функции (например, функции передачи данных без подтверждения приема), и заканчивая протоколами, которые гарантируют доставку в пункт назначения нескольких пакетов данных в надлежащей последовательности, мультиплексируют несколько потоков данных, обеспечивают механизм управления потоками данных и гарантируют достоверность принятых данных.

Некоторые протоколы сетевого уровня, называемые *протоколами без установки соединения*, не гарантируют, что данные доставляются по назначению в том порядке, в котором они были посланы устройством-источником. Некоторые транспортные уровни справляются с этим, собирая данные в нужной последовательности до передачи их на сеансовый уровень. *Мультиплексирование* (multiplexing) данных означает, что транспортный уровень способен одновременно обрабатывать несколько потоков данных (потоки могут поступать и от различных приложений) между двумя системами. *Механизм управления потоком данных* — это механизм, позволяющий регулировать количество данных, передаваемых от одной системы к другой. Протоколы транспортного уровня часто имеют функцию контроля доставки данных, заставляя принимающую данные систему отправлять подтверждения передающей стороне о приеме данных.

В этой книге рассматриваются три широко распространенных протокола транспортного уровня: протокол управления передачей данных (Transmission Control Protocol — TCP), который используется в сети Internet, протокол потокового обмена пакетами данных фирмы

Novell (Streams Packet Exchange — SPX) и транспортный протокол AppleTalk фирмы Apple (AppleTalk Transport Protocol — ATP).

Сетевой уровень

Сетевой уровень, который перенаправляет данные от одной системы к другой, позволяет выполнять адресацию при межсетевом взаимодействии. Межсетевой протокол (Internet Protocol — IP) определяет принципы присвоения адресов в глобальной сети Internet; протокол компании Novell определяет специфические принципы присвоения адресов для систем, использующих протокол межсетевого обмена пакетами (Internetwork Packet Exchange — IPX) в рамках архитектуры клиент-сервер; и, наконец, протокол AppleTalk компании Apple использует для обмена данными между системами Apple на сетевом уровне протокол доставки дейтаграмм (Datagram Delivery Protocol — DDP) и соответствующую ему систему адресации. В следующих главах будет рассмотрена специфика каждого из этих типов адресации на сетевом уровне.

Протоколы сетевого уровня маршрутизируют данные от источника к получателю и могут быть разделены на два класса: протоколы с установкой соединения и без него. Описать работу протоколов с установкой соединения можно на примере работы обычного телефона. Протоколы этого класса начинают передачу данных с вызова или установки маршрута следования пакетов от источника к получателю. После чего начинают последовательную передачу данных и затем по окончании передачи разрывают связь. Протоколы без установки соединения, которые посылают данные, содержащие полную адресную информацию в каждом пакете, работают аналогично почтовой системе. Каждое письмо или пакет содержит адрес отправителя и получателя. Далее каждый промежуточный почтамт или сетевое устройство считывает адресную информацию и принимает решение о маршрутизации данных. Письмо или пакет данных передается от одного промежуточного устройства к другому до тех пор, пока не будет доставлено получателю. Протоколы без установки соединения не гарантируют поступление информации получателю в том порядке, в котором она была отправлена. За установку данных в соответствующем порядке при использовании сетевых протоколов без установки соединения отвечают транспортные протоколы.

Канальный уровень

Уровень 2 (канальный уровень), отвечая за взаимодействие между физическим и сетевым уровнями, позволяет четко управлять передачей данных по сети. Ethernet, Fast Ethernet, Token Ring, Frame Relay и ATM (Asynchronous Transfer Mode — асинхронный режим передачи данных) — самые распространенные сегодня протоколы второго, канального уровня. Принципы присвоения адресов для канального уровня отличаются от таких принципов для сетевого уровня. Адресация канального уровня является уникальной для каждого логического сегмента канала передачи данных, в то время как адресация сетевого уровня используется во всей сети.

Физический уровень

Первый уровень модели OSI — это физический уровень. К физическому уровню относятся физические, электрические и механические интерфейсы между двумя системами. Физический уровень определяет такие свойства среды сети передачи данных как оптоволокно, витая пара, коаксиальный кабель, спутниковый канал передач данных и т.п. Стандартными типами сетевых интерфейсов, относящимися к физическому уровню, являются: V.35, RS-232C, RJ-11, RJ-45, разъемы AUI и BMC.

Примечание

Зачастую к семи уровням модели OSI добавляют восьмой, или *политический* уровень. Несмотря на то что это можно интерпретировать как шутку, термин *политический уровень* зачастую точен, поскольку все семь уровней в модели OSI подчиняются единой политике, созданной организацией, которая разработала саму сеть передачи данных.

Процесс обмена данными

Перечисленные в предыдущих разделах семь уровней модели OS1 работают вместе и представляют собой единую коммуникационную систему. Связь имеет место, когда, протокол, работающий на одной из систем и относящийся к определенному уровню модели, напрямую взаимодействует с протоколом того же уровня другой систем! Например, уровень приложений системы-источника логически связывается с уровне приложений системы-получателя. Уровень представлений системы-источника таю передает данные уровню представлений системы-получателя. Такое взаимодействие устанавливается между каждым из семи уровней модели.

Эта логическая связь между соответствующими уровнями семейства протоколов не включает в себя множество различных физических соединений между двумя коммуникационными системами.

Посылаемая информация каждого протокола упаковывается в предназначенное для нее место в протоколе уровня, непосредственно следующего за ним. Продукт упаковки данных одного уровня в другой называется *пакетом данных*.

Примечание

Упаковка данных — это процесс, при котором информация одного протокола вкладывается (или упаковывается) в область данных другого протокола. В модели OSI по мере продвижения данных по иерархии протоколов каждый уровень упаковывает ин формацию уровня, который находится непосредственно над ним.

Начиная с источника данных, как показано на рис. 1.2, данные приложения упаковываются в информацию уровня представлений. Для уровня представлений э данные являются исходными. Уровень представлений переправляет данные на сеанс вый уровень, который пытается поддерживать сеанс связи синхронизированным. Д лее сеансовый уровень передает данные на транспортный уровень, который осуществляет транспортировку данных от системы-отправителя системе-получателю. Сетевой уровень добавляет в пакет информацию об адресе и маршруте передачи пакета и передает его на канальный уровень. Этот уровень разбивает пакет на кадры и осуществляет подключение к физическому уровню.

На первом уровне, как показано на рисунке, физический уровень передает данные в виде битов через среду передачи данных, например, оптоволокно или медные провода. В сети пункта назначения пакеты проходят обратную процедуру с первого до седьмого уровня модели OSI. Каждое устройство на пути пакета считывает только ту информацию, которая необходима для дальнейшего перенаправления данных от отправителя к получателю. Каждый протокол разворачивает вложенные данные и считывает те данные, которые были посланы соответствующим уровнем системы-отправителя.

Рассмотрим процедуру передачи данных, инициируемую при открытии Web-страницы с помощью Web-броузера. Следуя адресу, например www.telegis.net, броузер посылает запрос протоколу TCP на открытие соединения с Web-сервером, расположенным по адресу www.telegis.net. (Многие приложения, которые используют протокол TCP, не включают в работу уровень представлений и сеансовый уровень, как в данном примере.) Затем протокол TCP посылает указание сетевому уровню (протокол IP) направить пакет с IP-адреса системы-отправителя на IP-адрес системы-получателя. Канальный уровень берет этот пакет и снова инкапсулирует его под тот тип канального уровня, который был на системе-отправителе, например Ethernet. Физический уровень доставляет сигнал от системы-отправителя к следующей системе по пути следования сигнала, например, к маршрутизатору. Маршрутизатор деинкапсулирует пакет данных, считывает информацию сетевого уровня, опять, если это необходимо, инкапсулирует пакет в соответствии с типом следующего канального уровня по пути следования к системе-получателю и маршрутизирует пакет.

Описанный выше процесс передачи данных продолжается до тех пор, пока пакет не будет доставлен на IP-адрес системы-получателя. По указанному IP-адресу канальный уровень деинкапсулирует пакет, видит, что адрес получателя соответствует адресу локальной системы, и передает данные IP-пакета на транспортный уровень. Транспортный уровень получает подтверждение установки соединения и передает данные броузера Web-серверу www.telegis.net. Затем Web-сервер отвечает на запрос браузера и отсылает ему назад данные страницы (используя ту же процедуру, но поменяв местами адреса системы-получателя и системы-отправителя).

Описываемые в этой книге устройства компании Cisco работают на трех уровнях модели OSI — физическом, канальном и сетевом. Используя информацию, предоставляемую этими уровнями, они передают данные из одного места в другое по пути следования пакета от отправителя к получателю. В этой книге авторы все время будут ссылаться на данные уровни и подробно объяснять, как ОС IOS использует информацию протокола каждого уровня. Некоторые из устройств компании Cisco, например, мосты и коммутаторы, работают только на

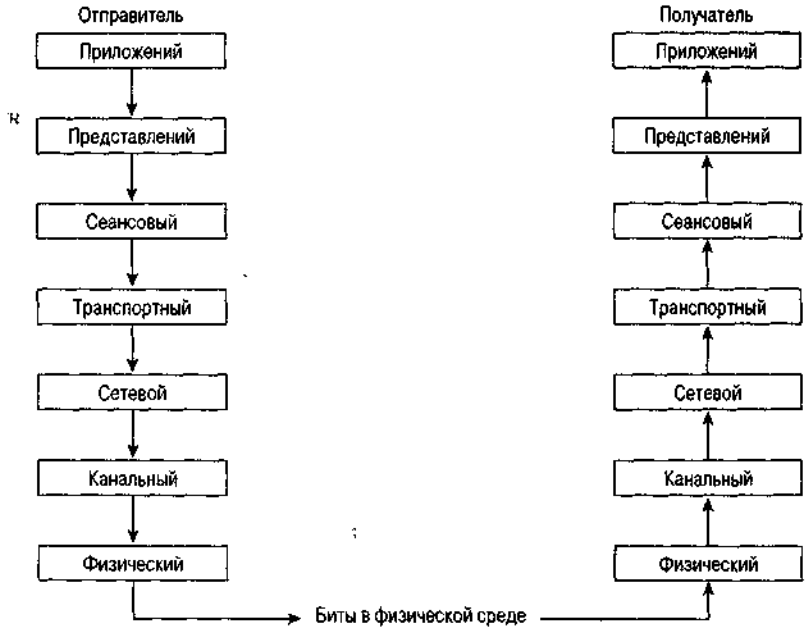


Рис. 1.2. Схема прохождения данных через семиуровневую модель OSI от системы-отправителя к системе-получателю

уровне канала передачи данных. Другие, например, маршрутизаторы, как показано на рис. 1.3, работают только сетевом уровне. В следующем разделе описаны различные типы устройств межсетевого взаимодействия.

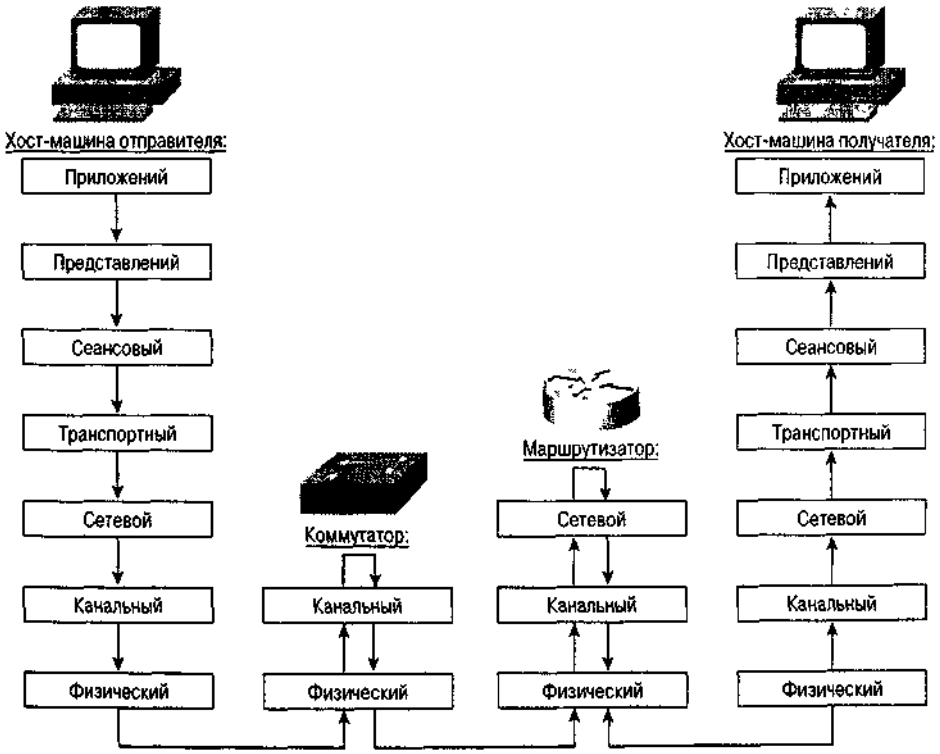


Рис. 1.3. Описание работы модели OSI при передаче данных через коммутатор и маршрутизатор от хост-машины отправителя к хост-машине получателя

Типы устройств межсетевого взаимодействия

Устройства компании Cisco можно разделить на три основные категории: мосты и коммутаторы, маршрутизаторы и серверы доступа. Рассмотрим сначала мосты и коммутаторы.

Мосты и коммутаторы

Мост — это сетевое устройство, которое оперирует на уровне канала передачи данных. Мост на уровне канала передачи данных соединяет несколько сетевых сегментов в один логический сетевой сегмент. Существует несколько различных типов мостов.

- Прозрачные или обучающиеся.
- Инкапсулирующие.
- Транслирующие.
- С маршрутизацией от источника.
- Транслирующие с маршрутизацией от источника.

Хотя ОС IOS содержит средства для работы со всеми вышеупомянутыми типами мостов, в данной книге описываются лишь первые три. Мосты с маршрутизацией от источника и транслирующие мосты с маршрутизацией от источника используются в сетях Token Ring.

Применение мостов позволяет разделить физический и логический виды трафика и этим снизить нагрузку на сегмент сети. Основным достоинством применения мостов является доступность, надежность, масштабируемость и управляемость сети за счет ее разбиения на отдельные физические составляющие. В этой книге описаны принципы мостовых соединений с точки зрения их отношения к маршрутизации.

Мосты осуществляют свои функции, проверяя в каждом пакете информацию канального уровня и выполняя дальнейшую переадресацию пакетов лишь при необходимости. Информация о том, какие пакеты следует переадресовывать в какой сетевой сегмент, заносится в мост в процессе обучения и хранится в таблицах переадресации. На рис. 1.4 показана такая таблица. Она содержит список известных адресов канального уровня и соответствующие этим адресам сегменты сети, в которых

находятся указанные устройства.

Чтобы определить наилучший метод переадресации пакетов на заданный канальный уровень получателя, мосты обмениваются данными, используя для этого протокол охватывающего дерева (Spanning Tree Protocol — STP). Этот протокол позволяет мостам строить безкольцевую топологию переадресации пакетов. *Безкольцевая топология* гарантирует, что пакет проходит по одному сегменту сети лишь один раз. Применение данной технологии необходимо для того, чтобы избежать возникновения "штормов широковещания" (broadcast storms) и множественной пересылки параллельно включенными мостами одного и того же пакета в данный сегмент. *Шторм широковещания* — это событие в сегменте сети, при котором *широковещательный пакет*, т.е. пакет, предназначенный каждой рабочей станции, находящейся в сегменте сети, посылается по замкнутому кольцу, вызывая перегрузку сегмента сети по трафику.

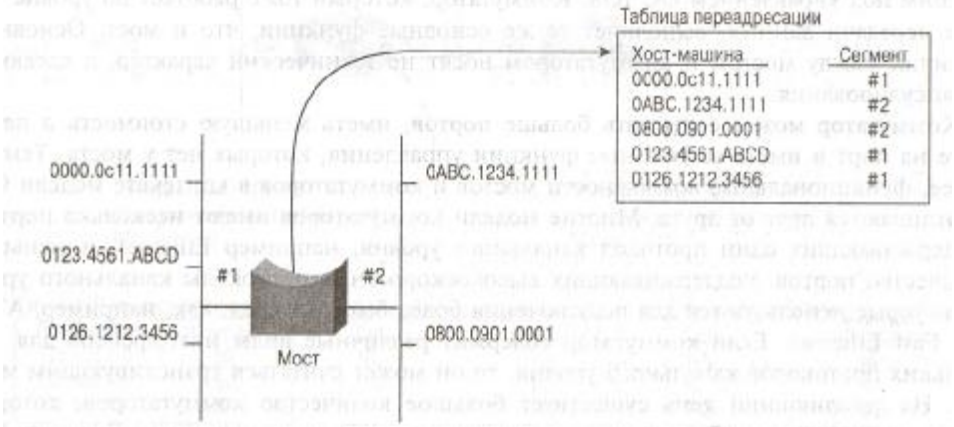


Рис. 1.4. Таблица переадресации отображает адреса канального уровня на физические сегменты сети

Простейшим мостом является *прозрачный мост*, который способен обрабатывать соединение только с одинаковыми протоколами канального уровня. *Инкапсулирующие* и *транслирующие* мосты можно рассматривать в качестве прозрачных мостов, которые обладают дополнительными функциями, позволяющими взаимодействовать различным протоколам канального уровня.

Инкапсулирующий мост вкладывает кадр канального уровня одного типа в кадр канального уровня другого типа, что делает возможным прозрачное мостовое соединение между одинаковыми канальными уровнями передачи данных, когда они физически разделены вторым отличающимся канальным уровнем. Как пример можно привести два инкапсулирующих моста, каждый из которых оснащен портом Ethernet и портом последовательной передачи данных. Эти мосты позволяют связать между собой два сегмента сети Ethernet, которые связаны между собой каналом последовательной передачи данных. Канал последовательной передачи данных отличается от канала Ethernet средой передачи данных уровня 2. Мостовое соединение с инкапсуляцией позволяет передавать весь кадр сети Ethernet из одного сегмента Ethernet в другой, физически отделенный сегментом с последовательной передачей данных, поскольку мост инкапсулирует кадр сети Ethernet в протокол канала последовательной передачи данных. Результатом этого процесса является то, что устройства взаимодействуют так, словно находятся в одном логическом сегменте Ethernet.

Транслирующие мосты выполняют функции прозрачного моста между двумя различными типами протоколов канального уровня. Например, транслирующий мост может преобразовывать кадры сети Ethernet в кадры сети Token Ring, выполняя трансляцию на уровне передачи данных. Если соединить два устройства, использующих различную среду передачи данных, с помощью транслирующего моста, то это будет выглядеть так, как будто два устройства работают в одном логическом сегменте сети. Прозрачное взаимодействие устройств с различной средой передачи данных обеспечивает необходимые условия соединения между двумя устройствами, которым требуется взаимодействовать на уровне передачи данных.

Коммутатор компании Cisco по сути своей является многопортовым мостом, работающим под управлением ОС IOS. Коммутатор, который тоже работает на уровне канала передачи данных,

выполняет те же основные функции, что и мост. Основные различия между мостом и коммутатором носят не технический характер, а касаются инкапсулирования.

Коммутатор может содержать больше портов, иметь меньшую стоимость в пересчете на порт и иметь встроенные функции управления, которых нет у моста. Тем не менее, функциональные возможности мостов и коммутаторов в контексте модели OSI не отличаются друг от друга. Многие модели коммутаторов имеют несколько портов, поддерживающих один протокол канального уровня, например Ethernet, и меньшее количество портов, поддерживающих высокоскоростные протоколы канального уровня, которые используются для подключения более быстрых сред, как, например, ATM или Fast Ethernet. Если коммутатор содержит различные виды интерфейсов для нескольких протоколов канального уровня, то он может считаться транслирующим мостом. На сегодняшний день существует большое количество коммутаторов, которые имеют интерфейсы, работающие с различными скоростями, например, Ethernet, Fast Ethernet и Gigabit Ethernet.

На рис. 1.5 показан пример небольшой сети, в которой используются коммутаторы.

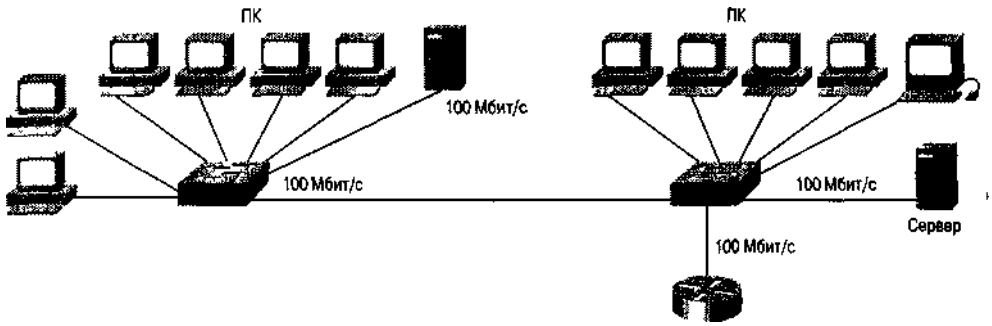


Рис. 1.5. Структурная схема малой сети с использованием коммутаторов

Маршрутизаторы

Маршрутизатор — это устройство, распределяющее пакеты по сети с помощью информации сетевого уровня. В данной книге основное внимание уделено трем протоколам сетевого уровня: IP, IPX и AppleTalk. Маршрутизатор извлекает данные об адресации сетевого уровня из пакета данных. В маршрутизаторе также имеются алгоритмы, называемые протоколами маршрутизации, с помощью которых он строит таблицы. В соответствии с этими таблицами и определяется тот маршрут, по которому должен быть направлен пакет, чтобы достичь конечного пункта назначения. Если маршрутизатор является многопротокольным, т.е. понимает несколько форматов адресов сетевого уровня и может работать с несколькими протоколами маршрутизации, к каковым и относятся маршрутизаторы компании Cisco, то, как показано на рис. 1.6, он хранит отдельные таблицы маршрутизации для каждого из протоколов сетевого уровня, маршрутизация которого осуществляется.

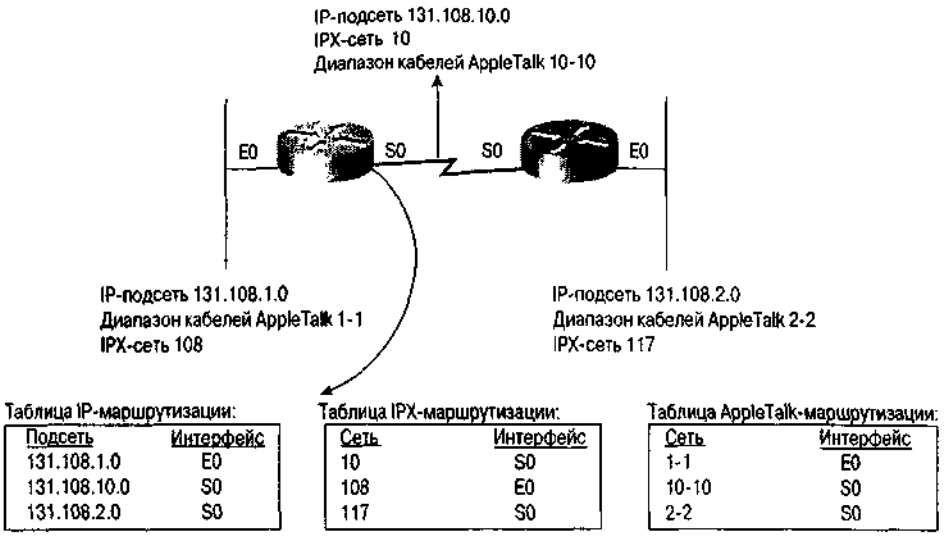


Рис. 1.6. Многопротокольные маршрутизаторы хранят таблицы маршрутизации для каждого из обрабатываемых ими протоколов сетевого уровня

Мосты и коммутаторы соединяют несколько физических сетей в одну логическую сеть, а маршрутизаторы соединяют логические сети и маршрутизируют пакеты данных между ними, используя информацию, полученную протоколами маршрутизации и хранящуюся в таблицах маршрутизации. Основным преимуществом маршрутизатора (по сравнению с использованием моста любого типа) является то, что он физически и логически разделяет сеть на несколько управляемых сегментов. Это позволяет контролировать маршрутизируемые пакеты данных и одновременно обрабатывать множество различных протоколов сетевого уровня. В этой книге рассматривается множество опций конфигурирования маршрутизаторов, управляемых ОС IOS.

Серверы доступа

Сервер доступа, также называемый *коммуникационным сервером*, — это сервер, который позволяет подключать к сети асинхронные устройства. Чаще всего сервер доступа используется для подключения к сети Internet компьютеров, оборудованных модемами. Сервер доступа совмещает в себе функции маршрутизатора с функциями асинхронных протоколов.

Если компьютер подключен к серверу доступа через асинхронный интерфейс, то сервер доступа с помощью определенного программного обеспечения позволяет подключенной машине работать так, как если бы она была подключена к сети. К примеру, сервер доступа может иметь 16 асинхронных портов и один порт Ethernet. Любое устройство, подсоединенное к асинхронному порту данного сервера доступа, работает так, как будто включено непосредственно в сеть Ethernet, в которой находится сервер доступа. Сервер доступа дает возможность пользователям, работающим с протоколами IP, IPX или AppleTalk, действовать в локальной сети с удаленного компьютера так, как если бы они были подключены к локальной сети непосредственно. В этой книге рассматриваются вопросы, касающиеся функций и конфигурирования серверов доступа.

Пример сетевого комплекса

На рис. 1.7 показана структура сети, которая используется в этой книге в качестве примера. Эта сеть призвана пояснить использование ОС IOS в следующих средах:

- Различные технологии локальных сетей: Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring и Fiber Distributed Data Interface (FDDI) (см. главу 3, "Основы интерфейсов устройств Cisco").
- Различные технологии синхронных и асинхронных глобальных сетей: HDLC, PPP, Frame Relay, ATM и ISDN (см. главу 3).
- IP-маршрутизация (см. главу 4, "Основы TCP/IP").
- IPX-маршрутизация (см. главу 6, "Основы IPX").

- AppleTalk-маршрутизация (см. главу 5, "Основы AppleTalk").

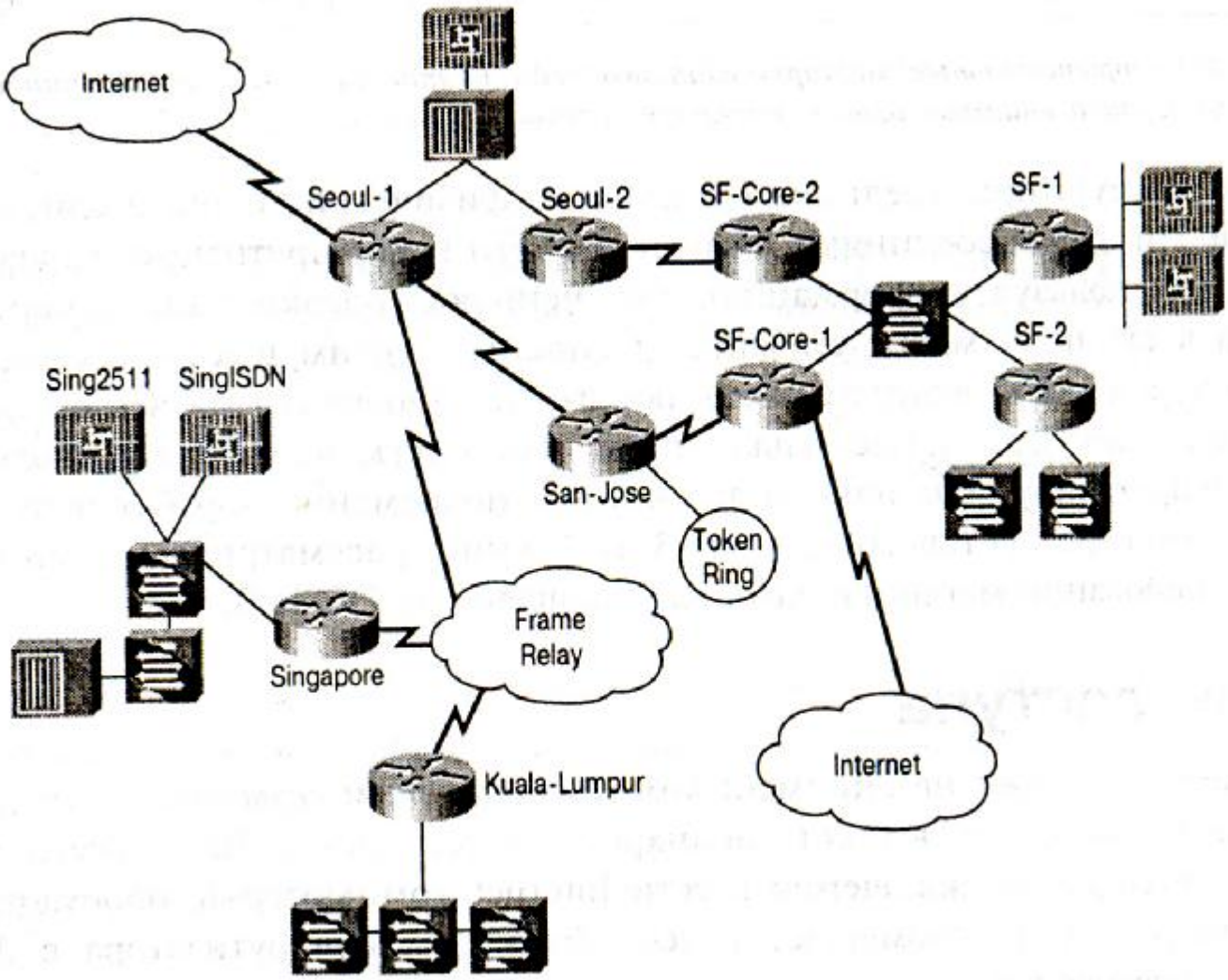


Рис. 1.7. Сетевой комплекс компании Zoom Integrated Products (ZIP)

Эта сеть принадлежит вымышленной компании Zoom Integrated Products (ZIP), корпоративные офисы которой находятся в Сан-Франциско, штат Калифорния. Компания производит компоненты для электронной промышленности. Азиатский отдел продаж расположен в Сеуле. Как корпоративные офисы, так и отдел продаж в Азии подключены к сети Internet. Компания ZIP также имеет производственные мощности в Сингапуре и Куала-Лумпуре (Малайзия).

Для связи заводов в Сингапуре и Куала-Лумпуре с отделом продаж в Сеуле сеть фирмы ZIP использует технологию Frame Relay. Узел доступа в Сеуле имеет входные линии ISDN для предоставления удаленного доступа к сети. Корпоративные офисы соединены между собой магистральным каналом, работающим по технологии Gigabit Ethernet, и с помощью трех сетевых сегментов, основанных на Fast Ethernet. Два сегмента сети Fast Ethernet обеспечивают скоростной канал между зданиями офисов, а один сегмент обслуживает серверы доступа для удаленных корпоративных клиентов, использующих коммутируемые каналы передачи данных. Также в сети корпорации существуют дополнительные серверы доступа для локальных удаленных пользователей в городах Сеуле и Сингапуре, которые работают через коммутируемые каналы связи. Корпоративные офисы соединены с отделами продаж с помощью резервных HDLC-каналов. Завод по сборке, расположенный в Сан-Хосе (Калифорния), имеет два сдвоенных HDLC-канала, один — для связи с корпоративными офисами, второй — для связи с отделами продаж в Сеуле. Завод в Сан-Хосе использует в цехах сеть Token Ring.

Компания ZIP использует в своей сети различные протоколы, включая IP, IPX и AppleTalk. Коммутаторы компании Cisco используются во внутриофисных сетях, а маршрутизаторы — для связи

всех офисов и городов. (Как показано на рис. 1.7, каждый маршрутизатор имеет свое название.) В большинстве городов имеется как минимум один сервер доступа, обслуживающий удаленных пользователей по коммутируемым каналам связи.

Весь сетевой комплекс компании ZIP состоит из нескольких сетей, территориально распределенных по всему миру. В нем используется несколько протоколов сетевого уровня и протоколы для глобальных сетей. В нем также применяются комбинации из маршрутизаторов, коммутаторов и серверов доступа. Последние обслуживают асинхронные устройства. Хотя этот сетевой комплекс является всего лишь примером, описанная структура представляет собой типичное на сегодняшний день решение. На примере сети компании ZIP будет показано, как необходимо сконфигурировать все работающие под управлением ОС IOS устройства Cisco, чтобы превратить данную вымышленную сеть в реальность.

Резюме

Изучив эту главу, читатель станет увереннее обращаться с моделью OSI, а также усвоит основы функционирования мостов, маршрутизаторов, коммутаторов и серверов доступа. В следующей главе приводятся основы конфигурирования устройств Cisco.

Постарайтесь запомнить следующие основные концептуальные положения данной главы.

- Операционная система Cisco IOS — это программное обеспечение, под управлением которого работают устройства Cisco.
- Рассматриваемые в этой книге устройства Cisco работают на трех уровнях модели OSI — физическом, канальном и сетевом.
- ОС IOS использует информацию протоколов каждого уровня модели OSI.
- Мосты и коммутаторы работают на канальном уровне и соединяют несколько сетевых сегментов на основе различных канальных уровней в один логический сетевой сегмент.
- Маршрутизаторы работают на сетевом уровне и управляют передачей пакетов через сеть на основе информации сетевого уровня.
- Серверы доступа связывают асинхронные устройства с сетью, позволяя им работать в сети.

Дополнительная литература

Более полное описание материала, приведенного в данной главе, можно найти в следующих изданиях:

1. Halsall, F. *Data Communications, Computer Networks and Open Systems,* Fourth Edition. Reading, Massachusetts: Addison-Wesley Publishing Company, 1996,

2. Perlman, R. *Interconnections: Bridges, Routers, Switches and Internetworking Protocols,* Second Edition. Reading, Massachusetts: Addison-Wesley Publishing Company, 1999.

3. Peterson, L. and B.S. Davie. *Computer Networks: A System Approach,* Second Edition. San Francisco, California: Morgan Kaufmann Publishers, 1999.

Глава 2

Основы конфигурирования устройств

Ключевые темы этой главы

- **Предварительное конфигурирование.** Основы конфигурирования работающих под управлением ОС IOS устройств Cisco, начиная с получения устройства в транс портной упаковке
- **Система помощи.** Использование встроенной в ОС IOS системы помощи
- **Непривилегированный и привилегированный режимы.** Описание двух уровней об ращения к устройствам Cisco %
- **Вопросы конфигурирования памяти.** Описание двух видов памяти, используемых в устройствах Cisco, — энергонезависимой и флэш-памяти
- **Пользовательский режим конфигурирования.** Один из методов конфигурирования IOS, применяемых для динамического конфигурирования устройств Cisco из встроенной памяти, из командной строки и с помощью сервера
- **Команды конфигурирования.** Структура команд конфигурирования, используемых ОС IOS, и основные команды конфигурирования

В этой главе описываются основы конфигурирования ОС IOS, ссылки на которые делаются на протяжении всей книги. Сначала перечисляются основные действия после изъятия устройства из заводской упаковки. Затем рассмат риваются основные компоненты IOS, включая систему помощи, конфигурирование памяти и структуру команд конфигурирования. Контекстом примеров конфигурации устройств будет служить представленная в предыдущей главе сеть компании ZIP.

Предварительное конфигурирование

Все устройства, работающие под управлением IOS, поставляются с завода сконфигурированными в минимально возможном объеме. Например, мосты и коммутаторы поставляются с установками, которые позволяют им переадресовывать пакеты с использованием алгоритма охватывающего дерева на всех портах, но конфигурирование таких сложных функций, как фильтрация пакетов, не выполняется В маршрутизаторах и серверах доступа компания Cisco производит установку лишь минимального количества параметров конфигурирования Это влечет за собой необходимость ввода в устройства информации, только посте осуществления которого они смогут выполнять свои функции Когда маршрутизатор (или сервер доступа) присылается с завода, все его интерфейсы или выключены или административно заблокированы

Чтобы начать установку конфигурации устройства Cisco, вставьте вилку устройства в розетку сети питания и найдите тумблер включения, располагающийся на задней стенке Если включить тумблер питания (иногда он обозначается цифрой 1), то на устройство подастся напряжение, и на передней панели загорятся светодиодные индикаторы статуса.

Следует запомнить исключение из описанного выше правила Оно касается популярных маршрутизаторов серии Cisco 2500 Маршрутизаторы именно этой серии не имеют на передней панели светодиодных индикаторов, показывающих, что на устройство подано питание Однако подачу электропитания можно проверить с помощью индикатора состояния, расположенного на задней стенке устройства рядом со вспомогательным (AUX) портом консоли.

Примечание

Многие светодиодные индикаторы, относящиеся к другим элементам устройства, например, интерфейсам локальных или глобальных сетей, не засветятся, пока эти элементы не будут сконфигурированы. Конфигурирование интерфейсов локальных или глобальных сетей невозможно без подачи питания и ввода соответствующих команд конфигурирования ОС IOS.

Порт консоли

На следующем этапе конфигурирования работающего под управлением ОС IOS устройства необходимо найти порт консоли. Каждое устройство Cisco имеет порт консоли, который используется для обращения к нему с помощью непосредственно подключаемого терминала. Порт консоли часто представляет собой порт интерфейса RS-232C или RJ-45 и обозначается надписью "Console" ("Консоль").

Обнаружив порт консоли, необходимо подключить выделенный для этой цели терминал или ПК с эмулятором терминала. Компанией Cisco с каждым устройством поставляются необходимые для этого кабели. Если к устройству подключается терминал, следует воспользоваться разъемом RS-323C на терминале, подключить к нему кабель RJ-45 и затем подсоединить всю эту сборку непосредственно к устройству.

Для некоторых устройств, например маршрутизатора Cisco 7500, необходимо использовать разъем RS-232C на обоих концах кабеля RJ-45, тогда как другие, например устройства серии Cisco 2500, этого не требуют. Если планируется подключить к устройству ПК, то, возможно, предется подсоединить к последовательному порту ПК разъем DB-9, а затем использовать для подключения к устройству кабель RJ-45. Если устройство имеет порт консоли типа RJ-45 (как, например, маршрутизаторы серий Cisco 2500 или Cisco 3600), то необходимо иметь только соответствующий переходник от порта RJ-45 к консоли (часто это разъем RS-232C) или к персональному компьютеру (часто это разъем DB-9).

Установив физическое соединение между терминалом или ПК и устройством, необходимо произвести конфигурирование терминала для его соответствующего взаимодействия с устройством. Для этого следует настроить параметры терминала (или программы эмуляции терминала на ПК) таким образом, чтобы поддерживались следующие установки:

- тип эмулируемого терминала — VT100;
- скорость передачи данных — 9600 бод;
- запрет контроля четности;
- 8 бит данных;
- 1 стоп-бит.

После проверки правильности этих установок следует подать на устройство питание. На экране терминала появится сообщение, подобное тому, что выводится маршрутизатором Cisco 7206:

```
System Bootstrap, Version 12.1 (1), SOFTWARE'
Copyright (c) 1986-2000 by Cisco Systems

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) 7200 Software (C7200-P-M), RELEASE SOFTWARE 12.0(5)
```

Мосты и коммутаторы, которые функционируют под управлением IOS, могут выводить или не выводить подобное сообщение. Это зависит от модели устройства и выполняемых им функций. Вне зависимости от вида выводимого сообщения после подачи на устройство питания на экране терминала или устройства, эмулирующего терминал, должен высветиться какой-то результат. В некоторых случаях в зависимости от программы эмуляции терминала и установок для того, чтобы увидеть какую-нибудь реакцию, необходимо нажать на клавиатуре терминала клавишу <Enter> или <Return>.

Если сообщения на экране терминала или устройства, его эмулирующего, нет, проверьте соединения и удостоверьтесь в правильности установок терминала. Возможно, надо будет заглянуть в руководство *Getting Started Guide ("С чего начать")*, которое поставляется с каждым устройством Cisco.

Режим диалога конфигурирования системы

При первом включении питания все маршрутизаторы и серверы доступа входят в режим диалога конфигурирования системы. Этот интерактивный режим отображается на экране консоли и, задавая вопросы, помогает сконфигурировать основные элементы ОС IOS. В режиме диалога конфигурирования системы сначала запрашиваются глобальные параметры системы, а затем параметры интерфейсов.

При входе в режим диалога конфигурирования системы на экране терминала должно появиться следующее сообщение.

```
        System Configuration Dialog
At any point you may enter a question mark '?' for help.
Refer to the 'Getting Started' Guide for additional help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.
Would you like to enter the initial configuration dialog? [yes]:
```

Примечание
Показанные в данной главе примеры запросов и опций для интерфейсов могут не совпадать с тем, что будет выводить ваш маршрутизатор. IOS автоматически модифицирует диалог конфигурирования системы в зависимости от платформы и интерфейсов, установленных в маршрутизаторе. Приведенный выше пример взят для маршрутизатора серии Cisco 2500.

После этого для начала работы в режиме диалога конфигурирования системы можно нажать клавишу <Return> или <Enter>:

```
Would you like to enter the  initial  configuration dialog?   [yes]:
First,   would you like to see the current  interface summary?   [yes]:
```

Приведенный ниже список интерфейсов соответствует состоянию устройства при его непосредственной поставке с завода: устройство еще не сконфигурировано. Поэтому все его интерфейсы показаны как не прошедшие конфигурирование (об этом говорит слово NO в столбце Ok?). Поскольку для интерфейсов IP-адреса не определены, в столбце IP Address для всех интерфейсов стоит значение unassigned (не присвоен). В столбце Method стоит значение not set (нет установок). Значение этого столбца показывает, как конфигурировался интерфейс: вручную или автоматически через сеть. На данный момент установки для интерфейсов еще не производились. Последние два столбца называются Status ("Состояние") и Protocol ("Протокол"). Столбец status показывает статус интерфейса, а столбец Protocol — тип запущенного на данном интерфейсе протокола канального уровня. По умолчанию на новом устройстве все интерфейсы и протоколы канального уровня имеют статус down ("выключен").

```
Interface IP-Address OK? Method Status Protocol
Ethernet0 unassigned NO not set down down
Serial0 unassigned NO not set down down
```

Продолжая интерпретировать этот список интерфейсов, нужно отметить, что интерфейс Ethernet является интерфейсом локальной сети, а последовательный интерфейс Serial — интерфейсом глобальной сети. Имя интерфейса Ethernet0 обозначает первую подключаемую к данному устройству локальную сеть Ethernet, а имя интерфейса Serial0 — первую подключаемую глобальную сеть с последовательной передачей. Физические порты на задней стенке корпуса устройства имеют точно такие же названия. Более подробно различные типы интерфейсов будут обсуждаться в главе 3, "Основы интерфейсов устройств".

Следующими этапами конфигурирования устройства являются присвоение имени устройству, т.е. логического названия, которое будет ассоциироваться с данным физическим устройством, и задание пароля доступа. Начнем с имени устройства. В качестве примера конфигурируемого устройства используем маршрутизатор сети компании ZIP в Сингапуре:

```
Configuring global parameters:
Enter host name [Router]: Singapore
```

Совет
Указанные в квадратных скобках значения ОС IOS принимает по умолчанию. В данном примере ввод значений по умолчанию в явном виде показан для большей наглядности.

Как будет показано в следующих разделах этой главы, в ОС IOS существуют два уровня команд: привилегированный и непривилегированный. Для каждого устройства следует задать пароль. Этот пароль является ключом для входа в привилегированный режим. Пароли для входа в привилегированный режим должны содержаться в секрете и требуют такого же обращения, как пароли суперпользователя или системного администратора. Настоятельно рекомендуется использовать для установки пароля не старый метод enable password, а новый метод установки пароля enable secret. Этой рекомендации следует придерживаться из-за того, что команда enable secret использует алгоритм одностороннего шифрования передаваемых

данных. В данном примере, чтобы показать все возможности ОС IOS, используются оба метода установки пароля, однако в остальных примерах в книге применяется метод enable secret. Таким образом, в данном примере для режима enable secret устанавливается пароль ! zippy2u, а для режима enable password — пароль ! zippy4me:

```
The enable secret word is a one-way cryptographic secret that is used
instead of the enable password word when it exists.
Enter enable secret:  \zippy2u
The enable password is used when there is no enable secret and when using
older software and some boot images.
Enter enable password:   !zippy4me
```

Термин "виртуальный терминал" обозначает одно логическое соединение терминала с устройством, работающим под управлением ОС IOS. По умолчанию все устройства, работающие под управлением ОС IOS, позволяют одновременно поддерживать пять Telnet-сеансов виртуального терминала (с номерами от 0 до 4). Если устройство с ОС ЮС активно работает в сети, то для доступа ко всем функциям ОС IOS через канал виртуального терминала можно использовать программу Telnet, причем этот доступ ничем не отличается от доступа через порт консоли. Например, можно использовать виртуальный терминал, чтобы подключиться к маршрутизатору и затем с помощью пароля режима enable secret войти в режим исполнения привилегированных команд. В данном примере устанавливается один пароль виртуального терминала Zipmein для всех пяти сеансов:

```
Enter virtual terminal password: Zipmein
```

Один и тот же пароль для каждого сеанса виртуального терминала устанавливается обычно по следующей причине. Подключаясь к маршрутизатору, пользователи зачастую не задают номер конкретного виртуального терминала, который хотят задействовать, а используют первый свободный.

Следующий шаг в диалоге конфигурирования системы связан с заданием желаемых типов протоколов. На этом этапе необходимо разрешить использование устройством протокола простого управления сетью (Simple Network Management Protocol — SNMP). Более подробно конфигурирование протокола SNMP описывается в главе 7, "Основы администрирования и управления". Итак, в данном примере разрешается использование протокола SNMP, и в качестве цепочки сообщества принимается значение по умолчанию public:

```
Configure SNMP Network Management?  [yes]:  yes
Community string  [public]:  public
```

Теперь диалог конфигурирования системы задаст вопрос относительно необходимости конфигурирования протокола DECnet, представляющего собой протокол сетевого уровня корпорации Digital Equipment Corporation. Поскольку этот протокол в сети компании ZIP не используется, вводится ответ по.

```
Configure DECnet?  [no]:  no
```

В сети компании ZIP применяется протокол AppleTalk с многозоновыми сетями. (Подробнее протокол AppleTalk обсуждается в главе 5, "Основы Apple Talk".)

```
Configure AppleTalk? [no]: yes Multizone networks? [no]: yes
```

В сети компании ZIP также используется протокол IPX:

```
Configure IPX? [no]: yes
```

Межсетевой протокол Internet Protocol (IP) является основным для корпоративной сети компании ZIP. Соответственно, на этом этапе необходимо разрешить его использование. При этом ОС IOS просит выбрать протокол IP-маршрутизации, который будет использоваться маршрутизаторами для обмена маршрутной информацией. В примере не дается разрешение на использование протокола внутренней маршрутизации между шлюзами Interior Gateway Routing Protocol (IGRP) в качестве протокола маршрутизации. Конфигурирование протоколов IP-

маршрутизации будет рассмотрено в главе 4, "Основы TCP/IP":

```
Configure  IP?   [yes]:
Configure  IGRP routing?   [yes]:   no
```

После выбора протоколов диалог настройки ОС IOS потребует ввода информации по каждому интерфейсу, установленному на маршрутизаторе. Для каждого интерфейса локальной или глобальной сети требуется ввести информацию об используемом протоколе. Различные типы интерфейсов локальных и глобальных сетей описываются в главе 3, а такие вопросы, касающиеся протоколов, как IP-адресация, номера IPX-сетей, значения диапазонов кабелей для сетей AppleTalk, обсуждаются в последующих главах. Расположенный в Сингапуре маршрутизатор компании ZIP имеет один интерфейс локальной сети — Ethernet и один интерфейс глобальной сети — Frame Relay. Конфигурирование протоколов IP, IPX и AppleTalk для каждого интерфейса осуществляется таким образом:

```
Configuring interface parameters:
Configuring interface EthernetO:
```

Следующий вопрос запрашивает данные о том, используется ли конфигурируемый интерфейс, т.е. необходимо ли, чтобы этот интерфейс был включен и не был административно заблокированным. На маршрутизаторе в Сингапуре интерфейсы EthernetO и SerialO должны быть включенными:

```
Is this interface in use? [no]: yes
```

Теперь необходимо сообщить маршрутизатору, чтобы на этом интерфейсе применялся протокол IP, а для него использовался IP-адрес 131.108.1.1 и маска подсети 255.255.255.128. Как уже было сказано, необходимая информация об IP-адресации, разбиении на подсети и конфигурировании дана в главе 4.

```
Configure IP on this interface?   [no]:  yes
  IP address  for this  interface:   131.108.1.1
Number of bits in subnet field   [0]:   9
Class B network is  131.108.0.0,   9 subnet bits,  mask is  /25
```

Одновременно с использованием протокола IP в сети компании ZIP в Сингапуре используются протоколы IPX и AppleTalk. Для включения данных протоколов необходимо ввести информацию о номере IPX-сети и о значении кабельного диапазона сети AppleTalk. Детально протокол IPX описывается в главе 6, "Основы IPX", а протокол AppleTalk — в главе 5, "Основы AppleTalk".

```
Configure IPX on this  interface?   [no]:  yes
  IPX network number   [1]:   4010
Configure AppleTalk on this  interface?   [no]:  yes
  Extended AppleTalk network?   [no]:  yes
  AppleTalk starting cable  range   [0]:   4001
```

На этом маршрутизаторе также необходимо сконфигурировать интерфейс SerialO с теми же протоколами сетевого уровня, что делается следующим образом:

```
Configuring interface SerialO:
Is this interface in use? [no]: yes Configure IP unnumbered on this interface?
[no]: no
IP address for this interface: 131.108.242.6
Number of bits in subnet field [0]: 14
Class B network is 131.108.0.0, 8 subnet bits; mask is /30 Configure IPX on this
interface? [no]: yes
IPX network number [2]: 2902 Configure AppleTalk on this interface? [no]: yes
Extended AppleTalk network? [no]: yes
AppleTalk network number [1]: 2902
```

Результатом работы диалога конфигурирования системы является скрипт команд конфигурирования, который интерпретируется устройством. Сам по себе диалог конфигурирования системы не выполняет собственно конфигурирование устройства, а создает скрипт команд конфигурирования, который затем интерпретируется устройством и используется для конфигурирования. Скрипт написан на языке, который необходимо знать для того, чтобы настраивать изделия компании Cisco, работающие под управлением ОС IOS. В остальной части данной книги собственно и рассматривается этот язык написания скриптов. Читатель, вероятно, сможет найти связь между вопросами, задаваемыми в ходе диалога конфигурирования системы, и приведенным ниже скриптом команд конфигурирования:

```
hostname Singapore
enable  secret  5  $2zu6m7$RMMZ8em/.8hksdkkh78p/TO
enable password   !zippy4me
line vty  0  4
password Zipmein
snmp-server community public
ip routing
!
ipx routing
appletalk routing
!
no decnet routing
!
interface EthernetO
ip address 131.108.1.1 255.255.255.128
ipx network 4010
appletalk cable-range 4001-4001
appletalk discovery
no mop enabled
!
interface SerialO
ip address 131.108.242.6 255.255.255.252
ipx network 100
appletalk cable-range 2902-2902
no mop enabled
!
end
Use this configuration? [yes/no]: yes
[OK]
Use  the enabled mode   'configure,'   command to modify this  configuration.
Press  RETURN  to  get  started!
```

После нажатия клавиши <Return> маршрутизатор должен вывести командную строку следующего вида:

```
Singapore>
```

Начиная с этого момента, пользователь попадает в режим EXEC, который используется для исполнения команд ОС IOS. Но прежде чем приступать к изучению режима исполнения команд, рассмотрим работу системы помощи Help.

Система помощи

В ОС IOS встроена система помощи, обратиться к которой можно из режима исполнения команд EXEC. Система помощи является контекстной, что означает, что оказываемая помощь зависит от того, что пользователь пытается сделать в ОС IOS на данный момент. Например, введя в командной строке знак "?", пользователь получит следующую информацию:

```
Singapore>?
```

```
Exec commands:
<1-99>            Session number to resume
access-enable     Create a temporary Access-List entry
access-profile    Apply user-profile to interface
attach            Attach to system component
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
mis               Exec mis router commands
mrinfo            Request neighbor and version information from a
                  multicast router
mstat             Show statistics after multiple multicast traceroutes
mtrace            Trace reverse multicast path from destination to source
name-connection   Name an existing network connection
pad               Open a X.29 PAD connection
ping              Send echo messages
—More—
```

В этом примере показан лишь первый экран помощи из доступных, так как здесь объем выводимой информации был ограничен. Чтобы получить полный список доступных в режиме EXEC команд, можно обратиться к сводному перечню команд операционной системы IOS компании Cisco (Cisco IOS Software Command Summary), который можно найти по адресу www.cisco.com/univercd/cc/td/doc/product/ software/ios!21/121cgcr/index.htm. Система помощи построена таким образом, что в левой части выводимого текста содержатся сами команды, а в правой — короткие пояснения к каждой из них. Некоторые команды состоят из одного слова; система помощи ставит пользователя в известность об этом, показывая, что единственным выбором у него является нажатие после этой команды клавиши возврата каретки обозначая это действие, выводя на экран символы <cr>:

```
Singapore>lock ?
<cr>

Singapore>lock
```

При использовании системы помощи пользователю не требуется снова вводит команду, получив справку или совет. После вывода справки ОС IOS сама вводит команду, по которой была запрошена информация. Это показано в предыдущем приме ре, где команда lock была автоматически введена ОС IOS после получения справки по этой команде.

Систему помощи также можно использовать для определения возможных опций команд режима EXEC. Как будет показано ниже, ОС IOS содержит множество команд для получения информации о текущем состоянии устройства. Многие из этих команд начинаются со слова show. В приведенном ниже примере перечислены все возможные опции команды show.

```
Singapore>show

alps              Alps information
backup            Backup status
bootflash:        Display information about bootflash: file system
bootvar           Boot and related environment variable
calendar          Display the hardware calendar
cef               Cisco Express Forwarding
ces               CES Show Commands
clock             Display the system clock
context           Show context information about recent crash(s)
dialer            Dialer parameters and statistics
diskO:            display information about diskO: file system
diskl:            Display information about diskl: file system
```

```
drip            DRiP DB
dss             DSS information
flash:          Display information about flash: file system
fras-host       FRAS Host Information
history         Display the session command history
hosts           IP domain-name, lookup style, nameservers, and host table
ipc             Interprocess communications commands
location        Display the system location
management      Display the management applications
microcode       Show configured microcode for downloadable hardware
mis             Multilayer switching information
modemcap        Show Modem Capabilities database
mpoa            MPOA show commands
ncia            Native Client Interface Architecture
PPP             PPP parameters and statistics
rmon            Rmon statistics
rtr             Response Time Reporter (RTR)
sessions        Information about Telnet connections
sgbp            SGBP group information
slotO:          Display information about slotO: file system
slotl:          Display information about slotl: file system
snmp            Snmp statistics
syscon          System Controller information
tacacs          Shows tacacs+ server statistics
terminal        Display terminal configuration parameters
traffic-shape   Traffic rate shaping configuration
users           Display information about terminal lines
version         System hardware and software status
vpdn            VPDN information

Singapore>show
```

тметим, что ОС IOS повторяет начальную часть введенной с клавиатуры команды, так что необходимость в ее повторении отсутствует.

Также встроенная в ОС IOS система помощи позволяет вводить команды не полностью, автоматически дополняя команду до конца при нажатии клавиши <Tab>. Если ввести часть команды, которая не имеет нескольких значений, и нажать клавишу <Tab>, то ОС IOS сама дополнит команду до конца. В качестве примера рассмотрим команду show sessions, которая позволяет увидеть все текущие Telnet-сеансы управления устройством через его каналы виртуального терминала. Если ввести

```
Singapore>show sess
```

и затем нажать клавишу <Tab>, то ОС IOS автоматически дополнит команду:

```
Singapore>show sessions
```

При вводе неоднозначной команды, например,

```
Singapore>show s
```

ОС IOS не сможет ее дополнить, потому что данная команда может быть интерпретирована как show sessions и как show snmp. В этом случае нажатие на клавишу <Tab> для большинства систем приведет к срабатыванию встроенного в терминал зуммера.

Совет

В режиме EXEC ОС IOS не обязательно вводить всю команду — однозначно интерпретируемые команды по умолчанию будут дополнены. Это означает, что ввод команд show sess и show sessions даст одинаковый результат.

Команда show sessions отличается от команды sessions. Команда sessions разрешает пользователю подключение к аппаратному модулю устройства с помощью сеанса виртуального терминала. Некоторые устройства Cisco имеют несколько аппаратных модулей, для обращения к каждому из которых нужен свой собственный виртуальный терминал. Примером этого являются

модули маршрутизирующего коммутатора (Route Switch Module — RSM) и асинхронного интерфейса передачи данных (Asynchronous Transfer Mode — ATM) в коммутаторах типа Catalyst. Пользователь может обозначить, к какому из нескольких установленных модулей он хочет подключиться, с помощью команды session, введя после нее номер модуля. Например, если в коммутаторе Catalyst имеется модуль ATM, и он считается модулем номер 3 (обычно, это означает, что он стоит в третьем слоте расширения устройства), то для получения доступа к модулю можно сделать следующее:

```
Router>session 3
Trying ATM-3. . .
Connected to ATM-3.
Escape character is '^]'.
ATM>
```

Выполнив эту команду, система устанавливает сеанс с ATM-модулем. Этот сеанс отличается по функциям от Telnet-сеанса с самим маршрутизатором или коммутатором: теперь все исполняемые команды будут выполняться ATM-модулем.

Непривилегированный и привилегированный режимы

В режиме EXEC возможно исполнение команд двух основных уровней. Команд] первого уровня исполняются в непривилегированном режиме. Непривилегированны режим обозначается символом "больше, чем" (>), размещаемым в командной строк после имени устройства, как показано ниже:

```
Singapore>
```

В этом режиме пользователю разрешено получать информацию о состоянии устройства, работающего под управлением IOS, но у него нет возможности изменять какие-либо параметры устройства.

Второй уровень включает команды привилегированного режима, который таю» известен под именем *разрешенный режим* (enable mode). Для того чтобы войти в этот режим, необходимо знать системный пароль, заданный в режиме `enable secret` Чтобы переключиться из непривилегированного режима в привилегированный, нужно ввести команду `enable`:

```
Singapore>enable
Password:
Singapore#
```

В приведенном выше примере после выдачи запроса пароля в соответствуй шей командной строке вводится пароль режима enable secret (в нашем случае это!zippy2u), который не повторяется на экране терминала. При смене режима на привилегированный система обозначает это заменой в командной строке символа ">" символом "#" (так называемая "решетка"). Для того чтобы перейти из привилегированного режима в непривилегированный, следует ввести команду режима EXEC disable:

```
Singapore #disable
Singapore>
```

Следует отметить, что в привилегированном режиме пользователю доступ большее количество команд, нежели в непривилегированном, что и показывает си тема помощи:

```
Singaporett?
Exec commands:
<1-99>            Session number to resume
access-enable     Create a temporary Access-List entry
```

```
access-profile    Apply user-profile to interface
access-template   Create a temporary Access-List entry
attach            Attach to system component
bfe               For manual emergency modes setting
calendar          Manage the hardware calendar
cd                Change current directory
clear             Reset functions
clock             Manage the system clock
configure         Enter configuration mode
connect           Open a terminal connection
copy              Copy from one file to another
debug             Debugging functions (see also 'undebug')
delete            Delete a file
dir               List files on a filesystem
disable           Turn off privileged commands
disconnect        Disconnect an existing network connection
enable            Turn on privileged commands
erase             Erase a filesystem
exit              Exit from the EXEC
format            Format a filesystem
help              Description of the interactive help system
lock              Lock the terminal
login             Log in as a particular user
logout            Exit from the EXEC
microcode         Microcode commands
mkdir             Create new directory
mls               Exec mis router commands
more              Display the contents of a file
mpoa              MPOA exec commands
mrinfo            Request neighbor and version information from a multicast router
mstat             Show statistics after multiple multicast traceroutes
mtrace            Trace reverse multicast path from destination to source
name-connection   Name an existing network connection
ncia              Start/Stop NCIA Server
no                Disable debugging functions
pad               Open a X.29 PAD connection
ping              Send echo messages
ppp               Start IETF Point-to-Point Protocol (PPP)
pwd               Display current working directory
reload            Halt and perform a cold restart
rename            Rename a file
–More–
```

Приведенный выше результат работы системы помощи для краткости усечен.

Вопросы конфигурирования памяти

Устройства Cisco имеют три блока памяти. Два из них отводятся под хранение конфигурации устройства, а третий — под хранение операционной системы IOS. Различие между командами конфигурирования и операционной системой заключается в том, что команды используются для формирования конфигурации устройства, а операционная система — это программное обеспечение, которое исполняется на этом устройстве.

Из данного раздела читатель узнает о двух типах памяти, в которой хранятся команды конфигурирования. Это оперативная память и энергонезависимая память. Также будет объяснено, как загрузить операционную систему IOS в третий тип памяти устройства — перезаписываемое постоянное запоминающее устройство с электронным стиранием, или флэш-память. Для того чтобы выполнять связанные с памятью команды, необходимо находиться в привилегированном режиме (как показано в приводимых ниже примерах).

Память конфигурации устройства

Текущую, или используемую, конфигурацию устройства, работающего под управлением ОС IOS, можно посмотреть с помощью команды show runing-config. Результатом работы данной команды является список команд конфигурирования ОС IOS, исполняемых

устройством, приведенный ниже:

```
Singapore#show running-config
Current configuration:
hostname Singapore
enable secret 5 $2zu6m7$RMMZ8em/.8hksdkkh78p/TO
enable password !zippy4me
line vty 0 4
password Zipmein
snmp-server community public

ip routing
ipx routing
appletalk routing
no decnet routing
!
interface Ethernet0
ip address  131.108.1.1  255.255.255.128
ipx network 4010
appletalk cable-range 4001-4001
appletalk discovery
no mop enabled
!
–More–
```

Объем выводимой информации здесь для краткости усечен.

Текущая конфигурация устройства хранится в оперативной памяти. При отключении питания устройства информация, содержащаяся в этом типе памяти, теряется. Если необходимо, чтобы после цикла отключения-включения питания устройстве восстанавливало текущую конфигурацию, ее следует записать в энергонезависимую память, называемую памятью стартовой конфигурации. Для копирования конфигурации в энергонезависимую память используется команда режима EXEC copy, которая производит запись из первой указываемой памяти во вторую:

```
Singapore#copy running-config startup-config
[OK] Singapore#
```

После ввода данной команды текущая конфигурация устройства, находящаяся в ОЗУ, копируется в качестве стартовой в энергонезависимую память. Также возможен обратный вариант, когда команда copy используется для копирования стартовой настройки в качестве текущей, как показано ниже:

```
Singapore#copy startup-config running-config
[OK]
Singapore#
```

Такое копирование может понадобиться, чтобы вернуться к стартовой конфигурации устройства после изменения текущей. Предположим, что в конфигурацию устройства внесено несколько изменений. После оценки поведения устройства оказывается, что изменения были неправильными. Если текущая конфигурация еще не была скопирована в качестве стартовой, то старую стартовую конфигурацию можно скопировать в качестве текущей. При копировании стартовой конфигурации из энергонезависимой памяти в ОЗУ в качестве текущей конфигурации следует помнить, что возможно слияние команд конфигурирования (см. раздел "Слияние и замещение команд конфигурирования" далее в этой главе).

Для просмотра стартовой конфигурации следует ввести команду режима EXEC show startup-config:

```
Singapore#show startup-config
Using 1240 out of 7506 bytes
!hostname Singapore
enable secret 5 $2zu6m7$RMMZ8em/.8hksdkkh78p/TO
```

```
enable password !zippy4me
line vty 0 4
password Zipmein
snmp-server community public
ip routing
ipx routing
appletalk routing
no decnet routing
!interface Ethernet0
ip address 131.108.1.1 255.255.255.128
ipx network 4010
appletalk cable-range 4001-4001
appletalk discovery
no mop enabled
!—More—
```

Этот результат исполнения команды, как и предыдущие, для краткости показан не полностью. Заметьте, что в первой строке указывается объем энергонезависимой памяти, использованный под стартовую конфигурацию, и общий ее объем.

Стартовая и текущая конфигурации совпадают после выдачи команды copy running-config startup-config. Однако, если производится переконфигурирование устройства (как будет показано далее), и измененная текущая конфигурация не сохраняется в качестве стартовой, то в следующий раз при отключении и включении питания устройство вернется к последней конфигурации, сохраненной в энергонезависимой памяти.

Стартовая конфигурация может быть полностью удалена из энергонезависимой памяти с помощью команды erase startup-config:

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

Если после этого перезагрузить маршрутизатор, отключив электропитание или используя привилегированную команду режима EXEC reload, то стартовая конфигурация будет пустой. Такая последовательность действий (стирание энергонезависимой памяти и перезагрузка устройства) приведет к тому, что вновь будет запущен описанный ранее в этой главе режим диалога конфигурирования системы.

Флэш-память хранения ОС IOS

Флэш-память — это то место, где устройства Cisco хранят двоичные исполняемые образы ОС IOS, которые и представляют собой операционную систему устройства. Не следует путать образы ОС IOS с ее конфигурациями. Как уже описывалось ранее в данной главе, конфигурация ОС IOS говорит устройству о его текущей конфигурации, тогда как образ ОС IOS является той самой двоичной программой, которая выполняет синтаксический анализ и собственно конфигурацию.

Устройство может хранить несколько образов ОС IOS. Это зависит от объема установленной флэш-памяти и размера образов операционной системы. Если в данном устройстве хранится несколько образов ОС IOS, то пользователь может указать, какой именно образ ОС IOS следует исполнять устройству после перезагрузки. Получаемые от компании Cisco образы ОС IOS могут быть скопированы в устройство с использованием нескольких различных протоколов передачи файлов, имеющих в основе протокол TCP/IP, включая простой протокол передачи файлов (Trivial File Transfer Protocol — TFTP), протокол передачи файлов (File Transfer Protocol — FTP), а также протокол удаленного копирования для платформы UNIX — UNIX remote copy protocol (rep). Далее мы рассмотрим применение для передачи образов ОС IOS в устройство только протоколов TFTP и FTP. Хотя протокол rcp также может быть использован в этих целях, он требует конфигурирования как устройства с ОС 1OS, так и rcp-сервера, а описание этого процесса выходит за рамки данной книги. Кроме того, использование протокола rcp в определенной

мере рискованно в плане защиты информации и требует более глубокого знания ОС IOS и сетевых технологий.

Решение об использовании протокола FTP или TFTP зависит от нескольких факторов.

- Наличие данных протоколов на сервере или рабочей станции (зависит от системного администратора). Например, если на сервере или рабочей станции протокол TFTP не установлен, то для передачи придется воспользоваться протоколом FTP.
- Тип сетевого соединения между сервером (рабочей станцией) и устройством под управлением IOS. Например, если сервер или рабочая станция включены непосредственно в сеть, в которую подключен и маршрутизатор, то протокол TFTP будет функционировать нормально, и время передачи данных не будет слишком велико. Если же между сервером или рабочей станцией расположено одно или несколько устройств, то протокол FTP будет работать лучше, сокращая время передачи образов ОС IOS из сервера в устройство.
- Уровень безопасности, который необходимо сохранять при передаче образов ОС IOS из сервера. Протокол TFTP не требует для передачи никакой идентификации или аутентификации. В протоколе FTP для начала передачи информации достаточно ввести имя пользователя и пароль.

Примечание

Для определения того, какой именно образ ОС IOS исполняется на ваших маршрутизаторах, настоятельно рекомендуем связаться с местной службой технической поддержки компании.

Использование для передачи образов ОС IOS протокола TFTP

До начала процесса передачи образа ОС IOS на устройство необходимо убедиться в его наличии на TFTP-сервере. Если это так, следует воспользоваться командой copy tftp flash. В примере, приведенном ниже, показан процесс копирования образа ОС IOS c-2500-i-1.120-5.P.bin в маршрутизатор, находящийся в Сингапуре. Стоит отметить, что до запроса подтверждения копирования маршрутизатор показывает текущее содержимое флэш-памяти, а затем просит ввести IP-адрес TFTP-сервера и имя файла, содержащего образ ОС IOS. На последнем этапе устройство проверяет, прошла ли загрузка файла без ошибок.

```
Singapore#copy tftp flash
System flash directory:
File Length Name/status
Memory Configuration Issues 33
1 2980876 c2500-is-mz.lll-3.P.bin
[2980876 bytes used, 5407732 available, 8388608 total]
IP address or name of remote host [255.255.255.255]? 131.108.20.45
Name of file to copy ? c2500-i-1.120-5.P.bin
Copy c2500-i-1.120-5.P.bin from 131.108.20.45 into flash memory? [confirm]
Loading from 131.108.20.45:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!! [OK - 1906676/4194240 bytes] Verifying via checksum...
Flash verification successful. Length = 1906676, checksum = Oxl2AD
```

Примечание

Все команды режима EXEC, которые используются сетью для выполнения какого-либо действия, возвращают символ восклицательного знака (!) при успешном своем выполнении и точку (.) в противном случае.

Если требуется провести обратный процесс, т.е. скопировать образ ОС IOS из флэш-памяти на TFTP-сервер, то необходимо воспользоваться командой режима EXEC copy flash

tftp. Рекомендуется держать копии всех образов ОС 1OS на сервере и регулярно производить их резервное копирование. При обновлении версии операционной системы наличие на сервере копии последнего, исправно работавшего в сети, образа ОС IOS является обязательным правилом. Эта предосторожность позволит при возникновении непредвиденных проблем с новой версией операционной системы воспользоваться командой copy tftp flash и вернуться к работающему образу ОС IOS.

Содержимое флэш-памяти можно просмотреть в любой момент, воспользовавшись командой режима EXEC show flash:

```
Singapore>show flash
System flash directory:
File Length Name/status
1 1906676 c2500-i-1.120-5.bin
[1906676 bytes used, 6481932 available, 8388608 total]
8192K bytes of processor board System flash
```

Примечание

Некоторые устройства Cisco работают с образом ОС IOS, находящимся непосредственно во флэш-памяти, и не могут осуществлять его перезапись, когда он исполняется. Для копирования образов ОС IOS с TFTP-сервера такие устройства используют систему под названием Flash load helper (помощник загрузки флэш-памяти).

Использование для передачи образов ОС IOS протокола FTP

В отличие от протокола TFTP, протокол FTP требует ввести имя пользователя и пароля для идентификации и аутентификации как устройства, работающего под управлением ОС IOS, так и его администратора. Это делается, чтобы определить, есть ли у них разрешение на работу с FTP-сервером еще до начала передачи образа ОС IOS. Для предоставления имени пользователя и пароля при передаче файлов используются два метода.

- Ввод имени пользователя и пароля в качестве части команды режима EXEC copy ftp.
- Предварительное задание имени пользователя и пароля с помощью команд глобального конфигурирования ip ftp username и ip ftp password.

Первый метод следует использовать, если модификацию версий программного обеспечения на маршрутизаторе выполняют различные люди. Второй метод полезен, если обновление версии выполняет только один человек, или для экстренной передачи образов ОС 1OS на сервере используется специальная процедура регистрации и специальный пароль. Но в любом случае соответствующие имена и пароли должны быть на FTP-сервере до начала процесса передачи. В приведенных ниже примерах будут использоваться имя пользователя joebob и пароль getmysoftware.

Как и при использовании протокола TFTP, перед тем, как начать передачу образа ОС IOS, необходимо убедиться в его наличии на FTP-сервере. Если это уже сделано, то для предоставления имени пользователя и пароля в целях аутентификации и для инициации процесса передачи следует воспользоваться привилегированной командой режима EXEC copy ftp://username:password flash. Если подставить выбранные ранее имя пользователя и пароль, то команда примет вид copy ftp://joebob: getmysoftware flash. В приведенном ниже примере осуществляется процесс записи файла образа ОС IOS c2500-i-l. 120-5.P.bin во флэш-память маршрутизатора, находящегося в Сингапуре. Напомним, что маршрутизатор До запроса о подтверждении копирования показывает текущее содержимое флэш-памяти, а затем просит ввести IP-адрес FTP-сервера и название файла, содержащего образ ОС IOS. Как вариант, IP-адрес FTP-сервера и имя файла образа ОС IOS могут вводиться аналогично имени пользователя и паролю в качестве части команды copy В виде ftp://username:passwordgftpservername/ios-image-name. Как и в предыдущем примере, на последнем этапе устройство проверяет, прошла ли загрузка файла без ошибок.

```
Singapore#copy ftp://joebob:getmysoftware flash
System flash directory: File Length Name/status
1 2980876 c2500-is-mz.lll-3.P.bin
[2980876 bytes used, 5407732 available, 8388608 total]
IP address or name of remote host [255.255.255.255]? 131.108.20.45
Name of file to copy ? c2500-i-1.120-5.P.bin
Copy c2500-i-1.120-5.P.bin from 131.108.20.45 into flash memory? [confirm] Loading
from 131.108.20.45:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!    [OK  -   1906676/4194240 bytes]
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful.   Length =  1906676,   checksum = Oxl2AD
```

Как уже говорилось ранее, имя пользователя и пароль для доступа к FTP-серверу могут быть занесены в исполняемую конфигурацию до инициализации процесса передачи образа ОС IOS и, как вариант, могут храниться в стартовой конфигурации для будущего использования. Имя FTP-пользователя и пароль задаются с помощью команд глобального конфигурирования ip ftp username и ip ftp password. В примере ниже выполняется процесс ввода имени FTP-пользователя joebob и пароля getmysoftware, а затем инициализируется передача образа ОС IOS:

```
Singapore#configure terminal
Singapore(config)#ip ftp username joebob
Singapore(config)#ip ftp password getmysoftware
Singapore(config)#^Z
Singapore#copy ftp flash
System flash directory:
File Length Name/status
1 2980876 c2500-is-mz.lll-3.P.bin
[2980876 bytes used, 5407732 available, 8388608 total]
IP address or name of remote host [255.255.255.255]? 131.108.20.45
Name of file to copy ? c2500-i-1.120-5.P.bin
Copy c2500-i-1.120-5.P.bin from 131.108.20.45 into flash memory? [confirm]
Loading from 131.108.20.45:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!! [OK - 1906676/4194240 bytes] Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvFla
sh verification successful. Length = 1906676, checksum = Oxl2AD
```

Примечание

Все вопросы, касающиеся методов и режимов конфигурирования ОС IOS, рассматриваются в следующем разделе "Пользовательский режим конфигурирования".

Как и в предыдущем примере, маршрутизатор в Сингапуре перед запросом подтверждения копирования показывает текущее содержимое флэш-памяти, а затем запрашивает IP-адрес FTP-сервера и имя файла, содержащего образ ОС IOS.

Как и при использовании протокола TFTP, возможна обратная операция — копирование содержимого флэш-памяти на FTP-сервер с помощью команды режима EXEC copy flash ftp.

Как и в предыдущем процессе, следует ввести имя пользователя и пароль, необходимые для FTP-передачи. Эти данные могут быть введены как часть команды либо указаны в рабочей конфигурации. Вне зависимости от типа используемого протокола передачи рекомендуется копировать все образы ОС IOS для своей сети на сервер и регулярно делать их резервные копии. При обновлении версии операционной системы наличие на сервере копии последнего, исправно работавшего в сети образа ОС IOS, является обязательным правилом. Эта предосторожность позволит в случае непредвиденных проблем с новой версией операционной системы воспользоваться командой copy ftp flash и вернуться к работающему образу ОС IOS.

Управление свободным пространством флэш-памяти

Все команды, используемые для перезаписи во флэш-память образов ОС IOS, выполняют оценку свободного пространства в памяти и, если это необходимо для высвобождения дополнительного объема памяти, предлагают стереть или сжать предыдущее содержимое флэш-памяти. С другой стороны, возможны ситуации, при которых надо стереть все содержимое флэш-памяти или его часть вне зависимости от процесса передачи. Все содержимое флэш-памяти можно стереть с помощью привилегированной команды режима EXEC erase flash. Чтобы из флэш-памяти стереть конкретный файл, нужно использовать команду delete. Например, для удаления из флэш-памяти файла образа ОС IOS c2500-i-1.120.P.bin вводится привилегированная команда режима EXEC delete c2500-i-1.120.P.bin. В устройствах Cisco, оснащенных внешними картами флэш-памяти (обычно устанавливаемыми в слот, который называется slotO), команда delete не стирает файл, а только помечает его как файл, доступный для удаления, и соответственно не высвобождает пространство флэш-памяти. Для завершения процесса удаления файла необходимо исполнить команду squeeze.

Пользовательский режим конфигурирования

Для конфигурирования устройства, работающего под управлением IOS, следует использовать привилегированную команду режима EXEC configure. Эта команда имеет три варианта:

- конфигурирование с терминала;
- конфигурирование из памяти;
- конфигурирование через сеть.

Совет

В привилегированном режиме EXEC, непривилегированном режиме EXEC и в пользовательском режиме конфигурирования ОС IOS позволяет повторять команды без их повторного набора на клавиатуре. Для этого необходимо подняться или опуститься до требуемой команды в списке уже использовавшихся команд и нажать клавишу <Enter>. Данная команда повторится в текущей командной строке. В большинстве терминалов клавиша <T> позволяет двигаться вверх по списку, а клавиша <•!> — вниз по списку. Если же данные клавиши не работают, то для движения вверх по списку можно использовать комбинацию клавиш <Ctrl+P>, а для движения вниз — комбинацию <Ctrl+N>.

При вводе команды configure ОС IOS просит указать тот ее вариант, который будет использоваться:

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

Вариант по умолчанию, который стоит первым в перечне, позволяет осуществлять конфигурирование устройства в реальном времени с использованием терминала. Команды выполняются ОС IOS сразу после их введения:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#
```

После выполнения команды система изменяет вид командной строки, показывая, что она находится в режиме конфигурирования и позволяет вводить команды конфигурирования. По окончанию набора команды вводится комбинация клавиш <Ctrl+Z> (AZ). В приведенном ниже примере с помощью команды глобального конфигурирования hostname имя Singapore изменяется на Seoul:

```
Singapore #configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#hostname Seoul
Seoul(config)#^Z
Seoul!
```

В данном примере команда исполняется немедленно, и имя устройства изменяется. Таким образом, для активации команд не нужно текущую конфигурацию переносить в стартовую.

Второй вариант команды — настройка из памяти — позволяет копировать хранящуюся в энергонезависимой памяти стартовую конфигурацию устройства в ОЗУ, где находится текущая конфигурация. Этот вариант полезен в тех случаях, когда после изменения в реальном времени какого-либо конфигурационного параметра необходимо вернуться к стартовой конфигурации. В данном случае команда configure выполняет те же самые функции, что и команда copy startup-config running-config, которая была описана в предыдущем разделе:

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

Третий вариант — настройка по сети — позволяет загружать файл конфигурации с TFTP-сервера:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]? network
Host or network configuration file [host]?
Address of remote host [255.255.255.255]? 131.108.20.45
Name of configuration file [singapore-confg]?
Configure using singapore-confg from 131.108.20.45? [confirm]
Loading singapore-confg !![OK]
Singapore#
```

Во всех приведенных выше примерах команды configure предлагаемые ОС IOS значения по умолчанию (показаны в квадратных скобках) принимались нажатием клавиши возврата каретки в ответ на вопрос.

TFTP представляет собой протокол, который позволяет ОС IOS запрашивать конкретный файл с TFTP-сервера. Протокол TFTP использует IP-протокол, и поэтому для нормальной работы этого варианта команды необходимо иметь настроенную и работающую IP-маршрутизацию между устройством и TFTP-сервером. Более подробная информации о конфигурировании IP-маршрутизации протокола IP дается в главе 4.

Когда конфигурирование устройства с ОС IOS производится с TFTP-сервера, оно по умолчанию пытается загрузить файл, название которого состоит из имени устройства, за которым следует цепочка символов -confg. В примере ниже устройство с именем Singapore

безуспешно пытается загрузить по умолчанию файл singapore-confg:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]? network
Host or network configuration file [host]?
Address of remote host [255.255.255.255]? 131.108.20.45
Name of configuration file [singapore-confg]?
Configure using singapore-confg from 131.108.20.45? [confirm]
Loading singapore-confg ... [timed out]
Singapore#
```

Устройство может потерпеть неудачу при загрузке файла конфигурации из-за проблем со взаимодействием в IP-сети или из-за нарушений правил протокола TFTP.

Команды конфигурирования

Команды конфигурирования используются для формирования конфигурации устройства. Как было показано в предыдущем разделе, эти команды могут вводиться с терминала, загружаться из стартовой конфигурации или сгружаться в виде файла с использованием протокола TFTP и команды configure. Все команды конфигурирования должны вводиться в устройство, которое находится в режиме конфигурирования, а не в режиме исполнения команд EXEC. Команда конфигурирования, введенная в командной строке с именем устройства, считается неправильной и не воспринимается:

```
Singapore#hostaame ^ Seoul
%   Invalid  input  detected at   '^'   marker
```

Команда, введенная в режиме конфигурирования, — верна.

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)thostname Seoul
Seoul(config)#^Z
Seoul!
```

Все команды ОС IOS делятся на три категории:

- глобальные команды;
- основные команды;
- подкоманды.

Глобальными называются команды, действие которых распространяется на всю ОС IOS. Примером таких команд являются рассмотренные в этой главе команды hostname, enable secret и ip routing. Эти команды были использованы В скрипте команд конфигурирования, созданном диалогом конфигурирования системы. Применение любой из этих команд вносит изменения в конфигурацию ОС IOS, не требуя при этом использования дополнительных команд. Например, команда hostname задает имя устройства, команда enable secret определяет пароль, который будет использоваться при входе в привилегированный режим, а команда ip routing включает IP-маршрутизацию.

Основные команды позволяют подкомандам конфигурировать устройство. Сами эти команды не вносят изменений в конфигурацию устройства. В приведенном ниже примере основная команда interface EthernetO сообщает ОС IOS о том, что последующие подкоманды будут относиться непосредственно к интерфейсу локальной сети с именем EthernetO. В этом примере подкоманда ip address назначает IP-адрес интерфейсу EthernetO:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interface EthernetO
Singapore(config-if)#ip address 131.108.1.1 255.255.255.128
Singapore(config-if)#^Z
```

```
Singapore#
```

Как уже было сказано, в данном примере ОС IOS восприняла команду interface EthernetO как основную. ОС сообщает об этом, изменяя заголовок командной строки с Singapore (config) на Singapore (config-if) . Тем самым она указывает на то, что последующие команды являются подкомандами и будут относиться к интерфейсу. Сама команда interface EthernetO не конфигурирует устройство — для этого она должна быть дополнена подкомандами.

Основные команды требуют четкого соответствия с контекстом подкоманд. Например, основная команда ip address 131.108.1.1 255.255.255.128 для правильной интерпретации требует указания конкретного интерфейса. Комбинация основной команды с подкомандой позволяет конфигурировать устройство.

Что касается ОС IOS версии 12.0, то в ней для некоторых основных команд существует дополнительный уровень подкоманд конфигурирования. Например, при конфигурировании ATM-интерфейса, который будет рассматриваться в главе 3, с помощью основной команды interface atmO задается интерфейс для настройки. Затем с помощью подкоманды pvc [name] vpi/vci для этого интерфейса может указываться идентификатор виртуального пути (vpi) и идентификатор виртуального канала (vci). Эта подкоманда имеет свою подкоманду дополнительного уровня, которая позволяет указать качество ATM-сервиса, ассоциируемого со значением VPI/VCI. Скажем, в примере ниже для ATM-интерфейса значение VPI/VCI устанавливается равным 5/42 при передаче с заранее не заданной скоростью (unspecified bit rate - UBR) в 384 Кбит/с:

```
Router#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Router(config)#interface atmO
Router(config-if)#pvc 5/42
Router(config-if)#ubr 384
Router(config-if)#^Z
Router#
```

Эта конфигурация в маршрутизаторе будет иметь следующий вид (показана лишь та часть экрана конфигурации, которая имеет отношение к рассматриваемому вопросу):

```
Router#show running-config
!
Current configuration:
interface ATMO
  pvc 5/42
  ubr 384
```

Как было показано в предыдущем разделе, конфигурирование устройства, работающего под управлением ОС IOS, может осуществляться с использованием файла конфигурации, загружаемого по протоколу TFTP с помощью команды configure с опцией network. Этот файл должен быть текстовым и содержать требуемые для конфигурирования устройства глобальные и основные команды с подкомандами. В процессе загрузки файла конфигурации устройство сразу же интерпретирует команды конфигурирования и исполняет их. Все происходит точно так же, как если бы эти команды вводились с помощью команды configure с опцией terminal.

Помощь в процессе конфигурирования

Встроенная в ОС IOS система помощи доступна и при конфигурировании устройства. Для получения списка имеющихся опций конфигурирования достаточно в любое время в процессе конфигурирования ввести команду в виде знака вопроса (?). В представленном ниже примере эта функция осуществляет поиск глобальных команд, доступных в режиме конфигурирования:

```
Singapore(config)#?
Configure commands:
aaa                          Authentication, Authorization and Accounting
access-list                  Add an access list entry
alias                        Create command alias
arp                          Set a static ARP entry
async-bootp                  Modify system bootp parameters
banner                       Define a login banner
boot                         Modify system boot parameters
bridge                       Bridging Group
buffers                      Adjust system buffer pool parameters
busy-message                 Display message when connection to host fails
cdp                          Global CDP configuration subcommands
chat-script                  Define a modem chat -script
clock                        Configure time-of-day clock
config-register              Define the configuration register
default-value                Default character-bits values
dialer-list                  Create a dialer list entry
dnsix-dmdp                   Provide DMDP service for DNSIX
dnsix-nat                    Provide DNSIX service for audit trails
downward-compatible- config  Generate a configuration compatible with older software
enable                       Modify enable password parameters

–More–
```

В показанном примере для экономии места приведена не вся выводимая информация.

Встроенная система помощи может также использоваться для получения списка подкоманд, которые можно вводить при вводе той или иной команды. В следующем примере осуществляется поиск подкоманд, доступных при конфигурировании интерфейса EthernetO на использование протокола IP:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interface EthernetO
Singapore(config-if)#ip ?
Interface IP configuration subcommands:
access-group              Specify access control for packets
accounting                Enable IP accounting on this interface
address                   Set the IP address of an interface
bandwidth-pe rcent        Set EIGRP bandwidth limit
broadcast-address         Set the broadcast address of an interface
directed-broadcast        Enable forwarding of directed broadcasts
gdp                       Gateway Discovery Protocol
hello-interval            Configures IP-EIGRP hello interval
helper-address            Specify a destination address for UDP broadcasts
hold-time                 Configures IP-EIGRP hold time
irdp ICMP                 Router Discovery Protocol
mask-reply                Enable sending ICMP Mask Reply messages
mobile                    Mobile Host Protocol
mtu                       Set IP Maximum Transmission Unit
policy                    Enable policy routing
probe                     Enable HP Probe support
proxy-arp                 Enable proxy ARP
rarp-server               Enable RARP server for static arp entries
redirects                 Enable sending ICMP Redirect messages
rip                       Router Information Protocol
route-cache               Enable fast-switching cache for outgoing packets

–More–
```

Этот список, как и предыдущий, сокращен.

Удаление команд конфигурирования

Для удаления команды конфигурирования из устройства в начало команды конфигурирования добавляется ключевое слово no. В примере ниже показано удаление IP-адреса, присвоенного интерфейсу EthernetO:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interface EthernetO
Singapore(config-if)#no ip address 131.108.1.1 255.255.255.0
Singapore(config-if)#^Z
Singapore#
```

Для удаления любой команды (глобальной, основной или подкоманды) необходимо следовать этой же процедуре.

Команды конфигурирования, используемые по умолчанию

Команды конфигурирования, используемые ОС IOS по умолчанию, не показываются в результатах исполнения команд show running-config или show startup-config. Если ввести такую команду, то устройство воспримет ее и не выдаст сообщения об ошибке. Например, как будет показано в следующей главе, все интерфейсы последовательной передачи данных в маршрутизаторах Cisco по умолчанию используют инкапсуляцию по высокоуровневому протоколу управления каналом передачи данных (High-Level Data Link Control — HDLC). Соответственно, ввод подкоманды конфигурирования интерфейса encapsulation hdlc при конфигурировании последовательного интерфейса не приведет к появлению новой строки в конфигурации маршрутизатора.

Все команды ОС IOS также имеют значение по умолчанию. Для возврата значения любой глобальной, основной команды или подкоманды в значение, принимаемое ею по умолчанию, эта команда предваряется командой конфигурирования default. Многие команды ОС IOS по умолчанию действуют противоположно их прямому действию, и поэтому использование этих команд со значением по умолчанию равносильно их использованию в форме с ключевым словом **no** впереди, которая была показана в предыдущем разделе. Например, представленная ниже конфигурация приведет к удалению IP-адреса, присвоенного интерфейсу EtehernetO маршрутизатора в Сингапуре:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interfaoe EthernetO
Singapore(config-if)#default ip address
Singapore(config-if)#*Z
Singapore#
```

Однако некоторые команды по умолчанию имеют конкретную конфигурацию. В таких случаях команда default приводит к принятию командой конфигурирования ее значения по умолчанию.

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#default hostname
Singapore(config-if}#^Z
Router!
```

В этом примере команде hostname разрешается присвоить устройству имя по умолчанию, каковым является имя "Router".

Слияние и замещение команд конфигурирования

Новая команда конфигурирования может замещать старую. В этом случае ОС IOS автоматически удаляет старую команду. С другой стороны, новая команда может не замещать, а сливаться с уже существующей командой. В качестве примера слияния команд можно привести случай использования двух команд snmp-server. Представим, что создается следующая конфигурация:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#snmp-server community public
Singapore(config)#^Z
Singapore#
```

После этого принимается решение о замене конфигурации команды snmp-server следующей:

```
Singapore#configure
Enter configuration commands, one per line. End with CTRL+Z.
Configuring from terminal, memory, or network [terminal]?
Singapore(config)#srmp-server community zipnet
Singapore(config)#^Z
Singapore#
```

Поскольку возможно использование нескольких команд snmp-server, вторая команда snmp-server сливается с существующей конфигурацией, и обе команды являются активными, что и показывает соответствующая часть результата исполнения команды show running-config:

```
!
snmp-server community public
snmp-server community zipnet
!
```

Для того чтобы заменить первую команду конфигурирования snmp-server второй, необходимо выполнить следующие действия:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#no snmp-server community public
Singapore(config)#snmp-server community zipnet
Singapore(config)#^Z

Singapore#
```

Примером несливающейся команды является команда hostname, которая устанавливает имя устройства. В примере ниже маршрутизатору в Сингапуре присваивается новое имя:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#hostname Sing-router
Sing-router(config)#^Z

Sing-router#
```

Команда hostname сразу после введения меняет предыдущую конфигурацию. Результат исполнения команды show running-config показывает наличие в конфигурации только одной команды hostname:

```
!
hostname  Sing-router
!
```

Следует помнить об этой особенности ОС IOS при добавлении новых команд в существующую конфигурацию.

Резюме

Данная глава посвящена основным командам и этапам конфигурирования, а следующая — конфигурированию интерфейсов. Следует запомнить ключевые моменты конфигурирования устройств.

- Для задания пароля входа в привилегированный режим ОС IOS рекомендуется использовать команду enable secret.
- Система встроенной помощи в режиме EXEC предоставляет информацию о том, какие команды доступны в данном режиме, что они делают, и какие опции их выполнения существуют (табл. 2.1). Система помощи также доступна в режиме конфигурирования устройства.
- В непривилегированном режиме можно лишь получать информацию о состоянии устройства, но нельзя изменять его параметры. В привилегированном режиме можно устанавливать и изменять параметры устройства.
- Оперативная и энергонезависимая память — два типа памяти, используемые устройством для хранения команд конфигурирования ОС IOS. Текущая конфигурация хранится в оперативной памяти. При потере питания содержащаяся в ней информация стирается. Информация о конфигурации, сохраняемая в энергонезависимой памяти, не стирается при отключении питания; к этой конфигурации устройство возвращается после возобновления
  подачи питания.
- Конфигурирование устройства возможно в реальном времени с терминала, из энергонезависимой памяти или по сети.
- Для того чтобы убрать команду из конфигурации устройства, необходимо ввести перед ней ключевое слово no.
- При вводе дополнений в существующую конфигурацию устройства некоторые команды конфигурирования сливаются с существующими, а другие замещают их (табл. 2.2).

Таблица 2.1. Сводная таблица команд режима EXEC для создания базовой конфигурации устройства

Команда	Описание
`configure`	Задает режим конфигурирования устройства с терминала, из памяти или по сети
`copy flash ftp`	Копирует файл образа ОС IOS из флэш-памяти на FTP-сервер
`copy flash tftp`	Копирует файл образа ОС IOS из флэш-памяти на TFTP-сервер
`copy ftp flash`	Копирует файл образа ОС IOS во флэш-память устройства с FTP-сервера
`copy running-config startup-config`	Сохраняет текущую настройку в энергонезависимой памяти
`copy startup-config running-config`	Делает стартовую конфигурацию текущей
`copy tftp flash`	Копирует файл образа ОС IOS во флэш-память устройства с TFTP-сервера
`delete` *`имя образа ОС IOS`*	Стирает указанный файл образа ОС IOS из флэш-памяти устройства
`disable`	Осуществляет выход из привилегированного режима и вход в непривилегированный режим
`enable`	Осуществляет вход в привилегированный режим Стирает всю флэш-память устройства
`erase flash`	
`erase startup-config`	Стирает начальную конфигурацию
`lock`	Блокирует текущий сеанс связи терминала
`session` `название модуля`	Устанавливает сеанс связи с указанным модулем
`show flash`	Показывает содержимое флэш-памяти
`show running-config`	Показывает текущую настройку устройства
`show sessions`	Показывает текущие сеансы связи терминала
`show startup-config`	Показывает начальную конфигурацию устройства

squeeze	Стирает файл, находящийся во флэш-памяти и имеющий метку для уничтожения

Таблица 2.2. Сводная таблица команд конфигурирования для создания базовой конфигурации устройства

Команда	Описание
default команда	Придает команде значение, принятое по умолчанию
enable password пароль	Устанавливает пароль для входа в привилегированный режим
enable secret пароль	Определяет односторонне шифрованный пароль для входа в привилегированный режим
hostname	Задает имя устройства
interface тип	Устанавливает тип конфигурируемого интерфейса
ip ftp password	Устанавливает пароль для аутентификации при использовании протокола FTP для передачи образов ОС IOS и других действий с помощью этого протокола
ip ftp usernante	Задает имя пользователя для выполнения идентификации при использовании протокола FTP для передачи образов ОС IOS и других действий с помощью этого протокола
no команда	Отменяет действие команды конфигурирования

Дополнительная литература

Конкретная документация по продуктам Cisco может быть получена на Web-сервере компании Cisco Systems, Inc. в разделе технической документации на изделия компании Cisco Product Documentation по адресу: www.cisco.com/univercd/cc/td/ doc/product/index.htm.

Глава 3

Основы интерфейсов устройств Cisco

Ключевые темы этой главы

- **Базовое конфигурирование интерфейсов.** Основы конфигурирования интерфейсов устройств, работающих под управлением Cisco 1OS
- **Технологии локальных сетей.** Краткий обзор технологий локальных сетей, поддерживаемых устройствами Cisco, включая Ethernet/IEEE 802 3, Fast Ethernet, Gigabit Ethernet, Token Ring/IEEE802.5 и FDDI.
- **Технологии глобальных сетей и технологии взаимодействия по коммутируемым каналам связи** Краткий обзор технологий глобальных сетей и технологий взаимодействия по коммутируемым каналам связи, поддерживаемых устройствами Cisco, включая HDLC, PPP, X 25, Frame Relay, ATM, DSL и ISDN

В этой главе рассматриваются основы технологий и конфигурирования различных типов интерфейсов, поддерживаемых устройствами Cisco. Для подробного изучения выбраны пять широко используемых технологий локальных сетей и семь технологий глобальных сетей.

Базовое конфигурирование интерфейсов

Термин *интерфейс* обозначает соединение устройства со средой передачи данных? В основе каждого интерфейса лежат технологии, которые определяют принципы передачи данных через физическую среду, например, медный или оптоволоконный бель. Протоколы, используемые на физическом уровне передачи данных модели, определяют физические характеристики интерфейса и среды передачи данных, протоколы, речь о которых будет идти в данной главе, работают на втором уровне дели OSI — канальном. Именно с помощью этих протоколов осуществляется пере; данных между сетевым и физическим уровнями модели OSI.

В устройствах Cisco каждый интерфейс называется *портом* и обозначается сколькими способами. В устройствах с фиксированной конфигурацией интерфейсы нумеруются последовательно без привязки к слоту, в котором они установлены, пример, в маршрутизаторе серии 2500 с одним интерфейсом Ethernet и двумя последовательными интерфейсам эти интерфейсы обозначаются ethernetO, serialO и ser соответственно.

В устройствах модульной конструкции с заменяемыми платами интерфейсов интерфейсы нумеруются с использованием синтаксиса типа слот/порт. Например, рой порт платы интерфейса Ethernet, установленной в первый слот, будет иметь ethernet 1/2. Для конфигурирования интерфейсов используется основная команда interface. Эта команда с указанием после нее номера порта интерфейса или кои нации слот/порт используется в режиме конфигурирования. Приведенный ниже пример иллюстрирует конфигурирование порта 0 интерфейса Token Ring в слоте 1:

```
San-Jose# configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
San-Jose(config)#interface tokenring 1/0
San-Jose(config-if)#^Z
```

Примечание

Для того чтобы показать пользователю, что он находится в режиме настройки интерфейса, ОС IOS изменяет вид заголовка командной строки с **config на config-if.** В режиме конфигурирования ОС IOS часто меняет заголовок командной строки. Это дает пользователю визуальную подсказку в процессе конфигурирования.

В некоторых маршрутизаторах Cisco применяются платы универсального процессора интерфейса (Versatile Interface Processor — VIP). Каждая VIP-плата имеет один или два слота для адаптеров порта. Адаптер порта — это печатная плата, на которой установлены интерфейсы, и которая, в свою очередь, устанавливается на VIP-плату. Каждый адаптер порта может содержать несколько интерфейсов. В устройствах этого типа (на данный момент возможностями для установки VIP-плат обладают лишь маршрутизаторы Cisco серий 7000, 7500 и 12000) для задания интерфейса используется синтаксис вида **номер слота расширения/номер адаптера порта/номер порта.** Например, чтобы обратиться к первому порту Token Ring первого адаптера порта (с номером 0), который, в свою очередь, установлен на VIP-плате с номером 2, следует использовать синтаксис **token ring 2/0/1.**

Команда show interfaces

Команда режима EXEC show interfaces позволяет просмотреть статус всех интерфейсов устройства Cisco, как показано в примере для интерфейса Ethernet:

```
EthernetO is up, line protocol is up
   Hardware is QUICC Ethernet, address is 0060.5cbc.Oef9  (bia
   0060.5cbc.0ef9)
```

```
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
Encapsulation ARPA, loopback not set, keepalive set (10 sec)
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:00, output 00:00:01, 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 1000 bits/sec, 1 packets/sec
5 minute output rate 1000 bits/sec, 1 packets/sec
116547 packets input, 13397137 bytes, 0 no buffer
Received 3402 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 input packets with dribble condition detected
273769 packets output, 84816409 bytes, 0 underruns
0 output errors, 1 collisions, 1 interface resets
0 babbles, 0 late collision, 29 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out
```

В этой книге обсуждаются различные виды информации, выводимые на экран терминала командой show interfaces. Следует отметить, что в первой строке результата, выводимого этой командой, дается информация о типе среды интерфейса (Ethernet) и его номере. Исходя из того, что выведено имя интерфейса ethernetO, можно сделать вывод, что это устройство с фиксированной конфигурацией; номера слотов, портов и адаптеров портов отсутствуют. Интерфейс, приведенный в примере, находится в активном состоянии, и его электронная часть функционирует нормально, получая соответствующие сигналы из подключенных к нему кабелей. Еще интерфейс может быть в состоянии отключения и в состоянии административной блокировки. Различие между административной блокировкой и отключением состоит в следующем: отключенный интерфейс находится в рабочем состоянии, но не обменивается данными с подключенной к нему средой. В состоянии административной блокировки интерфейс отключен на уровне конфигурации. Информация об изменении административного состояния интерфейса дается в следующем разделе "Команда shutdown".

Во второй строке результата, выводимого командой show interfaces, указывается название модели интерфейса и его адрес, используемый протоколом канального уровня. Четвертая строка показывает тип инкапсулирования для данного интерфейса. Обычно инкапсулирование данных для интерфейса локальной сети не нуждается в конфигурировании, тогда как интерфейсы глобальных сетей часто этого требуют. Такое различие объясняется тем, что интерфейсы локальных сетей, как правило, используют один тип протокола канального уровня, а интерфейсы глобальных сетей могут работать с несколькими различными протоколами канального уровня.

Команда encapsulation

Тип инкапсулирования интерфейса определяет формат передаваемых данных и тип протокола канального уровня для этого интерфейса. Тип инкапсулирование интерфейса устанавливается с помощью подкоманды конфигурирования интерфейса encapsulation. В приведенном ниже примере сначала с помощью системы помощи выясняются типы инкапсулирования, доступные для интерфейса глобальной сети serialO, а затем этот интерфейс конфигурируется на применение протокола HDLC:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z..
Singapore(config)#interface serial 0
Singapore(config-if)#encapsulation ?
atm-dxi     ATM-DXI encapsulation
frame-relay Frame Relay networks
hdlc        Serial HDLC synchronous
lapb        LAPB (X.25 Level 2)
ppp        Point-to-Point protocol
smds        Switched Megabit Data Service (SMDS)
x25         X.25
```

```
Singapore(config-if)#encapsulation hdlc
Singapore(config-if)# ^Z
```

Ниже в этой главе рассматриваются и другие типы протоколов инкапсулирование для интерфейсов глобальных сетей.

Команда shutdown

Команды конфигурирования shutdown или no shutdown применяются, если необходимо изменить административное состояние интерфейса — блокировать или, наоборот, включить его. Устройство Cisco не передает данные на интерфейс, если он заблокирован на административном уровне. В приведенном ниже примере данных, выводимых командой show interfaces, первая строка говорит о том, что интерфейс serial0 заблокирован на административном уровне:

```
Serial0 is administratively down, line protocol is down
  Hardware is 4T/MC68360
  MTU 1500 bytes, BW 512 Kbit, DLY 20000 usec, rely 137/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/64/0 (size/threshold/drops)
     Conversations 0/1 (active/max active)
     Reserved Conversations 0/0 {allocated/max allocated)
  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 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 packets output, 0 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
  DCD=down DSR=down DTR=down RTS=down CTS=down
```

В следующем примере с помощью команды конфигурирования no shutdown интерфейс переводится во включенное состояние:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interface serial 0
Singapore(config-if)#no shutdown
Singapore(config-if)# ^Z
```

Примечание

Команда конфигурирования ОС IOS no shutdown зачастую приводит пользователей в замешательство. С одной стороны, она вроде бы говорит устройству не отключать интерфейс, подразумевая, что он включен. Однако здесь имеет место двойное отрицание, которое означает, что интерфейс должен быть включен. Это неловкое использование английского языка было оставлено в ОС IOS по чисто историческим (или истерическим) причинам.

Теперь, если подключенные к этому интерфейсу кабели обеспечивают подачу на него соответствующих сигналов, то он административно и операционно находится во включенном и рабочем состоянии. Для административного блокирования интерфейса команда shutdown используется следующим образом:

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

```
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interface serial 0
Singapore(config-if)#shutdown
Singapore(config-if)# ^Z
```

Команда *description*

Для улучшения информативности с помощью интерфейсной подкоманды description можно добавлять текстовый комментарий, который будет выводиться на экран командой show interfaces. Длина этого комментария не может быть более 255 символов.

Совет

Рекомендуется добавлять комментарий к каждому из установленных интерфейсов с указанием его назначения. Например, в комментарий интерфейса локальной сети можно добавить название здания, этажа или отдела, связь с которыми обеспечивает данный интерфейс. Для интерфейса глобальной сети можно ввести описание конечных пунктов соединения и задокументировать идентификаторы используемых провайдером каналов.

В приведенном ниже примере иллюстрируется процесс добавления комментария для интерфейса глобальной сети serialO маршрутизатора сети компании ZIP в Сингапуре, который обеспечивает связь с Малайзией. Комментарий включает описание типа инкапсулирования и идентификатор канала:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interface serial 0
Singapore(config-if)#description IETF frame relay PVCs on Circuit Z-
234987-12-MS-01
Singapore(config-if)#^Z
```

Добавленный комментарий появляется в третьей строке результата исполнения команды **show interfaces serial 0:**

```
SerialO is administratively down, .line protocol is down
  Hardware is 4T/MC68360
  Description: IETF frame relay PVCs on Circuit Z-234987-12-MS-01
  MTU 1500 bytes, BW 512 Kbit, DLY 20000 usec, rely 137/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/64/0 (size/threshold/drops)
    Conversations 0/1 (active/max active)
    Reserved Conversations 0/0 (allocated/max allocated)
  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 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 packets output, 0 bytes, 0 underrurts
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
  DCD=down DSR=down DTR=down RTS=down CTS=down
```

Технологии локальных сетей

Устройства Cisco поддерживают множество различных технологий локальных сетей. В данной главе рассматриваются пять популярных технологий:

- Ethernet и IEEE 802.3;
- Fast Ethernet;
- Gigabit Ethernet;
- Token Ring;
- Fiber Distributed Data Interface.

Каждый из этих протоколов функционирует на канальном уровне модели OSI и используется в технологиях локальных сетей для передачи данных из одной точки в другую со скоростями от 4 Мбит/с до 1 Гбит/с. В этом разделе кратко описаны данные протоколы. Для тех, кто хочет изучить их более подробно, в конце главы приводится список дополнительной литературы.

Все описываемые протоколы локальных сетей используют одну и ту же схему адресации канального уровня. Адреса являются уникальными и имеют вид 6-байтовых чисел в шестнадцатеричной форме. Эти адреса называют адресами контроля доступа к среде (Media Access Control addresses) или MAC-адресами. Иногда их также называют аппаратными адресами, адресами станций или физическими адресами. Это означает, что каждое сетевое устройство имеет единственный во всем мире адрес канального уровня. MAC-адрес зашивается в полупостоянное запоминающее устройство (ППЗУ) с плавкими перемычками, располагающимися непосредственно на плате интерфейса.

Для того чтобы гарантировать уникальность адреса интерфейса, каждому производителю присваивается 20-битный префикс, используемый в дополнение к 6-байтовому адресу. Например, компании Cisco был присвоен 20-битный префикс 0060.5 (представлен в шестнадцатеричном формате, в котором каждая цифра представляет собой четыре бита). После этого производитель может определять оставшиеся 28 бит произвольным образом, но сохраняя уникальность MAC-адреса.

Уникальный MAC-адрес канального уровня для каждого интерфейса локальной сети от Cisco виден во второй строке результата, выводимого командой show interfaces. Ниже приведен пример, в котором можно увидеть MAC-адрес маршрутизатора компании ZIP, расположенного в Куала-Лумпуре:

```
Kuala-Lumpur>show interface ethernet0
Ethernet0 is up, line protocol is up
   Hardware is QUICC Ethernet, address is 0060.5cbc.0ef9 (bia
   0060.5cbc.0ef9)
   MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
   Encapsulation ARPA, loopback not set, keepalive set (10 sec)
   ARP type: ARPA, ARP Timeout 04:00:00
   Last input 00:00:00, output 00:00:01, 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 1000 bits/sec, 1 packets/sec
   5 minute output rate 1000 bits/sec, 1 packets/sec
      116547 packets input, 13397137 bytes, 0 no buffer
      Received 3402 broadcasts, 0 runts, 0 giants
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
      0 input packets with dribble condition detected
      273769 packets output, 84816409 bytes, 0 underruns
      65959 output errors, 1 collisions, 1 interface resets
      0 babbles, 0 late collision, 29 deferred
      65959 lost carrier, 0 no carrier
      0 output buffer failures, 0 output buffers swapped out
```

Примечание

Стоить отметить одну важную особенность ОС IOS. Технически она позволяет присваивать интерфейсу локальной сети адрес канального уровня, отличающийся от изначально прошитого в ППЗУ адреса. Подобная практика редка, но весьма полезна в некоторых сетях со сложной конфигурацией.

Технологии локальной сети Ethernet и IEEE802.3

Технология Ethernet и протокол IEEE 802.3, разработанный Институтом инженеров электротехники и электроники (Institute of Electrical and Electronic Engineers IEEE), являются наиболее широко распространенными на сегодняшний день. Протокол Ethernet был разработан в средине 70-х годов сотрудниками исследовательского центра фирмы Xerox в Пало-Альто (США). Позже, в 1978 году, эта технология бы стандартизирована компаниями Digital Equipment Corporation, Intel Corporation Xerox. После этого IEEE стандартизировал похожий протокол, который получил название IEEE 802.3. Различия в использовании полей кадра в протоколах Ethernet IEEE 802.3 незначительны.

Примечание

Названия многих протоколов, рассматриваемых в данной книге, начинаются с цифры 802, которые обозначают год и месяц формирования соответствующих этим стандартам комиссий.

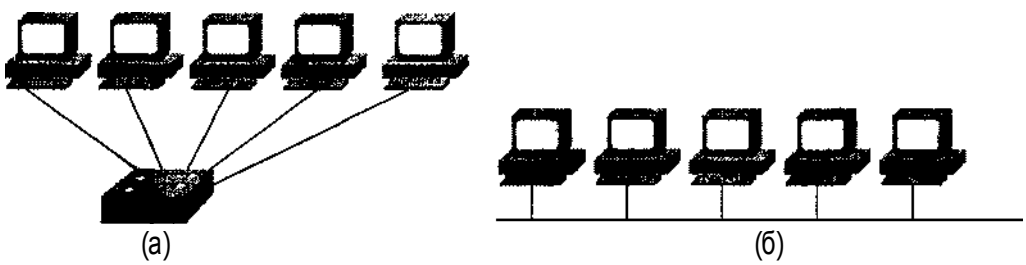
Оба вышеперечисленных протокола для доступа к общей шине с пропускной способностью 10 Мбит/с, по которой обмениваются данными все устройства в такой сети, используют технологию множественного доступа с контролем несущей и обнаружением конфликтов CSMA/CD. Устройства, подключенные к CSMA/CD-шине, мог проверять, идет ли в данный момент передача данных по шине (так называемый *контроль несущей),* а также контролировать, ведется ли одновременная передача двумя разными узлами *(обнаружение конфликтов).* Протокол CSMA/CD также определяет принципы взаимодействия устройств при обнаружении конфликтов.

Логически сегмент сети, построенный по технологии Ethernet или IEEE 802.3, имеет вид прямого отрезка провода, к которому подсоединены все устройства, как это показано на рис. 3.1.

Рис. 3.1. Сегмент сети Ethernet

Устройства, работающие по протоколу Ethernet или IEEE 802.3, могут обмениваться данными в режиме полудуплекса. Это режим, в котором устройство может передавать или принимать кадр, но не может вести одновременно и прием, и передачу данных. Обычно сегменты сети Ethernet или IEEE 802.3 работают в полудуплексном режиме. В рамках протоколов Ethernet и IEEE 802.3 существует также дуплексный режим работы. В этом режиме устройство может одновременно и принимать, и передавать кадр. Однако такой режим работы возможен, если только два устройства, использующие протокол Ethernet или IEEE 802.3, напрямую соединяются друг с другом. Примером такой топологии является соединение устройства с Ethernet-коммутатором.

Для логического объединения сегментов сетей Ethernet могут использоваться мосты и коммутаторы Cisco. Такое объединение осуществляется путем создания прозрачного, транслирующего или инкапсулирующего мостового соединения. В этой среде устройства Cisco соединяют между собой несколько сегментов локальной сети, создавая единый сегмент на канальном уровне, но с разделенными физическими CSMA/CD-сегментами или доменами конфликтов. На рис. 3.2 показаны как физическая, так и логическая топологии сегментов сети Ethernet, соединяемых с помощью мостов и коммутаторов Cisco.



(а) (б)

Рис. 3.2. Физическая (а) и логическая (б) топологии сегмента сети Ethernet

Для физического и логического разделения сегментов сети можно использовать маршрутизаторы Cisco. Каждый Ethernet-интерфейс имеет свой собственный адрес, и маршрутизатор будет перенаправлять пакеты между интерфейсами, основываясь на информации сетевого уровня.

Технология Fast Ethernet

Широкий успех технологии Ethernet и метода разделения доступа CSMA/CD привели к разработке технологии Fast Ethernet. Fast Ethernet представляет собой протокол CSMA/CD, который работает со скоростью 100 Мбит/с, что в десять раз быстрее, чем работа протоколов Ethernet или IEEE 802 3. Успех технологии Fast Ethernet объяснялся главным образом тем фактом, что она использовала ту же физическую среду (медные кабеля, витую пару или оптоволокно), что и стандартная технология Ethernet, позволяя тем самым в некоторых сетях увеличивать скорость передачи с 10 до 100 Мбит/с без изменения физической инфраструктуры.

Поскольку Fast Ethernet представляет собой протокол CSMA/CD, то логическая топология сети Fast Ethernet точно такая же, как у сети Ethernet. И точно так же, как в технологии Ethernet, протокол Fast Ethernet может использоваться в полудуплексном и дуплексном режимах. Большинство устройств, поддерживающих технологию Fast Ethernet, способны автоматически определять, является ли сегмент, к которому они подключаются, сегментом сети Ethernet (10 Мбит/с) или сети Fast Ethernet (100 Мбит/с). Кроме того, они также автоматически определяют соответствующий режим передачи данных: полудуплекс или полный дуплекс.

Интерфейсами Fast Ethernet оснащаются такие устройства Cisco, как маршрутизаторы, коммутаторы и мосты. Интерфейсы Fast Ethernet на коммутаторах часто используются в качестве порта подключения интерфейсов Ethernet к магистральному каналу. На рис. 3.3 показан пример типовой топологии, в которой коммутатор объединяет десять сегментов Ethernet в один сегмент Fast Ethernet. Затем для обеспечения доступа к глобальной сети этот сегмент Fast Ethernet подключается к маршрутизатору.

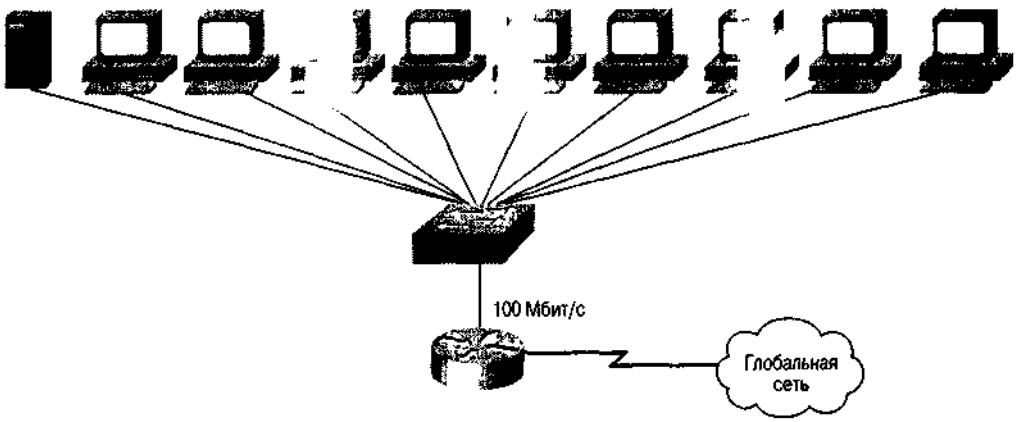


Рис 3 3. Коммутатор Ethernet с выходом на магистральный канал к маршрутизатору через интерфейс Fast Ethernet

Подкоманды конфигурирования интерфейсов Fast Ethernet и Ethernet

На некоторых маршрутизаторах Cisco серий 4000 и 7000 каждый интерфейс Fast Ethernet и Ethernet позволяет устанавливать тип среды передачи данных, подключаемой к маршрутизатору. Для того чтобы сообщить маршрутизатору тип активного соединения на интерфейсе, необходимо воспользоваться подкомандой конфигурирования интерфейса media-type. В примере ниже устанавливается тип среды передачи данных для маршрутизатора Seoul-

```
1:
Seoul-#lconfigure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Seoul-1(config)#interface ethernet 0
Seoul-1(config-if)#media-type lObaseT
Seoul-1(config-if)#^Z
```

Для интерфейсов Ethernet и IEEE 802.3 разрешенными типами сред передачи данных являются интерфейсы подключения сетевых устройств (attachment unit interfaces — AUI) и разъемы RJ-45 (называемые в ОС IOS 1OBaseT для обозначения разводки с использованием кабелей типа "витая пара"). Физически интерфейсы AUI представляют собой 15-контактные разъемы. Для интерфейсов Fast Ethernet разрешенными типами сред являются средонезависимые интерфейсы (media independent interfaces — МП) и разъемы RJ-45.

На интерфейсах Fast Ethernet можно вручную устанавливать дуплексный режим работы, для чего необходимо воспользоваться подкомандой конфигурирования интерфейса full-duplex. Если удалить эту команду из конфигурации с помощью команды no full-duplex, то интерфейс по умолчанию перейдет в режим полудуплексной связи. В приведенном ниже примере выполняется установка порта Fast Ethernet маршрутизатора Seoul-1 в режим полного дуплекса:

```
Seoul-l#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Seoul-1(config)#interface ethernet0
Seoul-1 (config-if) #full-duplex
Seoul-1(config-if)#^Z
```

Технология Gigabit Ethernet

Подобно технологии Fast Ethernet, технология Gigabit Ethernet (или IEEE 802.3z) построена на стандарте IEEE 802.3 Ethernet. Основное различие, как следует из названия, состоит в том, что скорость передачи данных между устройствами, работающими по этой технологии, составляет 1 Гбит/с. И точно таким же образом, как технология Fast Ethernet обеспечивает десятикратное увеличение скорости передачи данных по сравнению с технологией Ethernet или IEEE 802.3, так и технология Gigabit Ethernet обеспечивает скорость передачи данных в десять раз выше, чем технология Fast Ethernet. Но, в отличие от технологии Fast Ethernet, переход на технологию Gigabit Ethernet требует изменения физического интерфейса устройства.

Начиная с протоколов канального уровня и далее на более высоких уровнях модели OSI, технология Gigabit Ethernet идентична технологии Ethernet. На физическом же уровне технология Gigabit Ethernet использует тип интерфейса, который свойствен другой высокоскоростной технологии локальных сетей, называемой оптоволоконным каналом (Fiber Channel). Технология Gigabit Ethernet соединяет в себе использование физического уровня технологии Fiber Channel и формата кадра канального уровня технологий IEEE 802.3, Ethernet и Fast Ethernet. В ней используется алгоритм CSMA/CD, и она может обеспечивать работу как в режиме полудуплекса, так и в режиме полного дуплекса. Стандартом для режима полного дуплекса технологии Gigabil Ethernet является стандарт IEEE 802.3x.

Маршрутизаторы серии 7500 и коммутаторы Catalyst серии 5500 поддерживают интерфейс Gigabit Ethernet. На данный момент в маршрутизаторах серии 7500 поддерживается один интерфейс Gigabit Ethernet на слот. Как показано в примере ниже, если интерфейс Gigabit Ethernet установлен во второй слот маршрутизатора серии 7500, то ему присваивается имя Gigabit Ethernet 2/0/0 (номер слота/номер адаптера порта/номер порта):

```
Router>show interface gigabitethernet 2/0/0
GigabitEthernet2/0/0 is up, line protocol is up
Hardware is cyBus GigabitEthernet, address is 0000.Oca4.db61 (bia 0000.0ca4.db61)
Internet address is 10.0.0.2/8
   MTU 1500 bytes, BW 1000000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
   Encapsulation ARPA, loopback not set, keepalive set (10 sec)
```

```
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:00, output 00:00:01, 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 2300 bits/sec, 2 packets/sec
5 minute output rate 3000 bits/sec, 3 packets/sec
    116547 packets input, 13397137 bytes, 0 no buffer
    Received 3402 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    0 input packets with dribble condition detected
    273769 packets output, 84816409 bytes, 0 underruns
    65959 output errors, 1 collisions, 1 interface resets
    0 babbles, 0 late collision, 29 deferred
    65959 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
```

Технология Token Ring

Token Ring —- это технология локальных сетей, разработанная фирмой IBM (International Business Machines) и стандартизированная в виде протокола IEEE 802.5. В соответствии с названием протокол Token Ring работает в топологии, представляющей собой логическое кольцо, а не шину, как это имеет место для технологии Ethernet. В технологии Token Ring для доступа к среде передачи данных используется метод захвата маркера. Эта технология реализована для двух скоростей передачи данных: 4 Мбит/с и 16 Мбит/с.

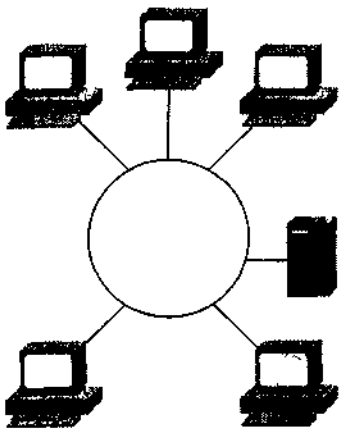


Рис. 3.4. Логическая топология сети Token Ring

Заложенный в этот протокол алгоритм относительно прост для понимания. Устройство, находящееся в сети Token Ring, должно захватывать специальный пакет называемый *маркером*. Маркер передается по логическому кольцу в направлении противоположном направлению вращения часовой стрелки. Если у устройства есть данные, которые необходимо отправить, и оно видит проходящий по кольцу маркер, то оно может осуществить захват этого маркера. Захватив маркер, устройстве передает кадр по кольцу. В процессе прохождения кадра по кольцу система-получатель копирует данные из этого кадра. Когда кадр, посланный устройством возвращается обратно, отправитель удаляет его из сети и высвобождает маркер, который вновь начинает передаваться по кольцу. В 16-мегабитных сетях Token Ring устройство-отправитель высвобождает маркер раньше, чем получает назад свой отосланный кадр, используя функцию, именуемую *ранним высвобождением маркера* (early token release). В отличие от протокола CSMD/CD, протокол захвата маркера делает невозможными конфликты при передаче, так как передать кадр в сеть Token Ring может только то устройство, которое захватило маркер. Кроме того, здесь возможен расчет максимального времени ожидания, которое будет необходимо устройству, прежде чем оно сможет отправить кадр. Это позволяет сделать технологию Token Ring детерминированной. Для некоторых сетевых приложений, например, обработки транзакций в реальном времени, такой детерминизм является очень важным требованием для протокола локальной сети. Логическая топология сети Token Ring показана на рис. 3.4.

Примечание

В сетевой промышленности постоянно идут споры относительно достоинств и недостатков протокола CSMA/CD по сравнению с методом захвата маркера. Авторы этой книги не хотят вступать в "религиозные" перепалки и поэтому не приводят суждения о преимуществах какого-либо из протоколов. На настоящий момент, безотносительно технических достоинств одного или другого протокола, на рынке протоколов локальных сетей со всей очевидностью доминирует протокол CSMA/CD.

Приведенный ниже пример является результатом выполнения команды режима *EXEC* show interfaces для порта 0 интерфейса Token Ring, стоящего в слоте 1 маршршрутизатора в Сан-Хосе:

```
San-Jose#show interfaces tokenring 1/0
Tokenring 1/0  is up,   line protocol  is up
lardware  is  16/4  Token Ring,   address  is  5500.2000.dc27   (bia5500.2000.dc27)
   MTU 8136 bytes, BW 16000 Kbit, DLY 630 usec, rely 255/255, load 1/255
   Encapsulation SNAP, loopback not set, keepalive set (10 sec)
   ARP type: SNAP, ARP Timeout 4:00:00
   Ring speed: 16 Mbps
   Single ring node, Source Route Bridge capable
   Group Address: 0x00000000, Functional Address: 0x60840000
   Last input 0:00:01, output 0:00:01, output hang never
   Output queue 0/40, 0 drops; input queue 0/75, 0 drops
   Five minute input rate 0 bits/sec, 0 packets/sec
   Five minute output rate 0 bits/sec, 0 packets/sec
       16339 packets input, 1496515 bytes, 0 no buffer
       Received 9895 broadcasts, 0 runts, 0 giants
       0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
       32648 packets output, 9738303 bytes, 0 underruns
       0 output errors, 0 collisions, 2 interface resets, 0 restarts
       5 transitions
```

Как видно из этого примера, интерфейс операционно находится во включенном состоянии. Во второй строке результата приведен зашитый в ППЗУ адрес интерфейса Token Ring, а в шестой строке указана скорость передачи данных — 16 Мбит/с.

Подкоманды конфигурирования для интерфейса Token Ring

Для задания скорости передачи данных интерфейса Token Ring (4 Мбит/с или 16 Мбит/с) используется подкоманда конфигурирования интерфейса ОС IOS ring-speed. В сети Token Ring все устройства должны работать с одной и той же скоростью передачи данных; протоколом запрещается использование конфигураций с различными значениями этой скорости. Использование устройств с разной скоростью передачи данных может привести к невозможности работы кольца.

Если принимается решение об использовании в кольце со скоростью передачи 16 Мбит/с функции раннего высвобождения маркера, то во всех устройствах в сети Token Ring эта функция должна быть активирована. Если хотя бы одно устройство (компании Cisco или другого производителя) в сети Token Ring не будет поддерживать данную функцию, то ее не сможет использовать все кольцо. Для активации функции раннего высвобождения маркера в интерфейсе Token Ring используется подкоманда конфигурирования интерфейса ОС IOS early-token-release.

В ниже представленном примере выполняется установка скорости интерфейса Token Ring в 16 Мбит/с и активация функции раннего высвобождения маркера:

```
San-Jose#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
San-Jose(config)#interface tokenring 1/0
San-Jose(config-if)#ring-speed 16
San-Jose(config-if)#early-token-release
San-Jose(config-if) #^Z
```

Технология FDDI

Распределенный интерфейс передачи данных по оптоволоконному каналу (Fiber Distributed Data Interface — FDDI) — это еще одна технология локальных сетей с захватом маркера. Технология FDDI была стандартизирована комитетом по стандартам ANSI X3T9.5 в середине 1980-х годов. Она во многом похожа на технологию Token Ring, но, вместо архитектуры с одним кольцом, предусматривает использование кольца из двух оптоволоконных кабелей, данные по которым передаются в противоположных направлениях. Во время нормального

функционирования технология FDDI использует только одной кольцо, называемое *основным*. Если же основное кольцо отказывает, задействуется второе, так называемое *резервное* кольцо. Когда на основном кольце происходит единичный разрыв, ближайшее к точке разрыва устройство переходит в режим возврата данных и использует резервное кольцо для формирования петли, гарантируя тем самым целостность FDDI-кольца. Этот процесс показан на рис. 3.5.

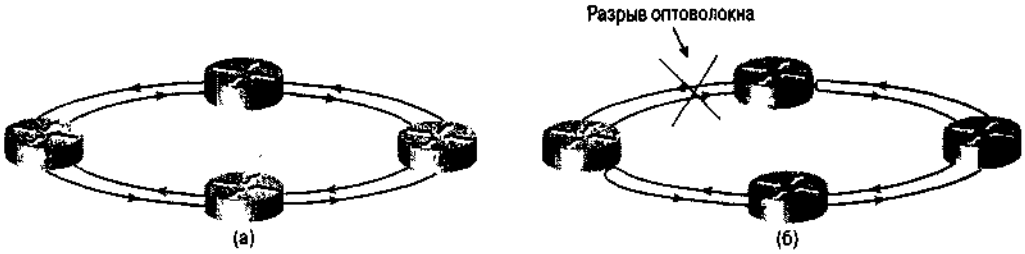


Рис. 3.5 Топология, используемая в технологии FDDI при нормальной работе (а) и в режиме возврата (б)

Аналогично Fast Ethernet интерфейс FDDI обеспечивает передачу данных на скорости 100 Мбит/с. Ввиду такой высокой скорости передачи данных, а также благодаря избыточности интерфейс FDDI часто применяется для реализации высокоскоростного магистрального канала связи от коммутатора к маршрутизатору или в качестве технологии для создания магистрального канала для комплекса зданий (кампусный магистральный канал). Коммутаторы, мосты и маршрутизаторы фирмы Cisco поддерживают технологию FDDI в рамках протоколов сетевого уровня, обеспечивающих мостовые соединения, коммутацию и маршрутизацию данных в режиме прозрачной передачи данных и в режиме трансляции. В сети фирмы ZIP корпоративный офис в Сан-Франциско использует технологию FDDI для связи между маршрутизаторами, установленными на разных этажах здания. Ниже приведен результат выполнения команды show interfaces для интерфейса FDDI маршрутизатора SF-Core-1:

```
SF-Core-l>show interfaces fddi o/o
FddiO/0 is up, line protocol is up
  Hardware is cBus Fddi, address is 0000.Oc06.8de8 (bia 0000.Oc06.8de8)
  MTU 4470 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation SNAP, loopback not set, keepalive not set
  ARP type: SNAP, ARP Timeout 4:00:00
  Phy-A state is active, neighbor is B, cmt signal bits 008/20C, status
  ILS
  Phy-B state is connect, neighbor is unk, cmt signal bits 20C/000,
  status QLS
  ECM is insert, CFM is c_wrap_a, RMT is ring_op
  token rotation 5000 usec, ring operational IdOl
  Upstream neighbor 0000.OcOб.8b7d, downstream neighbor OOOO.OcOб.8b7d
  Last input 0:00:08, output 0:00:08, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 5000 bits/sec, 1 packets/sec
  Five minute output rate 76000 bits/sec, 51 packets/sec
     852914 packets input, 205752094 bytes, 0 no buffer
     Received 126752 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     8213126 packets output, 616453062 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets, 0 restarts
     5 transitions, 0 traces
```

Как видно из этого примера, интерфейс операционно включен, во второй строке результата указан изначально зашитый канальный адрес, и полоса пропускания (BW) 100 Мбит/с указана в третьей строке. Физические характеристики каждого оптоволоконного кольца (Phy-A — основное кольцо; Phy-B — резервное) указаны в шестой и седьмой строках результата, соответственно.

Технологии глобальных сетей и технологии взаимодействия по коммутируемым каналам связи

Устройства фирмы Cisco поддерживают большое количество протоколов глобальных сетей и удаленного доступа к сети по коммутируемым каналам связи. В этой главе рассматриваются наиболее популярные из них:

- High-Level Data Link Control (HDLC);
- Point-to-Point Protocol (PPP);
- X.25;
- Frame Relay;
- Asynchronous Transfer Mode (ATM);
- Digital Subscriber Line (DSL);
- Integrated Services Digital Network (ISDN).

Подобно протоколам локальных сетей, которые были рассмотрены в данной главе, протоколы глобальных сетей работают на канальном уровне модели OS1. Эти протоколы осуществляют передачу данных из одного места в другое через интерфейсы синхронной или асинхронной последовательной передачи данных.

Примечание

При последовательной синхронной передаче данных цифровые сигналы передаются от одного устройства другому в точно синхронизированные моменты времени. При последовательной асинхронной передаче данных такая синхронизация не производится, а для управления процессом передачи данных используется контрольная информация (называемая стартовыми и стоповыми битами), которая обозначает начало и конец данных.

HDLC, первый из рассматриваемых синхронных протоколов, работает только по принципу "из точки в точку", соединяя устройства друг с другом с минимально необходимой инкапсуляцией и адресной информацией. Протокол PPP, изначально разработанный для каналов последовательной передачи данных по двухточечному принципу, впоследствии эволюционировал и стал применяться как для синхронной, так и для асинхронной передачи данных. Протоколы X.25, Frame Relay и ATM не работают в средах последовательной передачи данных, в которых используется только один двухточечный принцип. Вместо этого эти протоколы предусматривают использование для передачи данных виртуальных каналов. DSL — это технология, отличительной особенностью которой является предоставление алгоритма кодирования для высокоскоростной передачи данных по традиционным медным кабелям, но на ограниченные расстояния. ISDN — это технология глобальных сетей, которая использует телефонную сеть для передачи оцифрованных данных. Протокол ISDN может работать как в двухточечном режиме, так и в многоточечном (из одной точки во многие).

Виртуальный канал — это механизм связи, при котором, перед тем, как начать передачу данных, системы устанавливают путь прохождения информации. Следовательно, выполняется так называемый процесс размещения звонка. Все пакеты данных, относящиеся к этому звонку, передаются по сети по одному и тому же маршруту, чем гарантируется, что данные придут получателю именно в том порядке, в каком они были посланы отправителем. По окончанию процесса передачи данных звонок заканчивается, и канал уничтожается. Существуют *коммутируемые виртуальные каналы* — каналы, созданием и уничтожением которых может управлять сеть, и *постоянные виртуальные каналы* — каналы, создаваемые сетью и существующие постоянно.

Как можно понять из вышеприведенной информации, один интерфейс маршрутизатора Cisco может поддерживать несколько виртуальных каналов (постоянных или коммутируемых). В этом случае каждый канал рассматривается как отдельный интерфейс, называемый *подынтерфейсом*. Подынтерфейсы могут быть реализованы для любой технологии глобальной сети, использующей виртуальные каналы. Преимущества и некоторые подробности использования подынтерфейсов

будут рассмотрены в этой главе на примерах конфигурирования протокола Frame Relay.

Аналогом виртуальных каналов является знакомая всем телефонная сеть. Каждый звонок другому абоненту можно считать виртуальным каналом. Почти все телефонные звонки, которые мы делаем в жизни, представляют собой аналоги коммутируемых виртуальных каналов. Но если сделать звонок и оставить его активным навсегда, то это был бы постоянный виртуальный канал.

Протоколы глобальных сетей, поддерживаемые в устройствах Cisco, передают данные двумя различными методами: по методу коммутации пакетов и по методу ретрансляции ячеек. *Коммутация пакетов* — это метод передачи, при котором данные передаются блоками переменной длины, или пакетами. При использовании метода коммутации пакетов канальный уровень берет пакеты с сетевого уровня и инкапсулирует их с добавлением адресной информации в формате конкретного протокола канального уровня. В процессе перемещения таких канальных пакетов по сети каждый промежуточный узел коммутации пакетов на пути между отправителем и получателем считывает из пакета канальный адрес и соответствующим образом переадресовывает его. Пакет передается по установленному ранее виртуальному каналу до тех пор, пока не достигнет канального адреса получателя. Метод коммутации пакетов используется в протоколах Frame Relay и X.25.

Протоколы ATM и Switched Multimegabit Data Service (SMDS) (протокол SMDS в этой книге не рассматривается) преобразовывают пакеты данных в ячейки фиксированной длины и ретранслируют их по сети. *Ретрансляция ячеек* — это метод передачи, при котором данные посылаются небольшими блоками фиксированного размера, или ячейками, которые могут быстро и эффективно обрабатываться аппаратурой. Метод ретрансляции ячеек подобен методу коммутации пакетов и отличается только тем, что данные системы-отправителя сначала преобразовываются в ячейки фиксированной длины, а не группируются в пакеты. В табл. 3.1 сведены те методы передачи данных, которые используются в протоколах глобальных сетей, рассматриваемых в данной главе.

Необходимо помнить, что в адресации при передаче данных с использованием метода коммутации пакетов или ретрансляции ячеек участвуют два уровня модели OSI. Адреса для коммутации пакетов и ретрансляции ячеек находятся на канальном уровне модели OSI. Не следует их путать с адресами сетевого уровня модели OSI, используемыми протоколами IP, IPX и Apple Talk. Для маршрутизаторов Cisco является обычной маршрутизация пакетов сетевого уровня, например IP-пакетов, через сеть с коммутацией пакетов, например через сеть Frame Relay.

Таблица 3.1. Характеристики протоколов передачи данных в глобальных сетях

Протокол	Двухточечная передача	Коммутация пакетов	Ретрансляция ячеек	Асинхронный	Синхронный
HDLC	да	нет	нет	нет	да
PPP	да	нет	нет	да	Да
X25	да	да	нет	нет	да
Frame Relay	да	да	нет	нет	да
ATM	да	нет	да	нет	да
DSL	да	нет	нет	нет	да
ISDN	да	нет	нет	да	да

При маршрутизации пакетов сетевого уровня через сеть с коммутацией пакетов маршрутизатор использует IP-адреса на сетевом уровне для определения маршрута пакета к следующему маршрутизатору на пути, ведущем к пункту конечного назначения.

Затем маршрутизатор инкапсулирует весь IP-пакет в кадр протокола Frame Relay, добавляя адресацию, присущую технологии Frame Relay. После этого пакет, проходя по установленному ранее виртуальному каналу, переключается коммутаторами сети Frame Relay. Каждый коммутатор в сети Frame Relay использует для передачи пакета по каналу от отправителя к получателю только адресную информацию протокола Frame Relay.

Маршрутизаторы же считают себя непосредственно подключенными к сети Frame Relay; они как бы не "видят" коммутаторов в качестве промежуточных узлов трафика сетевого уровня.

Такая же аналогия может быть применена и к протоколам локальных сетей. Если заменить коммутаторы Frame Relay коммутаторами Ethernet, то пример по-прежнему будет корректен, исключая только тот факт, что технология Ethernet не предусматривает использование виртуальных каналов.

Протокол HDLC

Протокол HDLC — это позиционный протокол синхронной передачи данных, разработанный Международной организацией по стандартизации (ISO). Этот протокол используется для соединения одного маршрутизатора Cisco с другим. Маршрутизаторы Cisco по умолчанию используют HDLC-инкапсуляцию на всех интерфейсах синхронной последовательной передачи данных.

Компания Cisco имеет собственную версию протокола HDLC, которая не совместима с протоколами HDLC других производителей. Создание собственных вариантов протоколов не является чем-то необычным. Все реализации этого протокола различными производителями специфичны, потому что сам протокол HDLC является развитием протокола SDLC (Synchronous Data Link Control), который изначально был разработай компанией IBM. В приведенной ниже информации, выводимой маршрутизатором сети компании ZIP в Сан-Хосе, содержатся данные об интерфейсе serialO/0, использующем HDLC-инкапсуляцию:

```
San-Jose>show interface serial ○/○
SerialO/0 is up, line protocol is up
   Hardware is QUICC Serial
   MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
   Encapsulation HDLC, loopback not set, keepalive set (10 sec)
   Last input 00:00:00, output 00:00:03, output hang never
   Last clearing of "show interface" counters never
   Input queue: 0/75/0 (size/max/drops); Total output drops: 0
   Queueing strategy: weighted fair
   Output queue: 0/64/0 (size/threshold/drops
      Conversations 0/6 (active/max active)
      Reserved Conversations 0/0 (allocated/max allocated
   5 minute input rate 28000 bits/sec, 2 packets/sec
   5 minute output rate 1000 bits/sec, 2 packets/sec
      4396629 packets input, 1382566679 bytes, 2 no buffer
      Received 518019 broadcasts, 0 runts, 0 giants, 0 throttles
      1824 input errors, 661 CRC, 542 frame, 0 overrun, 0 ignored, 621
      abort
      4674425 packets output, 430814377 bytes, 0 underruns
      0 output errors, 0 collisions, 10 interface resets
      0 output buffer failures, 0 output buffers swapped out
      2 carrier transitions
      DCD=up DSR=up DTR=up RTS=up CTS=up
```

В четвертой строке выводимых данных указывается тип инкапсуляции для последовательного интерфейса — HDLC (метод инкапсуляции, используемый по умолчанию всеми последовательными интерфейсами компании Cisco). Стоит отметить, что для получения данных об интерфейсе использовалась однозначно интерпретируемая форма команды show interfaces serialO/0.

Протокол Point-to Point

Протокол PPP (Point-to-Point) — это еще один протокол глобальных сетей, поддерживаемый устройствами Cisco. Он был разработан как открытый протокол, работающий с несколькими протоколами сетевого уровня, включая протоколы IP, IPX и AppleTalk. Протокол PPP можно считать открытой версией протокола HDLC, хотя их базовые протоколы существенно разнятся. Поскольку в этом протоколе для обозначения начала или конца кадра используется специальный флаг, то он может работать как в режиме асинхронной, так и в режиме синхронной

инкапсуляции. При асинхронной инкапсуляции этот флаг применяется в качестве стартового и стопового бита кадра. Также он используется для организации бит-ориентированной синхронной инкапсуляции.

Протокол PPP основывается на протоколе контроля соединений Link Control Protocol (LCP), который отвечает за установку, конфигурирование и проверку соединений, используемых протоколом PPP. Протокол контроля сети (Network Control Protocol — NCP) представляет собой группу протоколов (один для каждого типа протокола сетевого уровня модели OSI, поддерживаемого протоколом PPP), отвечающих за установку и конфигурирование протоколов сетевого уровня для работы поверх протокола PPP. Для протоколов IP, IPX и AppleTalk существуют NCP-протоколы IPCP, IPXCP и ATALKCP, соответственно.

Подкоманды конфигурирования интерфейса PPP

Для того чтобы на интерфейсе последовательной передачи данных активировать синхронный вариант протокола PPP, используется подкоманда конфигурирования интерфейса encapsulation ppp. Ниже показано конфигурирование интерфейса seriall/1 маршрутизатора в Сан-Хосе на работу с синхронным протоколом PPP:

```
San-Jose#configure
Enter configuration commands, one per line. End with CTRL+Z.
Configuring from terminal, memory, or network [terminal]?
San-Jose(config)#nterface  serial 1/1
San-Jose(config-if)#encapsulation ppp
San-Jose(config-if)# ^Z
```

Стоит отметить, что в этом примере использовались однозначно интерпретируемые формы основной команды interface serial 1/1 и команды encapsulation ppp.

Информация, которую выводит маршрутизатор сети компании ZIP в Сан-Хосе, как раз и содержит данные об интерфейсе seriall/1, использующем инкапсуляцию по протоколу PPP:

```
Seriall/1 is up, line protocol is up
  Hardware is HD64570
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive not set
  LCP Open
  Open: IPGP
  Last input 0:00:01, output 0:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/64/0 (size/threshold/drops)
    Conversations 0/4 (active/max active)
    Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  1433 packets input, 117056 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
  714 packets output, 150299 bytes, 0 underruns
  0 output errors, 0 collisions, 11 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
```

В пятой строке этого примера показано, что протокол LCP включен. Шестая строка говорит о том, что также включен протокол IPCP. Исходя из того факта, что активирован протокол IPCP, можно сделать вывод, что протокол PPP на данном интерфейсе сконфигурирован на инкапсулирование IP-пакетов.

Протокол X.25

Протокол X.25 был разработан в 1970-х годах. Этот протокол является представителем протоколов, использующих метод коммутации пакетов, и поддерживает работу Как с

коммутируемыми, так и постоянными виртуальными каналами. Административное управление развитием протокола X.25 осуществляет Международный телекоммуникационный союз (1TU) — агентство, работающее под эгидой Организации Объединенных Наций. Ввиду международного признания протокола X.25 он, возможно, является одним из самых распространенных протоколов глобальных сетей.

Так же, как и все технологии с коммутацией пакетов, протокол X.25 рассматривает сеть передачи данных, по сути, как обыкновенную телефонную сеть и передает данные, используя виртуальные каналы. Обмен данными между двумя устройствами начинается с того, что одно устройство дозванивается до другого для создания коммутируемого или постоянного виртуального канала. Затем происходит сам процесс передачи данных, после чего выполняется процедура завершения звонка. Протокол X.25 предусматривает двухточечный обмен данными между оконечным пользовательским устройством (DTE) и оконечным устройством канала передачи данных (OCE). Устройства DTE (например, маршрутизаторы Cisco) подключаются к устройствам OCE (например, модемам), которые, в свою очередь, подключаются к одному или нескольким коммутаторам сети X.25 и, в конечном итоге, к другим устройствам DTE.

Примечание

Оконечное устройство канала передачи данных (DCE) — это устройство, которое является концевым устройством интерфейса пользователь-сеть со стороны сети. Устройства DCE обеспечивают физическое соединение с сетью, пропускают через себя трафик и вырабатывают сигналы синхронизации, используемые для синхронизации передачи данных между устройствами DCE и DTE.

Оконечное пользовательское устройство (DTE) — это устройство, находящееся с другой, пользовательской стороны интерфейса пользователь-сеть. Эти устройства могут выступать в роли отправителя данных, получателя или совмещать эти функции. Устройство DTE соединяется с сетью передачи данных с помощью устройств DCE (например, модемов) и обычно использует сигналы синхронизации, вырабатываемые устройствами DCE.

Так называемый "звонок" в сети X.25 начинается с того, что устройство DTE, выступающее отправителем данных, совместно с тем устройством OCE, с которым соединено, инициирует сеанс связи. Коммутаторы в сети X.25 определяют маршрут передачи звонка от отправителя к получателю. Затем все данные коммутируются от устройства DTE-отправителя на устройство DTE-получателя. Этот механизм изображен на рис. 3.6.

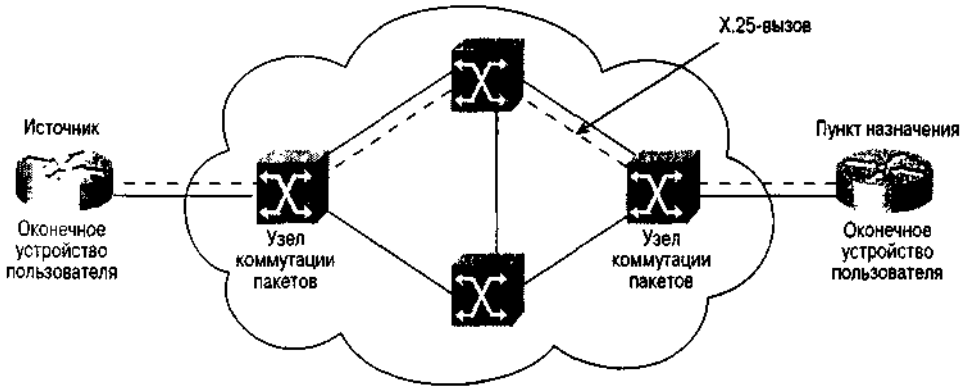


Рис. 3.6. Сеть X.25

В протоколе X.25 используется схема адресации, называемая X.121. Форматы адресов отправителя и получателя канального уровня для протокола X.25 определены рекомендательным документом ITU-T Recommendation X.121. Коммутаторы сети X.25 маршрутизируют звонок вдоль пути виртуального канала, основываясь на X.121-адресах устройства-отправителя и устройства-получателя.

Адреса, определяемые документом X.121, не имеют четко фиксированной длины и могут содержать до 14 десятеричных цифр. Первые четыре цифры адреса называются идентификационным кодом сети передачи данных (data network identification code — DNIC).

Оставшиеся цифры адреса могут быть использованы администратором сети по своему усмотрению.

Подкоманды конфигурирования интерфейса X.25

Для того чтобы использовать протокол X.25 на последовательном интерфейсе в устройствах Cisco, необходимо так его сконфигурировать, чтобы он мог выполнять инкапсулирование пакетов протокола X.25. Это делается с помощью команды encapsulation x25. ,

В отличие от канальных адресов, используемых в локальной сети, адреса канального уровня X.121 протокола X.25 не зашиваются в ППЗУ. Это означает, что сетевому администратору необходимо указать маршрутизатору Cisco X.I21-адрес его последовательного интерфейса, работающего с протоколом X.25. Это осуществляется с помощью подкоманды конфигурирования интерфейса x25 address. Некоторые производители коммутаторов X.25 требуют от пользователей, чтобы те устанавливали максимальный размер входных и выходных пакетов (по умолчанию размер пакета составляет 128 байт). Для обеспечения оптимальной работы в конкретной сети X.25 может понадобиться конфигурирование последовательного интерфейса маршрутизатора Cisco на соответствующие размеры входных (ips) и выходных (ops) пакетов. Для этого используются команды x25 ips и x25 ops.

Сети X.25 имеют принимаемые по умолчанию размеры входных и выходных окон для пакетов, которые используются механизмом управления потоком данных. Для нормального функционирования сети X.25 может понадобиться установка используемых по умолчанию размеров входного (win) и выходного окон (wout). (Исходный размер входного и выходного окон составляет 2 пакета.) Для этой цели используются подкоманды конфигурирования интерфейса x25 win и x25 wout.

При использовании коммутаторов X.25 необходимо свериться с рекомендациями изготовителя относительно размеров пакетов и окон. Согласование этих параметров между устройствами DTE и OCE зачастую является необходимым условием нормального функционирования протокола X.25 на канальном уровне.

В приведенном ниже примере показывается конфигурирование маршрутизатора в Сан-Хосе на использование инкапсуляции протокола X.25 и X.121-адреса интерфейса канального уровня 537000000001. Также здесь устанавливается размер входных и выходных пакетов — 256 байт и размер входных и выходных окон — 7 пакетов:

```
San-Jose#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
San-Jose(config)#interface serial 1
San-Jose(config-if)#encapsulation x25
San-Jose(config-if)#x25 address 537000000001
San-Jose(config-if)#x25 ips 256
San-Jose(config-if)#x25 ops 256
San-Jose(config-if)#x25 win 7
San-Jose(config-if)#x25 wout 7
San-Jose(config-if)#^Z
```

Следует отметить большое количество подкоманд конфигурирования, используемых в этом примере.

Следующий пример — результат выполнения команды show interfaces в отношении использующего инкапсуляцию протокола X.25 интерфейса маршрутизатора :

```
Serial 0 is up, line protocol is up
  Hardware is MCI Serial
  MTU 1500 bytes, BW 512 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation X25-DTE, loopback not set, keepalive set
  LAPB state is CONNECT, Tl 3000, N1 12000, N2 20, K7, TH 3000
  Window is closed
  IFRAMEs 12/28 RNRs 0/1 REJs 13/1 SABMs 1/13 FRMRs 3/0 DISCS 0/11
```

```
Last input 0:00:00, output 0:00:00, output hang never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 1 packets/sec
5 minute output rate 1000 bits/sec, 1 packets/sec
  261 packets input, 13212 bytes, 0 no buffer
  Received 33 broadcasts, 0 runts, 0 giants
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  238 packets output, 14751 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets, 0 restarts
```

В приведенном выше примере в четвертой строке вывода указывается, что тип инкапсулирования для интерфейса: X.25 DTE. В следующих за ней трех строках показывается статистика протокола сбалансированной процедуры доступа к каналу связи Link Access Procedure, Balanced (LAPB). LAPB — это протокол канального уровня, используемый группой протоколов X.25 и основанный на протоколе HDLC. Для того чтобы посмотреть состояние виртуальных каналов X.25 на устройстве, следует воспользоваться командой режима EXEC show x25 vc.

Протокол Frame Relay

Протокол Frame Relay тоже относится к протоколам глобальных сетей, которые используют метод коммутации пакетов. Первоначально этот протокол был разработан для применения в цифровых сетях с интегрированными службами (Integrated Services Digital Network — ISDN). (Технология ISDN рассматривается в этой главе позже.) Первые предложения по стандартам протокола Frame Relay были представлены в Консультативный комитет по международной телефонной и телеграфной связи (CCITT) в 1984 году. Несмотря на то что этот протокол уже был стандартизирован, существовали проблемы со взаимодействием его версий от разных поставщиков. Именно поэтому данная технология не получала широкой поддержки в промышленности вплоть до конца 1980-х годов.

Подобно протоколу X.25, в протоколе Frame Relay используется метод коммутации пакетов и постоянные и коммутируемые виртуальные каналы. На сегодня в большинстве сетей Frame Relay используются постоянные виртуальные каналы, так как технические решения работы с коммутируемыми виртуальными каналами только-только начали находить практическое воплощение. Протокол Frame Relay использует ту же самую технологию установки звонка, передачи данных и закрытия связи, которая была описана для протокола X.25. Оконечные устройства, например маршрутизаторы, инициируют звонок в сети Frame Relay. После установки связи маршрутизатор передает данные и выполняет процедуру закрытия связи. Для постоянных виртуальных каналов звонок всегда активен, что позволяет маршрутизатору посылать данные без инициирования звонка.

Так же, как в протоколе X.25 используются адреса X.121, протокол Frame Relay использует адреса, называемые идентификаторами канала соединения (data link connection identifiers — DLCI). Каждый идентификатор DLCI может иметь в сети Frame Relay локальное или глобальное значение. На сегодняшний день наиболее распространенной практикой является использование идентификаторов DLCI только с локальным значением. Это означает, что устройства, например маршрутизаторы, на разных сторонах виртуального канала в сети Frame Relay могут иметь один и тот же DLCI-номер, поскольку протокол Frame Relay предусматривает отображение локального DLCI-номера на виртуальный канал на каждом из коммутаторов, стоящих в глобальной сети. Пример сети Frame Relay приведен на рис. 3.7.

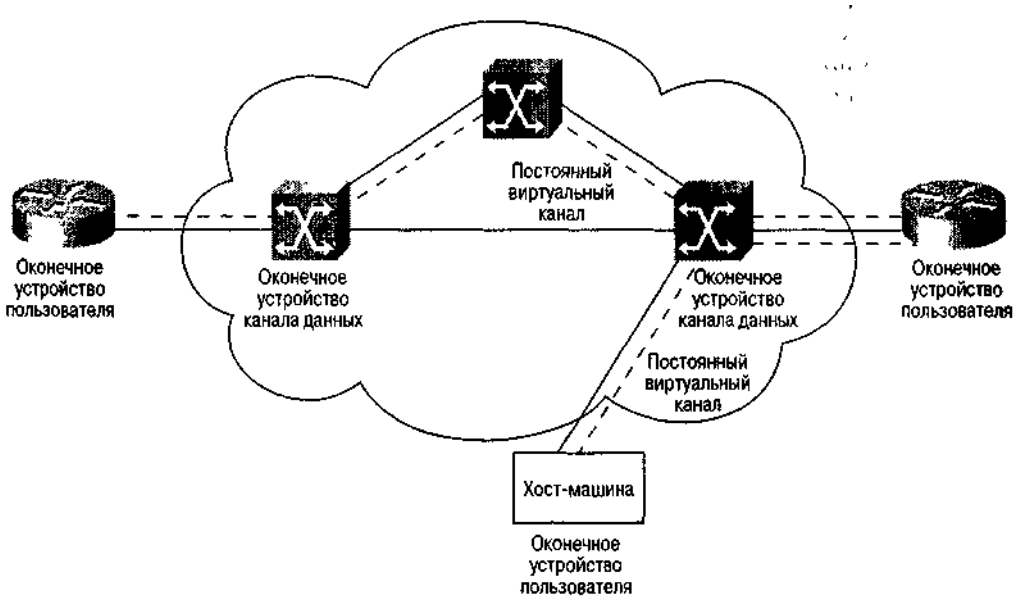


Рис. 3.7. Пример сети Frame Relay, использующей постоянные виртуальные каналы

В 1990 году компании Cisco, Digital Equipment Corporation, Northern Telecom и StrataCom образовали консорциум, целью которого было дальнейшее развитие технологии Frame Relay и обеспечение совместимости его версий от различных поставщиков. Эта группа производителей взяла за основу протокол Frame Relay, одобренный комитетом CC1TT, и добавила к нему расширения, позволяющие устройствам межсетевого взаимодействия оптимально обмениваться данными в сети Frame Relay.

Эти расширения, называемые интерфейсом локального управления (Local Management Interface — LMI), позволяют DTE-устройствам сети Frame Relay (например, маршрутизаторам) общаться с DCE-устройствами и производить обмен служебной информацией, которая используется для передачи межсетевого трафика по глобальной сети Frame Relay. Сообщения интерфейса LMI предоставляют информацию о текущих значениях DLCI, их характере (локальные они или глобальные) и о статусе виртуальных каналов.

Примечание

Консорциум LMI, созданный фирмами Cisco, DEC, NT и StrataCom теперь известен под названием *Большая четверка LMI,* или *Cisco LMI. В* дополнение к документам консорциума LMI Американский национальный институт стандартов (ANSI) разработал стандарт протокола LMI, называемый Annex-D (Приложение D), который используется в сетях Frame Relay во всем мире.

Подкоманды конфигурирования интерфейса Frame Relay

Для того чтобы сконфигурировать интерфейс последовательной передачи данных маршрутизатора Cisco на работу с протоколом Frame Relay, следует ввести подкоманду конфигурирования интерфейса encapsulation frame-relay. Затем нужно воспользоваться подкомандой frame-relay interface-dlci и присвоить этому интерфейсу значение идентификатора DLCI. Устройства Cisco по умолчанию используют на интерфейсах Frame Relay интерфейс Cisco LMI. Однако с помощью подкоманды конфигурирования интерфейса frame-relay imi-type можно явно установить тип интерфейса LMI

Если использовать в качестве примера сеть компании ZIP, то интерфейс Frame Relay маршрутизатора в Сингапуре может быть сконфигурирован следующим образом:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interface serial 0
Singapore(config-if)#encapsulation frame
Singapore(config-if)#frame-relay interface-dlci 100
```

```
Singapore(config-if)#frame-relay Imi-type ansi
Singapore(config-if)#^Z
```

Приведенный выше пример представляет собой базовую конфигурацию, предусматривающую работу последовательного интерфейса Cisco с одним виртуальным каналом. Как уже упоминалось, один последовательный интерфейс может поддерживать несколько виртуальных каналов, каждый из которых может рассматриваться в качестве отдельного интерфейса, называемого подинтерфейсом. Подинтерфейс можно считать аппаратным интерфейсом, определенным в ОС IOS.

Польза от применения концепции подинтерфейсов заключается в возможности назначать каждому подинтерфейсу и виртуальному каналу различные характеристики сетевого уровня. Например, можно назначить одному подинтерфейсу задачу маршрутизации по протоколу IP, а другому — по протоколу AppleTalk. Для задания виртуального интерфейса используется команда interface serial slot/port.number. Параметр number (номер) определяет номер подинтерфейса, связанного со значениями номера слота и порта slot/port.

Существуют два типа подинтерфейсов: *двухточечные* и *многоточечные.* Двухточечные подинтерфейсы используются тогда, когда два маршрутизатора соединяются между собой одним виртуальным каналом. Двухточечный подинтерфейс можно считать виртуальным каналом, эмулирующим выделенную линию последовательной передачи данных. Многоточечные подинтерфейсы используются, если маршрутизатор является центральным узлом виртуальных каналов с топологией "звезда". Оба примера использования подинтерфейсов показаны на рис. 3.8.

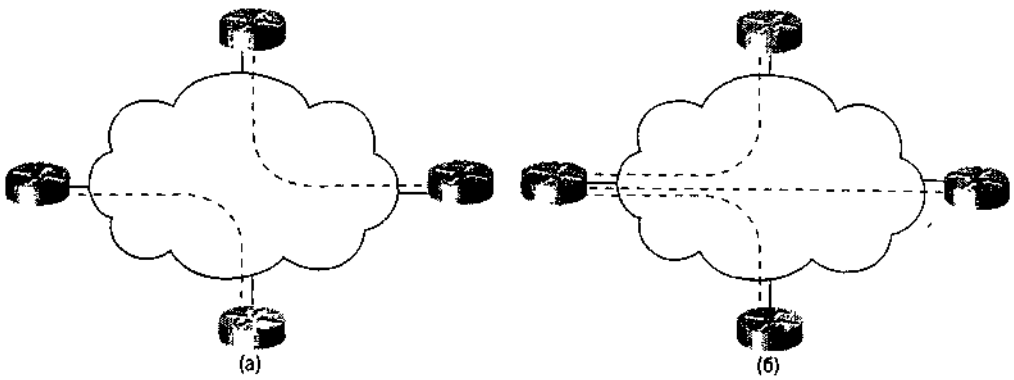


Рис. 3.8. Двухточечные (а) и многоточечные (б) сети Frame Relay

На одном физическом интерфейсе можно задать неограниченное количество подинтерфейсов (ограничение накладывает лишь объем памяти маршрутизатора). В примере ниже на маршрутизаторе в Сингапуре выполняется конфигурирование подинтерфейса serial 0.100:

```
Singapore#configure
Configuring  from terminal,   memory,   or network   [terminal]?
Enter configuration commands,   one per  line.   End with CTRL+Z.
Singapore(config)#interface serial 0
Singapore(config-if}#encapsulation frame
Singapore(config-if)#interface serial 0.100 point-to-point
Singapore(config-subif)#frame-relay interface-dlci  100
Singapore(config-subif)#frame-relay Irai-type ansi
Singapore(config-subif)#^Z
```

Совет

При создании подинтерфейсов рекомендуется твердо придерживаться выбранной схемы их нумерации. Также следует присваивать подинтерфейсу номер в соответствии с DLCI-номером для данного виртуального канала.

Статус интерфейса Frame Relay можно узнать с помощью команды show interfaces. Ниже

приведен результат выполнения команды show interfaces s 0 для маршрутизатора компании ZIP в Сингапуре:

```
Serial0 is up,   line protocol  is up
  Hardware is HD64570
  MTU  1500  bytes,   BW  256  Kbit,   DLY  20000  usec,   rely  255/255,   load  1/255
  Encapsulation  FRAME-RELAY,   loopback not  set,   keepalive  set   (10  sec)
  LMI  enq sent   459618, LMI stat  recvd  459618,   LMI  upd  recvd 0,   DTE  LMI  up
  LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 100 LMI type is CISCO frame relay DTE
  Broadcast queue 0/64, broadcasts sent/dropped 121505/0, interface
  broadcasts 121505
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/64/0 {size/threshold/drops)
     Conversations 0/9 (active/max active)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 0 bits/sec, 1 packets/sec
     34278826 packets input, 2790079482 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     17 input errors, 7 CRC, 9 frame, 0 overrun, 0 ignored, 1 abort
     29613202 packets output, 1145345093 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up DSR=up DTR=up RTS=up CTS=up
```

Как видно из данных в четвертой строке, режим инкапсуляции на интерфейсе соответствует работе с протоколом Frame Relay. Следующие три строки содержат информацию интерфейса LMI. В показанном ниже примере представлен результат исполнения команды show interfaces s 0.100, которая позволяет посмотреть статус подинтерфейса:

```
Serial0.100  is  up,   line protocol  is up
     Hardware  is  HD64570
     MTU  1500  bytes, BW 256 Kbit, DLY 20000  usec, rely  255/255, load 1/255
     Encapsulation  FRAME-RELAY
```

Объем информации результата исполнения этой команды гораздо меньше, чем объем, получаемый в результате исполнения нормальной команды show interfaces. Объясняется это тем, что подинтерфейс наследует все диагностические данные основного интерфейса, с которым он связан (в данном случае это интерфейс serial0).

Состояние виртуальных каналов Frame Relay можно проверить, воспользовавшись командой режима EXEC show frame pvc или show frame svc *maplist*. Получение данных о коммутируемых виртуальных каналах требует применения опции maplist. Это дополнение представляет собой список соединений между данным устройством и другими устройствами, используемыми при установке коммутируемых виртуальных каналов. Ниже приведен пример выводимого результата для постоянного виртуального канала с DLCI-номером 100 на маршрутизаторе компании ZIP в Сингапуре:

```
PVC Statistics for interface Serial 0 (Frame Relay DTE)

DLCI = 100, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE =
Serial0.100
  input pkts 34263984 output pkts 29648752 in bytes 3135739012
  out bytes 1083480465 dropped pkts 93 in FECN pkts 170
  in BECN pkts 11741 out FECN pkts 0 out BECN pkts 0
  in DE pkts 15741022 out DE pkts 0
  pvc create time 7w5d, last time pvc status changed IdlOh
```

Технология Asynchronous Transfer Mode

Асинхронный режим передачи (Asynchronous Transfer Mode — ATM) — это разработанный сектором по стандартизации телекоммуникаций Международного союза по электросвязи ITU-T стандарт использования метода ретрансляции ячеек. В технологии ATM длина ячеек составляет 53 байт.

Используя ретрансляцию ячеек, технология ATM позволяет поддерживать большое количество сетевых служб, включая передачу голоса, видеоизображений и данных. Сеть ATM состоит из коммутаторов ATM (DCE) и оконечных устройств ATM (DTE). Оконечные устройства передают данные коммутаторам ATM, которые разбивают данные на ячейки и передают эти ячейки по сети. Этот процесс одинаков для всех трех типов трафика, поддерживаемых в сетях ATM.

Сектор ITU-T использовал в качестве основы для технологии ATM стандарт широкополосной цифровой сети с интегрированными службами (Broadband Integrated Services Digital Network — BISDN), который первоначально был разработан для передачи по общедоступным сетям звука, видео и данных. Группа компаний сформировала рабочий форум по ATM, результатом деятельности которого стал выпуск в свет спецификаций, призванных обеспечить совместимость ATM-продуктов различных производителей, и расширений стандарта ATM для общедоступных и частных сетей. На настоящий момент форум разработал три версии интерфейса пользователь-сеть (User-Network Interface — UNI). UNI представляет собой протокол, концептуально похожий на протокол интерфейса LMI в технологии Frame Relay, который предназначен для стандартизации обмена данными между ATM-устройствами и коммутаторами. ATM-форум также выпустил документы, определяющие стандартную процедуру связи ATM-коммутаторов между собой (так называемый интерфейс частная сеть-сеть Private Network-to-Network Interface — PNNI), и метод эмуляции классической архитектуры локальных сетей в ATM-сети, получивший название LAN Emulation (LANE).

Подобно рассматривавшимся ранее технологиям с коммутацией пакетов, ATM поддерживает два типа ориентированных на соединение виртуальных каналов: коммутируемые виртуальные каналы и постоянные. ATM также имеет службы для работы без соединения, которые позволяют ей вести себя подобно технологии локальной сети. Используя виртуальные каналы, ATM поддерживает оба метода передачи данных: с установлением соединения и без такового. Виртуальный канал ATM аналогичен виртуальному каналу в технологии X.25 или Frame Relay.

В ATM-сети соединения рассматриваются как некие виртуальные пути (virtual paths), которые маркируются номерами, называемыми идентификаторами виртуальных путей (virtual path identifiers — VPI). *Виртуальный путь* — это группа виртуальных каналов, которые коммутируются в сети ATM на основе одного значения идентификатора VPI. Виртуальный путь можно рассматривать в качестве механизма группировки ряда виртуальных каналов при маршрутизации.

Виртуальный канал в технологии ATM идентифицируется комбинацией из VPI и идентификатора виртуального канала (virtual channel identifier — VCI). VPI определяет виртуальный путь, который используется виртуальным каналом в сети, а VCI идентифицирует уникальное соединение в группе, соответствующей данному VPI. Нумерация идентификаторов VPI и VCI носит только локальный характер. Этим они похожи на идентификаторы DLCI в технологии Frame Relay, которые чаще всего тоже нумеруются локально. Коммутаторы ATM составляют свою комбинацию VPI/VCI для каждого постоянного соединения на пути к следующему узлу передачи данных (в направлении системы-получателя).

В сетях ATM виртуальные пути объединяются в так называемые *пути передачи*. Путь передачи содержит определенное количество виртуальных путей, которые, в свою очередь, содержат группы виртуальных каналов, что и изображено на рис. 3.9.

Рис 3.9. Взаимосвязь между виртуальными каналами, виртуальными путями и путями передачи в АТМ-сети

В АТМ-сетях используется два различных типа адресации: адресация на основе стандарта ЕЛ64 (схема адресации, похожая на телефонные номера) и адресация с использованием адресов точек доступа к сетевой службе в открытых системах (OSI Network Service Access Point — NSAP). Схема адресации Е.164 была разработана в ITU-T, а метод адресации, основанный на NSAP, был предложен АТМ-форумом. Обычно адресация в соответствии со схемой Е.164 используется в АТМ-сетях общего пользования, предоставляемых операторами телекоммуникационных услуг, а NSAP-адресация используется в частных АТМ-сетях, например, в сетях, обеспечивающих связь АТМ-коммутаторов с устройствами межсетевого взаимодействия.

Как отмечалось ранее, АТМ-технология разрабатывалась для реализации сетевых служб передачи голоса, видеоизображений и данных. Для того чтобы спрятать некоторые сложные моменты реализации протокола АТМ от этих служб верхнего уровня, были введены три уровня адаптации АТМ (ATM adaptation layer — AAL). AAL-уровни — это уровни, 'которые расположены на канальном уровне модели OSI. Каждый из них называется AAL и отвечает за предоставление различных АТМ-сервисов протоколам сетевого уровня модели OSI. Уровень AAL1 представляет собой ориентированную на установление соединения АТМ-службу, которая обычно используется для эмуляции в АТМ-сети выделенных каналов передачи данных. Наиболее ярким примером приложений уровня AAL1 является организация соединений для передачи голоса и видеоизображений. Следующий уровень, AAL3/4, поддерживает передачу данных как с установлением соединения, так и без него. Обычно соединения по протоколу уровня AAL3/4 используются поставщиками сетевых услуг для передачи данных без установления соединения. Уровень AAL3/4 был разработан для обеспечения более легкой интеграции в сеть со службой коммутируемой мультимегабитной передачи данных (Switched Multimegabit Data Service — SMDS), являющейся другой стандартной технологией, использующей метод ретрансляции ячеек. Третий уровень AAL, AAL5, также поддерживает службы с установлением соединения и без него. Уровень AAL5 используется для передачи информации, для которой не требуется интеграция в SMDS-сети, например, данных частных локальных сетей или глобальных сетей. На сегодняшний день большинство АТМ-соединений в частных сетевых комплексах пользуются уровнем AAL5.

Другая важнейшая особенность технологии АТМ заключается в способности поддерживать сетевую службу качества (Quality of Service — QoS). Каждое АТМ-устройство взаимодействует с АТМ-сетью, обеспечивая определенное качество обслуживания для каждого виртуального пути и основываясь при этом на контрактных условиях обеспечения трафика, определенных методах его формирования и политике обеспечения трафика. Контрактные условия трафика определяют требования виртуального канала к величине пиковой полосы пропускания, средней полосы и размеру пачки пакетов. Методы формирования формы трафика позволяют удерживать трафик в рамках контракта за счет ограничения размеров передаваемых пачек пакетов данных и передачи ячеек согласованным потоком. Политика обеспечения трафика включает в себя методы обеспечения условий контракта на основе сравнения фактического трафика с тем, который определен контрактом. Процедуры проверки соответствия позволяют коммутаторам в случае переполнения канала уничтожать ячейки, если они не соответствуют контракту. Эти функции службы качества в АТМ-сетях делают данную технологию мощным средством для удовлетворения требований различных данных при их передаче по сети мультимедийной связи (звук, изображение, информация).

Подкоманды конфигурирования интерфейсов ATM

Интерфейсы ATM компании Cisco реализуются в виде специально выделяемых процессоров интерфейса (или адаптеров порта для VIP-плат). Это означает, что для ATM-интерфейса нет необходимости использовать подкоманду конфигурирования интерфейса encapsulation. Сама аппаратура поддерживает только режим инкапсулирования протокола ATM. Единственное, что необходимо сделать, так это задать виртуальные каналы, существующие на данном интерфейсе, для чего используется интерфейсная подкоманда atm pvc. Ниже показан пример конфигурирования постоянного виртуального канала PVC1 для соединений уровня AAL5 со значениями VPI 0 и VCI 100:

```
Router#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Router(config)#int atm2/0
Router(config-if)#atm pvc 1 0 100 aalSsnap
Router(config-if)#^Z
```

Для просмотра статуса ATM-интерфейса используется команда show interfaces. Ниже приведен результат исполнения команды show interface atm2/0 для показанной выше конфигурации:

```
ATM2/0 is up, line protocol is up
   Hardware is cxBus ATM
   MTU 4470 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255 Encapsulation
   ATM, loopback not set, keepalive set (10 sec) Encapsulation(s): AAL5, PVC mode
   256 TX buffers, 256 RX buffers, 1024 Maximum VCs, I Current VCs Signalling vc = 1,
   vpi = 0, vci = 100
   ATM NSAP address: BC.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.13 Last input
   0:00:05, output 0:00:05, output hang never
   Last clearing of "show interface" counters never
   Output queue 0/40, 0 drops; input queue 0/75, 0 drops
   Five minute input rate 0 bits/sec, 0 packets/sec
   Five minute output rate 0 bits/sec, 0 packets/sec
     144 packets input, 3148 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     154 packets output, 4228 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets, 0 restarts
```

В четвертой строке результата указано, что инкапсуляция ATM активирована, пятая свидетельствует об активации ААЬ5-инкапсулирования и режима постоянного виртуального канала. Номера VC, VPI и VCI указаны в седьмой строке результата. Кроме того, этот интерфейс имеет ATM NSAP-адрес, который указан в восьмой строке результата.

Технология Digital Subscriber Line

Технология цифровых абонентских линий Digital Subscriber Line (DSL) — это технология, которая стала популярной в последние годы и позволяет предоставлять конечным пользователям выделенную полосу пропускания большой ширины. DSL работает в сетях с топологией "звезда", в которых от центра к листовым узлам проложены выделенные линии связи на основе медных кабелей типа "витая пара". Скорость передачи данных между центральным и листовыми узлами может лежать в пределах от 64 Кбит/с до 8 Мбит/с. Эта скорость зависит от характеристик используемого кабеля, количества физических соединений, расстояния, которое проходит сигнал, погодных условий и конкретной используемой технологии DSL. Короткие расстояния, минимальное количество соединений и использование кабеля с большим диаметром жилы способствуют более высокой скорости передачи данных.

На данный момент на рынке существует много различных вариантов технологии DSL. В сетевой промышленности эту группу технологий принято называть xDSL. В нее входят технология асимметричной абонентской линии (Asymmetric Digital Subscriber Line — ADSL), технология симметричной абонентской линии (Symmetric DSL — SDSL) и технология сверхскоростной абонентской линии (Very High Data Rate DSL - VDSL).

Технология ADSL обеспечивает асимметричную полосу пропускания в канале между центром звезды и листовым узлом. Скорость передачи данных от центра звезды к листовому узлу выше (как минимум в три раза), чем скорость передачи данных в противоположном направлении. Технология ADSL является весьма привлекательной услугой провайдера Internet-сервиса конечным пользователям, поскольку пользователи Internet обычно принимают гораздо больше данных, чем передают. В такой конфигурации провайдер Internet-сервиса находится в центре DSL-звезды, а пользователи представляют собой листовые узлы. Для создания канала провайдер устанавливает два ADSL-модема, которые соединяются между собой медным кабелем типа "витая пара". Установленные модемы создают в таком кабеле три независимых канала: канал потока данных от центра, дуплексный канал и канал базового телефонного сервиса. Провайдер ADSL-сервиса может использовать эти каналы для предоставления конечным пользователям услуг по телефонной связи и передаче данных.

Сетевое оборудование, например, маршрутизаторы или мосты, обычно подключается к ADSL-модему с использованием технологии глобальных сетей, к примеру, Frame Relay или ATM. Каждый конечный пользователь или иногда каждый канал конечного пользователя выглядит для сетевого оборудования как отдельный виртуальный канал. На одном высокоскоростном интерфейсе глобальной сети, к которому подключается ADSL-модем, маршрутизатор может поддерживать большое количество виртуальных каналов и соответствующих пользователей. Такая топология показана на рис. 3.10.

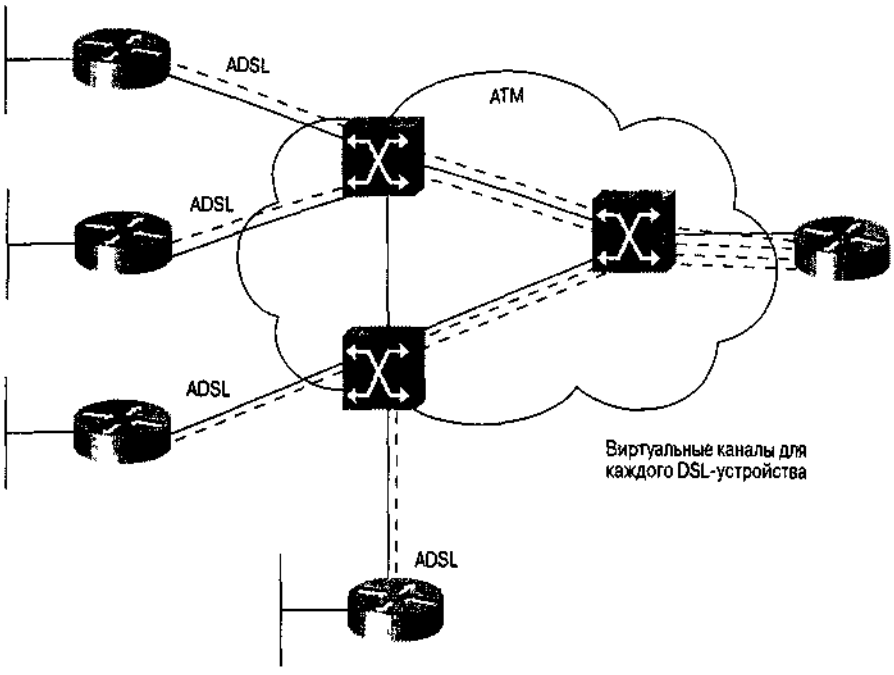


Рис. 3.10. Подключение сети ADSL к ATM-интерфейсу маршрутизатора Cisco с виртуальными каналами для каждого DSL-устройства

На текущий момент провайдеры Internet-сервиса предоставляют конечным пользователям только услуги по асимметричной передаче данных, используя технологию ADSL, хотя прогнозируется, что в скором будущем они будут предоставлять и услуги местной телефонной связи. Однако многие факторы, и административные в их числе, могут изменить это будущее.

Технология SDSL обеспечивает между центром звезды и листовыми узлами одинаковую

полосу пропускания в обоих направлениях (как любой другой канал полной дуплексной связи). На сегодняшний день эта технология используется на малых предприятиях для соединения офисов между собой и для доступа к сети Internet. Провайдеры Internet-сервиса и провайдеры других услуг подключаются к SDSL-каналам так же, как и при использовании технологии ADSL. Основное отличие заключается в том, что модемы, стоящие на каждом конце кабеля "витая пара", обеспечивают симметричный канал передачи данных.

Технология VDSL обеспечивает на коротких расстояниях широкополосные соединения по телефонным линиям на основе кабелей типа "витая пара". Так же, как и в Других технологиях DSL, фактическая скорость передачи данных зависит от длины кабеля между DSL-модемами. На текущий момент технология VDSL все еще разрабатывается, но высокие скорости передачи данных позволили бы провайдерам DSL-сервиса предоставлять больше новых услуг. Разработчики этой технологии серьезно говорят, что реально достижение скоростей передачи данных от 13 до 55 Мбит/с. Наиболее вероятно, что первая реализация технологии VDSL будет асимметричной со скоростями передачи данных к центру, лежащими в диапазоне от 1,6 до 2,3 Мбит/с.

Компания Cisco выпускает специальную серию маршрутизаторов, серию 600, оснащенную DSL-интерфейсами. Эти маршрутизаторы могут выполнять роль мостов и маршрутизаторов между сетями DSL и Ethernet или работать в качестве модемов для технологий ADSL и SDSL. На текущий момент эти маршрутизаторы работают под управлением варианта IOS, называемого Cisco Broadband Operating System (CBOS). Пока что процесс конфигурирования этой ОС отличается от процесса конфигурирования ОС IOS, однако Cisco планирует переработать пользовательский интерфейс ОС CBOS для большей совместимости с ОС IOS. С помощью комбинации продуктов компании Cisco, работающих под управлением IOS и CBOS, можно построить сеть передачи данных, использующую самые разные методы передачи данных.

Технология ISDN

Технология цифровой сети с интегрированными службами (Integrated Service Digital Network — ISDN) представляет собой технологию глобальных сетей с установлением соединения, использующую цифровую телефонию для передачи оцифрованной речи, видеоизображений, данных и другой информации по существующим телефонным кабелям. В настоящее время большое количество телефонных компаний во всем мире предлагают конечным пользователям ISDN в качестве цифровой абонентской службы для доступа к сети Internet, обеспечения обычной голосовой телефонной связи и для проведения видеоконференций. Результатом создания сети ISDN стала возможность для устройства ISDN размещать телефонные звонки в сети оператора телефонной связи, которая в этом случае будет способна транспортировать данные различных типов. В принципе ISDN-устройство можно считать цифровым модемом, который может передавать различные типы данных.

Устройства, которые подключаются к сети ISDN, называются терминалами. Существуют два типа терминалов: те, которые соответствуют стандартам ISDN и называются терминальным оборудованием типа 1 (TE1), и те, которые возникли до появления стандартов ISDN и называются терминальным оборудованием типа 2 (TE2). Терминалы TE2 подключаются к сети ISDN с помощью терминального адаптера (TA). Устройства TE1 в адаптерах не нуждаются. Следующим шагом в обеспечении обмена данными с сетью ISDN является подключение сетевого оконечного устройства (network termination device) типа 1 (NT1) или сетевого оконечного устройства типа 2 (NT2). Оконечные сетевые устройства обоих типов преобразовывают четырехпроводную линию, используемую операторами телефонной связи, в двухпроводную телефонную линию, которая обычно используется в жилых домах и на предприятиях.

В Северной Америке обычно устройством NT1 оборудовано само место расположения пользователя или оно уже есть в сетевом устройстве. Там большинство ISDN-соединений, инициируемых ISDN-платами ПК или ISDN-маршрутизаторами, используют встроенные устройства NT1. В других странах устройство NT1 предоставляет оператор телефонной

связи; оно не является частью ISDN-устройства в месте нахождения пользователя. Устройство типа NT2, добавляющее к устройству NT1 функции канального и сетевого уровня, обычно используется при подсоединении мини-АТС. Взаимоотношения между составляющими элементами ISDN-сети показаны на рис. 3.11.

Сеть ISDN обеспечивает для устройств работу служб двух типов: интерфейса передачи данных с номинальной скоростью (Basic Rate Interface — BRI) и интерфейса передачи данных с основной скоростью (Primary Rate Interface — PRI). Интерфейс BRI обеспечивает два B-канала и один D-канал (2B+D). Служба BRI B-канала, работающая со скоростью передачи данных 64 Кбит/с, используется для передачи пользовательских данных. Служба BRI D-канала, работающая со скоростью передачи данных 16 Кбит/с, обычно используется для передачи информации управления сети ISDN. В некоторых случаях D-канал может быть использован для передачи пользовательских данных. (Например, в Европе он часто применяется для пропуска трафика сетей X.25.) При использовании для передачи данных одного B-канала интерфейс BRI способен обеспечить скорость передачи данных 64 Кбит/с, а с помощью двух B-каналов скорость передачи данных может быть доведена до 128 Кбит/с.

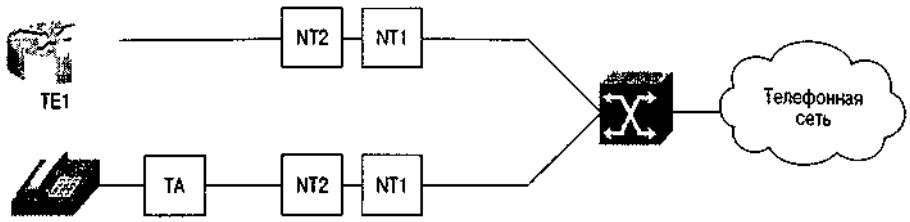


Рис 3.11. Компоненты сети ISDN

Интерфейс PRI предоставляет возможность использования 23 B-каналов и одного D-канала, работающих на скорости 64 Кбит/с (это утверждение верно только для Северной Америки и Японии). Это означает, что интерфейс PRI может использоваться для одновременной поддержки 23 независимых звонков цифровой телефонной связи. В Европе, Австралии и других частях мира ISDN-интерфейс PRI обеспечивает работу 30 B-каналов и одного D-канала, при этом все эти каналы функционируют со скоростью передачи данных 64 Кбит/с.

Идентификатор профиля службы (service profile identifier — SPID) представляет собой номер, который некоторые телефонные компании используют для того, чтобы определить доступные данному ISDN-устройству службы. Во многих случаях идентификатор SPID совпадает с телефонным номером устройства. ISDN-устройство передает значение SPID на ISDN-коммутатор, который затем разрешает устройству доступ в сеть для работы со службой BRI или PRI. Без предоставления правильного значения SPID многие коммутаторы ISDN не позволяют осуществить звонок.

Подкоманды конфигурирования интерфейса ISDN

Конфигурирование интерфейсов ISDN в устройствах Cisco, работающих под управлением ОС IOS, требует указания типа коммутатора, к которому подключается данный интерфейс. Эти данные необходимы, поскольку ISDN-терминалы по-разному ведут обмен данными с коммутаторами разных производителей.

Примечание

Сведения о типах ISDN-коммутаторов, с которыми способно работать используемое устройство, могут быть получены от системы помощи ОС IOS, для чего достаточно ввести isdn switch-type ?. Данные о типе коммутатора, к которому осуществляется подключение, должен предоставить оператор телефонной связи при оформлении заказа на предоставление ISDN-услуг.

Тип коммутатора, к которому подключается устройство с ОС IOS, вводится в конфигурацию с помощью основной команды isdn switch-type. Устройству Cisco необходимо знать

изготовителя ISDN-коммутатора, с которым оно общается, так как каждый изготовитель использует свой собственный протокол обмена сигналами. Без указания типа ISDN-коммутатора устройство Cisco не сможет обмениваться данными с коммутатором, установленным у оператора телефонной связи.

Для каждого интерфейса ISDN BRI необходимо указать значение SPID, для чего используются подкоманды конфигурирования интерфейса isdn spidl и isdn spid2. Каждый идентификатор SPID определяет уникальный B-канал к ISDN-коммутатору. Для интерфейса BRI необходимо указать два разных значения SPID.

Чтобы использовать возможности службы ISDN PRI в устройствах Cisco, необходимо иметь соответствующую аппаратуру. На текущий момент этот тип интерфейса поддерживается в маршрутизаторах и серверах доступа старших и средних моделей, к которым относятся маршрутизаторы серий Cisco 3600, Cisco 4000 и Cisco 7000, а также сервер доступа Cisco 5300.

Интерфейс PRI ведет обмен данными с ISDN-коммутатором с помощью контроллера T1. Контроллер T1 представляет собой набор программного обеспечения канального уровня, который управляет процессом формирования сигналов в канале передачи данных для интерфейса. Для контроллера необходимо задать тип кодирования в линии и метод разбиения на кадры. В примере ниже показывается процесс конфигурирования контроллера T1 для интерфейса seriall/0. В качестве метода разбиения на кадры задается метод расширенного суперкадра ESF (Extended Superframe), а в качестве типа кодирования в линии — кодирование по алгоритму с замещением 8 двоичных нулей B8ZS (binary 8-zero substitution). Также указывается, что интерфейс ISDN PRI будет использовать 24 временных слота. Метод ESF — это тип разбиения на кадры, используемый в каналах контроллера T1. Он предусматривает разбиение на 24 кадра по 192 бит данных *в* каждом, со 193-м битом, используемым для тактирования и других функций. B8ZS — это механизм кодирования, использование которого гарантирует примерно постоянную плотность единичек, передаваемых в канале. Обеспечивается это с помощью замены последовательности из 8 следующих друг за другом нулей специальным кодом, который затем удаляется на дальнем конце соединения.

```
Router#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Router(config)#controller Tl 1/0
Router(config-if)#framing esf
Router(config-if)#linecode b8zs
Router(config-if)#pri-group timeslots 1-24
Router(config-if)#^Z
```

В следующем примере показывается процесс конфигурирования интерфейса BRIO сервера доступа компании ZIP в Сеуле. Этот сервер подключается к ISDN-коммутатору DMS1000 компании Northern Telecom. ISDN-интерфейс конфигурируется на инкапсулирование данных по протоколу PPP. Ввиду того, что технология ISDN обеспечивает лишь метод обработки звонков, а не метод инкапсуляции пакетов канального уровня, указание этого метода (в данном случае — инкапсуляции по протоколу PPP) необходимо:

```
Seoul-ASl#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Seoul-ASl(config)#isdn switch-type basic-dmslOO
Seoul-ASl(config)#interface brio
Seoul-ASl(config-if)#encapsulation PPP
Seoul-ASl(config-if)#tisdn spidl 8864567832
Seoul-ASl(config-if)#isdn spid2 8864567833
Seoul-ASl(config-if)#^Z
```

Статус ISDN-интерфейса можно узнать с помощью команды show interfaces. В примере ниже показывается состояние интерфейса BRIO сервера доступа Seoul-ASl:

```
BRIO is up, line protocol is up (spoofing)
 Hardware is BRI with U interface and external S bus interface
 MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
 Encapsulation PPP, loopback not set
 Last input 00:00:02, output never, 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
   644807 packets input, 2938029 bytes, 0 no buffer
   Received 0 broadcasts, 0 runts, 0 giants
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   700200 packets output, 3329945 bytes, 0 underruns
   0 output errors, 0 collisions, 5 interface resets
   0 output buffer failures, 0 output buffers swapped out
   3 carrier transitions
```

Стоит отметить, что в первой строке вывода указано состояние интерфейса BRIO — *спуфинг* (spoofing). Это означает, что ISDN-интерфейс постоянно эмулирует рабочее состояние, даже если на нем нет ни одного соединения. ISDN-интерфейс имитирует рабочее состояние для протоколов маршрутизации и другого программного обеспечения устройств, работающих под управлением IOS. С помощью этой имитации интерфейс изображает рабочее состояние, принимает пакеты и лишь затем вызывает удаленную сторону. После установки соединения интерфейс переходит в нормальное рабочее состояние, а по истечении определенного времени разрывает соединение, если через него не передаются данные. Затем интерфейс возвращается в режим имитации, и процесс повторяется. Этот механизм — имитация нормальной работы интерфейса, получение данных, инициирование вызова, передача данных и затем разрыв соединения — называется *маршрутизацией с соединением по запросу* (dial-on-demand routing). В четвертой строке указывается, что ISDN-интерфейс использует инкапсуляцию по протоколу PPP.

Резюме

Основные команды конфигурирования ОС IOS для технологий локальных и глобальных сетей, рассмотренных в этой главе, сведены в табл. 3.2. Изучив данную главу и получив представление о технологиях глобальных и локальных сетей, которые работают на канальном уровне модели OSI, можно переходить к изучению сетевого уровня модели OSI и рассмотрению основ конфигурирования устройств Cisco для работы с протоколом IP (Internet Protocol).

- Основная команда interface используется для идентификации интерфейса по его имени и для начала его настройки Существует порядок именования устройств с фиксированной конфигурацией, устройств со сменными платами интерфейсов и устройств, оснащенных VIP-платами.
- Для увеличения эффективности администрирования и в целях документирования рекомендуется с помощью подкоманды description добавлять описание для каждого интерфейса. Введенное с помощью этой команды описание выводится в составе информации, получаемой в результате исполнения команды show interfaces
- Команда shutdown приводит к административному блокированию интерфейса
- Подкоманда encapsulation определяет формат посылаемых данных и протокол канального уровня для конкретного интерфейса Интерфейсы легальных сетей обычно не требуют конфигурирования этого параметра, тогда как интерфейсы глобальных сетей часто требуют его ввода.

Таблица 3.2. Сводная таблица команд конфигурирования ОС IOS для протоколов локальных и глобальных сетей

Протокол	Соответствующие команды	Описание/назначение
Ethernet и Fast Ethernet	**media-type {aui, l0baseT, mii, l00basex}**	Сообщает маршрутизатору о том, какой разъем активен на интерфейсе AUI, RJ-45 или MII

Fast Ethernet и Gigabit Ethernet	**full-duplex**	Активирует на интерфейсе режим полного дуплекса
Token Ring	**ring-speed {4116}**	Задает скорость передачи данных в кольце 4 Мбит/с или 16 Мбит/с
	early-token-ring	Активирует на интерфейсе режим раннего освобождения маркера
X25	**x25 address** *адрес X.121*	Присваивает локальный адрес X 121 последовательному интерфейсу X 25
	x25 ips; x25 ops	Конфигурируют на интерфейсе размер входных пакетов и размер выходных пакетов, соответственно
	x25 win; x25 wout	Конфигурируют размер входного и выходного окна, соответственно
Frame Relay	**frame-relay interface-dlci**	Присваивает интерфейсу DLCI-адрес
ATM	**frame-relay Imi-type**	Устанавливает на интерфейсе тип LMI
	atm pvc	Определяет постоянные виртуальные каналы, существующие на данном интерфейсе
DSL	**set bridging**	Устанавливает метод организации мостового соединения (только для ОС CBOS)
	set interface	Устанавливает параметры интерфейса (только для ОС СВОЗ)
ISDN	**isdn switch-type**	Задает тип коммутатора, к которому подключается устройство с ОС IOS
	isdn spidl; isdn spid2	Определяют значения идентификаторов SPID для каждого интерфейса BRI
	pri-group timeslots	Задает соответствующие временные слоты для интерфейса контроллера
T1	**framing**	Определяет протокол разбиения на кадры в каналах интерфейса контроллера
	linecode	Задает протокол кодирования в линии для каналов интерфейса

		контроллера

В табл. 3.3 приведены основные команды режима EXEC, используемые для проверки конфигурирования интерфейсов.

Таблица 3.3. Сводная таблица команд режима EXEC для локальных и глобальных сетей

Команда	Описание
show frame pvc	Выводит статистические данные для постоянных виртуальных каналов Frame Relay
show frame *список svc*	Выводит статистические данные для коммутируемых виртуальных каналов Frame Relay
`show interfaces`	Выводит статистические данные для интерфейсов устройства
`show x25 vc`	Выводит статистические данные для виртуальных каналов X 25

Дополнительная литература

Дополнительная информация по вопросам, изложенным в данной главе, содержится в следующих изданиях.

1. Cisco Systems, at al. *Internetworking Technologies Handbook*, Third Edition. Indianapolis Indiana: Cisco Press, 2001. (Готовится к изданию перевод на русский язык это книги, который должен выйти в конце 2001 года в ИД "Вильяме".)
2. Stallings, W. *Networking Standards: A Guide to OSI, ISDN, LAN and IVAN Standard.* Reading, Massachusetts: Addison-Wesley Publishing Company, 1993.

Глава 4

Основы TCP/IP

Ключевые темы этой главы

- **TCP/IP-адресация** Основные элементы структуры адреса и классы сетей в протоколе IP
- **Конфигурирование IP-адресов** Обзор критериев выбора и способов организации адресного пространства Примеры конфигурирования адреса для различных типов глобальных и локальных сетей
- **Конфигурирование IP-маршрутизации** Основы конфигурирования маршрутизации — статические маршруты, бесклассовая маршрутизация, сводные маршруты и маршруты по умолчанию, а также соответствующие команды show.
- **Конфигурирование протоколов IP-маршрутизации** Характеристики основных протоколов динамической маршрутизации и примеры базовой конфигурации для каждого из них Команды distnbute-list, passive-interface и no auto-summary
- **Конфигурирование IP-фильтрации с помощью списков доступа** Управление доступом к сети и ее защита путем применения команд access-list, ip access list И access-group
- **Конфигурирование основных IP-служб работы с коммутируемыми каналами передачи данных** Конфигурирование удаленного доступа при асинхронных и ISDN-соединениях
- **Верификация IP-взаимодействия и устранение неполадок** Идентификация проблем взаимодействия с помощью команд show, ping, trace и debug
- **Конфигурирование других опций протокола IP** Примеры конфигурирования служб имен доменов, переадресация широковещательных пакетов, IOS DHCP-сервер и протокол маршрутизатора горячего резерва Hot Standby Router Protocol

В данной главе рассматриваются вопросы, связанные с конфигурированием и настройкой популярного протокола Transmission Control Protocol/Internet Protocol (протокол управления передачей/межсетевой протокол), обычно называемого протокол! TCP/IP, в работающих под управлением ОС IOS устройствах компании Cisco. Разработанный еще в середине 70-х годов в рамках проекта Управления перспективных и следований и разработок Министерства обороны США (DARPA) по обеспечению и следовательских организаций и университетов коммуникационными услугами в мг штабах страны протокол TCP/IP стал фактическим стандартом протокола для сетевых систем с разнородными компьютерами

Глава начинается с краткого обзора некоторых основных положений протокола TCP/IP Рассматривается система адресации и классы сетей, а также вопросы организации адресного пространства сети Однако основное внимание в этой главе уделяется конфигурированию протокола TCP/IP в ОС IOS компании Cisco Более подробное описание протокола TCP/IP приводится в одной из нескольких по вившихся недавно монографий (смотрите в конце главы раздел "Дополнительная литература").

TCP/IP-адресация

Данный раздел знакомит со структурой IP-адреса, его элементами, определяют ми сеть, подсеть и хост-машину Здесь также объясняется, как пользователь выбирает систему IP-адресации и команды конфигурирования, чтобы реализовать желаем схему адресов.

Структура адреса

Протокол TCP/IP представляет собой группу коммуникационных протоколов, которые определяют то, как адресуются в сети различные компьютеры, какой метод используется для перемещения информации из одного компьютера в другой, а так некоторые доступные компьютерам услуги При реализации функций маршрутизации и коммутирования маршрутизатор имеет дело прежде всего с сетевым уровнем (IP) транспортными уровнями (UDP и TCP) модели OSI.

Стандарты протокола TCP/IP

Протокол TCP/IP часто называют *открытым стандартом.* Это означает, что ни одна компания или лицо не контролируют спецификации протокола или его работу. Эволюцией протокола управляет руководящий орган, который называется Комитетом по инженерным проблемам Internet (Internet Engeneering Task Force— IETF) и состоит из экспертов по вопросам организации сетей и представителей компаний. Рабочие группы из состава комитета IETF рассматривают, обсуждают, рекомендуют и утверждают предлагаемые изменения в стандартах с помощью инструмента, называемого *запросом на комментарий* (Request For Comment— RFC). Все концепции и многие из тем, рассматриваемых в данной главе, описаны в сотнях таких запросов, которые и составляют стандарты протокола TCP/IP. Хотя запросы на комментарий часто написаны сухим языком и по своей природе являются техническими документами, они, тем не менее, дают наиболее полные определения протоколов в стандарте TCP/IP. Что касается этих документов в письменном виде, то их можно получить на Web-сервере Института информатики Южно-калифорнийского университета (Information Sciences Institute of the University of Southern California — ISI) по адресу www.rfc-editor.org/rfc.html

Межсетевой протокол Internet Protocol (IP), являясь составляющей протокола TCP/IP, отвечающей за адресацию, работает в рамках уровня 3 модели OSI. Каждая рабочая станция, которая хочет обмениваться данными с другой рабочей станцией, обладает уникальным IP-адресом (как каждый дом на улице имеет свой адрес). IP-адрес более сложен, чем адрес дома (компьютерам нравятся нолики и единички), но после непродолжительного изучения он не будет казаться столь таинственным.

На уровне 4 модели OSI в стандарте TCP/IP существуют два основных транспортных протокола: протокол дейтаграмм пользователя (User Datagram Protocol — UDP) и TCP.

Транспортные протоколы отвечают за базовые механизмы передачи данных, управление потоком, надежность и проверку ошибок при обмене данными между рабочими станциями. Протокол UDP считается ненадежным, поскольку пакеты, пересылаемые с его помощью, не сопровождаются подтверждениями принимающей станции. Он относится к протоколам, не ориентированным на соединение, так как станция-отправитель не сообщает станции-получателю о необходимости сформировать коммуникационный канал для пересылки данных. В отличие от протокола UDP протокол TCP ориентирован на соединение, поскольку посылающая станция должна сообщать станции-получателю о своем желании сформировать коммуникационный канал. Отправляемые по протоколу TCP пакеты снабжаются тэгами с порядковыми номерами, и как посылающая, так и получающая станции подтверждают получение пакетов друг от друга.

IP-адрес представляет собой 32-разрядный двоичный адрес, записываемый четырьмя группами по 8 бит, которые называются *октетами*. Полный адрес отражает три компонента модели адресации протокола IP: элементы адреса, которые определяют сеть, подсеть и хост-машину. Рассмотрим числовое выражение адресов.

Типичный IP-адрес, записанный в виде 32-разрядного двоичного числа, может выглядеть, например, следующим образом:

```
10101100.00010000.00000001
```

Каждый из восьми битов в октете принимает значение 0 или 1. Поэтому в каждом октете значения могут находиться в диапазоне от 00000000 до 11111111. Оперирование с 32-разрядными адресами в двоичной форме обременительно и подвержено ошибкам. Осознав этот факт, разработчики протокола TCP/IP решили, что двоичную форму следует оставить компьютерам, а IP-адреса преобразовывать в десятичную форму (традиционный способ восприятия чисел людьми). Октет, в котором биты на всех позициях установлены в значение 1, в десятичной форме эквивалентен числу 255:

```
1   1   1   1   1 1 1 1 =  Позиции двоичных разрядов
128 64  32  16   8 4 2  1  = Десятичный  эквивалент
```

Суммируем десятичный эквивалент двоичного числа:

```
128+64+32+16+8+2+1  =  255
```

Теперь давайте преобразуем пример нашего адреса:

```
1    0  1    0  1  1  0  0  .  0    0    0  1  0  0  0  0  .
128  64 32 16   8  4  2  1   .128   64   32 16 8  4  2  1.
128+0+32+0+8*4+0+0=172    0+0+0+16+0+0+0+0=16
0    0    0  0  0  0  0  1  .  0    0  0    0  0  0  0  1  .
128 64   32 16 8  4  2  1.   128   64 32    16 8  4  2  1.
0+0+0+0+0+0+0+1=1           0+0+0+0+0+0+0+1=1
```

Таким образом, десятичное обозначение этого IP-адреса имеет вид 172.16.1.1.

IP-адрес отражает три компонента модели IP-адресации: сетевую составляющую, подсетевую составляющую и составляющую хост-машины. Эти данные описывают различные уровни детализации объекта из группы находящихся в сети систем. Составляющая хост-машины принадлежит к наиболее высокому уровню детализации и описывает адрес отдельной рабочей станции или отдельного сервера. Сетевая составляющая относится к наиболее общему уровню и описывает адрес группы хост-машин, принадлежащих одной логической компьютерной сети. Подсетевая составляющая относится к промежуточному уровню между сетевой составляющей и составляющей хост-машины и описывает адрес подсети хост-машин внутри общего адресного пространства сети.

Подсеть создается "заимствованием" части составляющей хост-машины для формирования подгрупп внутри одной логической сети. Обычно подсетевая составляющая идентифицирует

группу систем, находящихся внутри сегмента локальной или глобальной сети. Читаемый слева направо IP-адрес проходит от наименее определенной части адреса системы (сетевая) через более детальную часть (подсетевая) к наиболее подробной части (хост-машина). Местоположение границ между тремя уровнями адреса зависит от класса адреса и его разбиения на подсети.

В соответствии с исходными определениями, изложенными в запросах на комментарий, существует пять классов адресов сетей, отличающихся количеством начальных битов адреса, устанавливаемых в значение 1.

- Первоначально сетевые адреса класса А предназначались для очень крупных сетей. В адресах класса А первый бит первого октета резервируется и устанавливается в значение 0, а следующие семь битов используются для идентификации сетевой составляющей. Три оставшихся октета образуют составляющую хост-машины. При подобном группировании адреса класса А обеспечивают адресацию относительно небольшого количества сетей, но каждая сеть может вмещать внутри данного адресного пространства большое количество хост-машин.
- В сетевых адресах класса В резервируются два первых бита первого октета; первому биту присваивается значение 1, а второму — 0. Подобная конструкция дает адресам класса В 14 разрядов для сетевой составляющей и 16 разрядов для составляющей хост-машины. Сетевые адреса класса В допускают приблизительно равное количество сетей и хост-машин в этих сетях.
- В сетевых адресах класса С резервируются три первых бита первого октета; первым двум битам присваивается значение 1, а третьему — 0. Такая конструкция дает адресам класса С 22 разряда для сетевой составляющей и только 8 разрядов для составляющей хост-машин. Могут существовать миллионы сетей класса С, однако каждая из них способна поддерживать только 255 хост-машин.
- Адреса класса D резервируются для сетей с групповым вещанием. В адресах класса D резервируются четыре первых бита первого октета, при этом первым трем битам присваивается значение 1. Адрес группового вещания принадлежит не какой-либо одной рабочей станции, а представляет группу станций, настроенных на прием информации, При групповом вещании станция посылает один поток информации по конкретному IP-адресу группового вещания. А затем сетевые устройства, например, маршрутизаторы или коммутаторы, реплицируют этот поток и посылают его сразу многим станциям, которые должны его получить.
- Адреса класса Е определены протоколом IP. В настоящее время они не используются, так как зарезервированы для применения в будущем. В адресах класса Е значение 1 присваивается первым четырем битам первого октета.

На рис. 4.1 представлена структура адреса для сетей классов А, В и С.

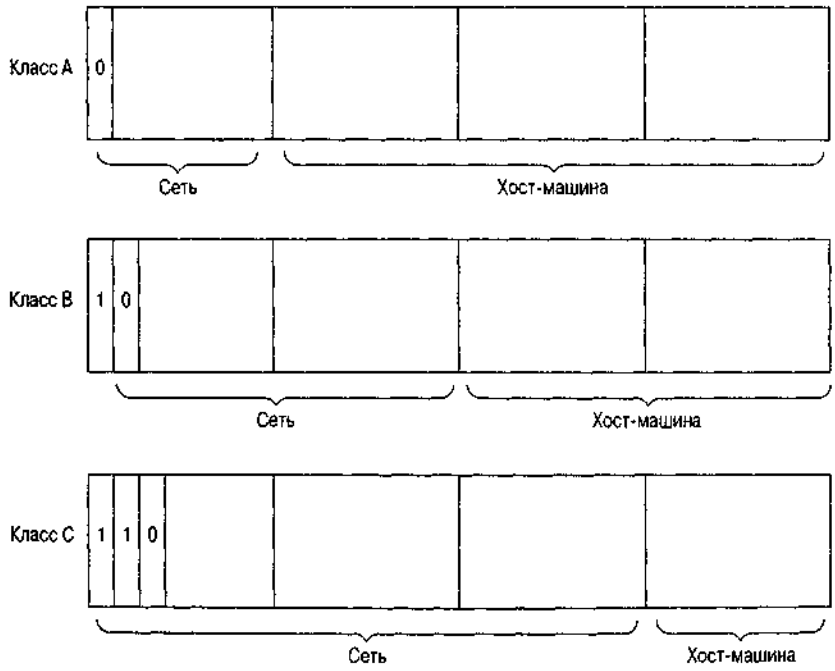


Рис. 4.1. Структура адресов классов A, B и C

При преобразовании IP-адреса из десятичной формы в двоичную путем подсчета количества начальных битов, которым присвоено значение 1, легко определить, к какому классу сетей принадлежит адрес. При отсутствии разбиения на подсети знание класса, к которому принадлежит адрес, говорит о том, какую часть адреса следует читать как сетевую, а какую — как часть, задающую хост-машину. Чтобы доставить данные по месту назначения, устройства, например маршрутизаторы, должны расшифровывать эту информацию.

Однако, если сеть разбита на подсети, сразу определить, какая часть составляющей хост-машины заимствована для адресации подсети, нельзя. Решить этот вопрос позволяет маска подсети (обычно ее называют *сетевой маской).* Как и IP-адрес, сетевая маска представляет собой сгруппированное в четыре октета 32-разрядное двоичное число, которое может выражаться в десятичной форме. Однако в отличие от IP-адреса в сетевой маске битам присваивается значение 1 на всех позициях, кроме той части IP-адреса, которая относится к хост-машине.

Например, сеть класса B без разбиения на подсети имеет маску 255.255.0.0, в которой 16 старших разрядов обозначают сетевую часть IP-адреса, а 16 младших разрядов относятся к части, отвечающей за адресацию хост-машины. Сеть класса B, в которой семь битов части адреса, относящейся к хост-машине, были использованы для разбиения на подсети, будет иметь маску 255.255.254.0. Если взять сеть класса C с четырьмя битами для разбиения на подсети, то она будет иметь маску 255.255.255.240. На рис. 4.2 показана взаимосвязь между сетевой маской и IP-адресом.

Разбиение на подсети позволяет сетевым администраторам присваивать каждому сегменту локальной или глобальной сети уникальный сетевой идентификатор, а не требовать отдельного пространства сетевых адресов для каждого из них. Например, вместо одного сетевого адреса класса B, имеющего один логический сегмент сети, который вмещает более 65 000 хост-машин, схема с разбиением на подсети (из составляющей хост-машины заимствуется 8 бит) позволяет иметь 255 логических сегментов сети с 255 хост-машинами каждый. Вводя пару из IP-адреса и его сетевой маски, можно точно определить, какие биты адреса соответствуют сетевой составляющей, составляющим подсети и хост-машины. Например, IP-адрес 131.108.3.4 с сетевой маской 255.255.0.0 имеет сетевую составляющую 131.108.0.0, составляющую хост-машины 3.4 и не имеет подсетевой составляющей. А IP-адрес 131.108.3.4 с сетевой маской 255.255.255.0 имеет сетевую составляющую 131.108.0.0, подсетевую составляющую 3 и составляющую хост-машины 4.

Благодаря современным протоколам маршрутизации, которые в последних своих версиях переносят информацию не только о сети, но и о сетевой маске, в одной логической IP-сети можно использовать несколько сетевых масок. Это повышает эффективность применения IP-адресов.

Концепция сетевой маски была использована для разбиения на подсети. В ответ на бурный рост глобальной сети Internet, количества запрашиваемых сетевых IP-адресов, нехватку адресного пространства IP-адресов и увеличение таблицы глобальной IP-маршрутизации организации, занимающиеся выдачей и регистрацией IP-адресов, прекратили выдачу IP-адресов с определяемыми классом границами. Вместо этого они получили возможность выбирать способ объединения нескольких сетевых IP-адресов заданного класса в так называемую *суперсеть,* или блок бесклассовой междоменной маршрутизации (classless interdomain route block — CIDR).

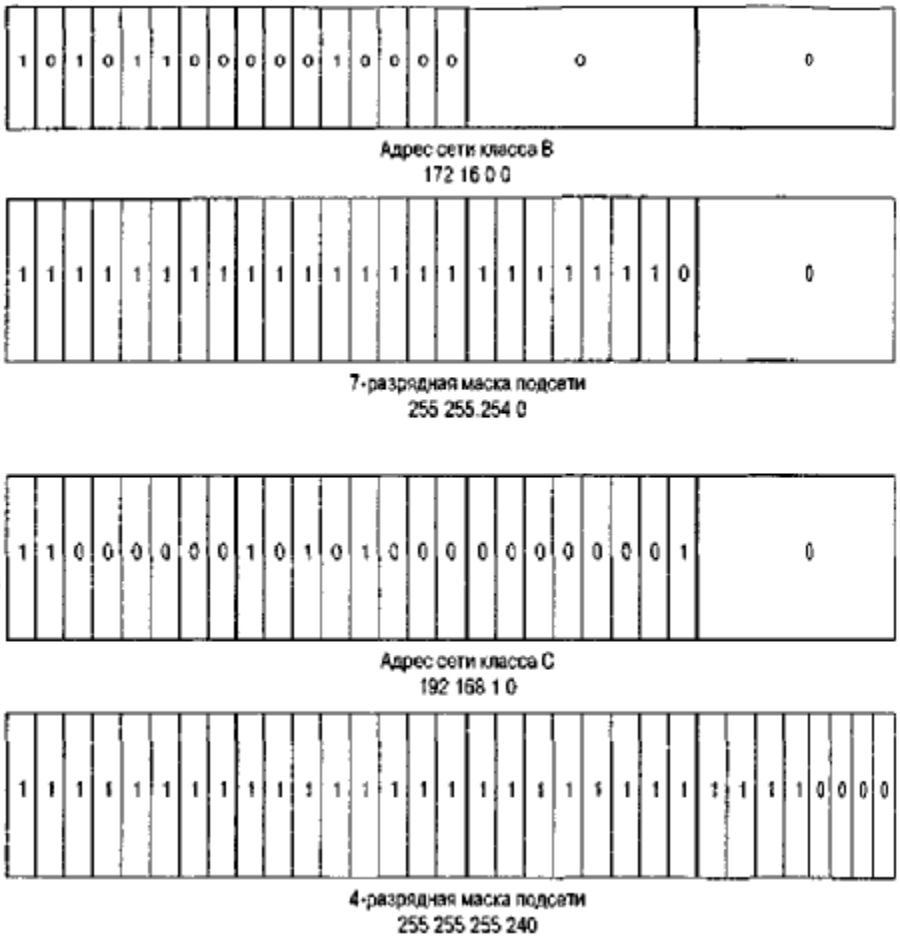


Рис. 4.2 Примеры сетевых масок

Кроме того, некоторые из ранее существовавших сетей класса A были разделены и распределены компаниям и провайдерам Internet-услуг в виде более мелких CIDR-блоков. В прошлом компании или провайдеру предоставлялась сеть класса B. Сегодня может быть выделено 255 адресов класса C, лежащих в диапазоне от 209.32.0.0 до 209.32.255.0. Если блок адресов не разделяется внутри на подсети, естественной маской таких сетей класса C является маска 255.255.255.0. Однако при укорочении маски и создании суперсети из таких адресов та же группа адресов может быть представлена сетевым адресом 209.32.0.0 и сетевой маской 255.255.0.0. Затем организация, получившая ClDR-блок, свободна делить пространство сетевых адресов либо между подсетями внутри своей логической сети, либо между своими клиентами.

Тот же метод может быть применен к адресам класса A в обратном порядке. Ранее сетевой адрес 12.0.0.0 с естественной сетевой маской 255.0.0.0 был бы приписан одной компании или провайдеру Internet-услуг. Теперь этот сетевой адрес рассматривается в качестве блока адресов, более мелкие части которого могут выделяться нескольким субъектам. Например, группа адресов с 12.1.0.0 по 12.1.255.0 может быть представлена в виде одного CIDR-блока с сетевым адресом 12.1.0.0 и сетевой маской 255.255.0.0. Ввод разбивки таких исходно больших блоков сетевых адресов сделал доступным большее количество сетевых IP-адресов и замедлил их расход.

Запись и описание сетевых адресов в виде четырех разделяемых точками октетов в десятичной форме, за которыми следует четыре октета в той же форме, принадлежащих сетевой маске, всегда были в определенной степени обременительными. При присвоении CIDR-блокам адресов желательно было иметь более точный и компактный способ описания адресного пространства. Создание бесклассовой системы сетевых IP-адресов дало сетевому сообществу новый стенографический метод записи сетевых IP-масок.

Согласно этому стенографическому методу вместо четерехоктетной разделяемой точками и записываемой в десятичной форме маски используется прямая косая линия "/", после которой

указывается количество битов, которым присваивается значение 1. Сетевая маска 255.255.0.0 имеет 16 бит единичек. Поэтому она может быть записана в виде /16 (произносится "косая, 16"). Сетевая маска 255.255.252.0 имеет 22 бит единичек, так что она может быть записана как /22. Такой тип маски известен под названием маски с контрольной суммой (bit-count mask). Объединив подобный тип записи маски с записью сетевого IP-адреса, получим укороченную форму 131.108.0.0/16, которая может быть использована для представления сети 131.108.0.0 с маской 255.255.0.0. Аналогично, запись 206.220.224.0/22 может быть использована для представления адреса 206.220.224.0 с маской 255.255.252.0 (который сам является CIDR-блоком, представляющим адреса класса С с 206.220.224.0 по 206.220.227.0, с маской 255.255.255.0).

Примечание

Во время диалога конфигурирования системы, описанного в главе 2, "Основы конфигурирования устройств", предполагалось, что все сетевые адреса попадают в границы описанных ранее классов сетей. Задаваемый пользователю вопрос *Number of bits in subnet field[0] (Количество битов в поле подсети [0])* собственно спрашивает о количестве битов составляющей адреса, относящейся к хост-машине, которое следует использовать для организации подсетей на основе номера класса сети, введенного пользователем. Если номер сети представляет сеть класса А, например 17,0.0.0, то для организации подсетей можно было бы использовать 24 бит поля хост-машины. Если же пользователь указывает, что для разбиения на подсети используется девять битов, то ОС IOS вычисляет соответствующую сетевую маску — в данном случае это 255. 255.128 . 0.

Конфигурирование IP-адресов

До назначения каких-либо адресов следует принять решение о том, какое адресное пространство используется для устройств вашей сети, и как данное адресное пространство распределяется Это решение очень важно — то, как будут назначены адреса сейчас, может оказать значительное влияние на сеть в будущем. Ответы на следующие вопросы помогут определить, какое адресное пространство использовать.

- Будет сеть подключаться к глобальной сети Internet через провайдера Internet-услуг или через провайдера сетевых услуг? Если да, то будет ли такое подключение осуществляться не через одного провайдера Internet- или сетевых услуг?
- Будет ли сеть иметь прямое соединение с сетью другой компании (например, с сетью родительской компании)?
- Сколько уникальных сегментов локальной и глобальной сети необходимо будет иметь в сети?
- Сколько уникальных хост-машин будет размешаться в типовом сегменте локальной сети? Каково их максимальное и минимальное количество?

Если сеть будет подключаться к глобальной сети Internet или к сети другой компании, то для нее важно иметь пространство уникальных сетевых адресов. Если выбираются сетевые адреса, совпадающие с теми, которые используются в другой сети, то маршрутизаторы сети Internet не смогут правильно различать дублирующиеся адреса. Если сеть подключается к сети Internet через одного провайдера Internet-или сетевых услуг, то обычно провайдер предоставляет пространство уникальных адресов из того большого пула, который был выделен ему организацией, ведущей реестр сетевых адресов. К таковым относятся: Американский реестр Internet-номеров (American Registry for Internet Numbers — ARIN), Европейский реестр IP-адресов (Reseaux IP Europeens — RIPE) и Азиатско-тихоокеанский информационный центр по сетям (Asia Pacific Network Information Center — APNIC). Провайдер Internet-услуг выделяет адресное пространство для сети на основе таких факторов, как количество хост-машин в сети, количество физических сегментов локальной и глобальной сети и ожидаемый рост сети.

Если сеть подключается к нескольким сервис-провайдерам, возможны два варианта получения адресного пространства. По первому варианту сеть получает IP-адреса от одного провайдера Internet-сервиса. Поскольку эти адреса назначаются из адресного пространства провайдера Internet-услуг, то входной трафик сети проходит через сеть этого сервис-провайдера. Предположим, что ваша сеть

подключена к нескольким сервис-провайдерам, так что трафик, выходящий из сети, может идти по другому пути, нежели входной трафик. Подобная ситуация известна под названием *асимметричная маршрутизация*. Этот сценарий подходит для сети, в которой доминирующим является исходящий трафик, и нужно разделить нагрузку. Такой метод также используется, если дополнительное соединение с провайдером Internet-услуг предназначается исключительно для дублирования (на случай отказа). Не рекомендуется применять этот вариант, если доминирующим трафиком является входной трафик, и стоит задача разделения нагрузки между несколькими провайдерами Internet-услуг.

Второй сценарий получения адресного пространства при подключения сети к нескольким провайдерам Internet-услуг — это прямой запрос пространства в региональном реестре. Ведущие реестр организации не поощряют подобную практику и имеют строгие правила относительно непосредственного выделения IP-адресов сетям конечных пользователей. Бурный рост количества уникальных сетей в рамках глобальной сети Internet привел к нехватке доступного адресного пространства и экспоненциальному росту таблиц маршрутизации во всей сети Internet. Эти проблемы и заставили регистрирующие организации перейти к строгой политике распределения IP-адресов.

Прямой запрос адресов в реестре подходит для ситуаций, когда доминирующим трафиком сети является входной трафик, и есть необходимость распределить эту нагрузку между несколькими сервис-провайдерами. Недостатком запроса адресного пространства непосредственно в реестре является то, что выдающий орган может выделить вашей сети только очень небольшой объем адресов. В результате, не все сервис-провайдеры сети Internet будут распространять информацию о вашей сети в глобальном масштабе. Если же информация о вашей сети не имеет широкой доступности, то будут сети, которые не смогут добраться до вашей сети и наоборот.

Примечание

Описание порядка запроса пространства IP-адресов у организаций, ведущих реестр, можно найти на Web-сервере соответствующего органа.
Американский реестр Internet-номеров — www.arin.net
Европейский реестр IP-адресов — www.ripe.net
Азиатско-тихоокеанский информационный центр по сетям — www. apnic. net

Если подключение к глобальной сети Internet не планируется, или если вы намерены использовать современный брандмауэр и методику трансляции сетевых адресов (Network Address Translation — NAT), реализованную в таких продуктах, как Private Internet' Exchange (PIX) (Межсетевой обмен в частных сетях) компании Cisco Systems, то тогда в высшей степени желательно использование IP-адресов, относящихся к классу адресов, установленных Комитетом по инженерным проблемам Internet (IETF) для использования в качестве частных. Адреса этого класса считаются частными, поскольку информация о таких сетях не распространяется по сети Internet ни одним провайдером Internet- или сетевых услуг. Поскольку информация об этих адресах не распространяется, они могут повторно использоваться многими компаниями, сберегая тем самым количество доступных широкой публике адресов. Диапазон частных IP-адресов определен в Запросе на комментарий № 1918 "Выделение адресов для частных сетей Internet" следующим образом:

```
10.0.0.0 - 10.255.255.255
172.16.0.0 - 172.31.255.255
192.168.0.0 - 192.168.255.255
```

После того как адреса назначены провайдером Internet-услуг или реестром либо было выбрано для использования пространство частных адресов, полученное адресное пространство должно быть распределено по всей сети. Способ распределения адресного пространства зависит главным образом от того, сколько хост-машин будет подключено к данному сегменту локальной сети, сколько всего сегментов локальной/глобальной сети будет в планируемой сети, и какой объем адресного пространства доступен для использования. Если сеть использует частные IP-адреса, то объем доступного адресного пространства не является предметом беспокойства. Частная IP-сеть 10.0.0.0 в зависимости от схемы

выделения адресов подсетям может поддерживать до четырех миллионов хост-машин или сегментов локальной/глобальной сети. Тут администратор сети может принять решение о выделении всем сегментам локальной или глобальной сети 24-разрядных адресов подсетей сети 10.0.0.0. Это позволяет иметь 255 хост-машин в каждом данном сегменте, что более чем достаточно для большинства сегментов локальных сетей и при использовании сегментов в технологии двухточечной глобальной сети с лихвой накрывается всего двумя устройствами.

Если пространство IP-адресов было выделено провайдером Internet-услуг или реестром и, возможно, в условиях высокого спроса, сетевой администратор может выбрать путь назначения сегментам локальной и глобальной сети подсетей переменной длины. Например, двухточечным сегментам глобальной сети вместо сетевого адреса, который может поддерживать более двух устройств, назначается сетевой адрес с 30-разрядной маской. Тогда единственное пространство адресов класса С, которое способно поддерживать 255 устройств, при разбиении на подсети с помощью 30-разрядной маски может быть сконфигурировано на поддержку 64 двухточечных сегментов глобальной сети. Аналогичный подход может быть использован и в отношении сегментов локальной сети, когда выбирается схема разбиения на подсети и сетевые маски, которые поддерживают только то количество устройств, которое реально будет размещаться в данном сегменте. Например, небольшому удаленному офису, в котором работает всего 10 человек, не нужен адрес, который может поддерживать 128 пользователей.

Если для создания подсетей, поддерживающих переменное количество хост-машин, используется несколько различных масок, то вполне вероятно, что выделенное сети адресное пространство будет использоваться более эффективно и не так быстро расходоваться.

Совет

Мы рекомендуем, чтобы вне зависимости от объема выделенного сети адресного пространства всегда использовалась эффективная схема разбиения на подсети, которая бы не приводила к избыточному выделению адресов таким сегментам, как интерфейсы двухточечной глобальной сети. Что касается написанного, то в Центре технической поддержки компании Cisco Systems был создан прикладной продукт IP Subnet Design Calculator (Калькулятор проектировщика IP-подсетей), который доступен зарегистрированным сертифицированным пользователям устройств компании Cisco на сервере по адресу www cisco.com/techtools/ip_addr.html. Этот продукт поможет в выборе и проектировании схем IP-нумерации.

Конфигурирование интерфейса локальной сети

Такие устройства, как, например, маршрутизаторы, имеют уникальный адрес в каждом сегменте подключенной к ним локальной сети. Таким образом, маршрутизатор знает, какие сети подключены к каждому интерфейсу и куда следует посылать пакеты для этих сетей. В отличие от них, такие устройства, как мосты и коммутаторы, имеют только один IP-адрес во всей системе. Обычно этот IP-адрес используется исключительно для удаленного администрирования и управления сетью.

Каждый из пяти типов локальных сетей (Ethernet, Token Ring, Fast Ethernet, Gigabit Ethernet и FDDI), описанных в главе 3, "Основы интерфейсов устройств Cisco", поддерживает концепцию динамического отображения канального адреса (обычно называемого MAC-адресом) сетевого адаптера на IP-адрес, присвоенный интерфейсу. Этот процесс, который называется преобразованием адресов, поддерживается протоколом, называемым протоколом разрешения адресов (Address Resolution Protocoll — ARP).

Когда одной IP-станции необходимо связаться с другой IP-станцией, находящейся в той же логической сети, и она не знает канального адреса этой станции, IP-станция посылает широковещательный запрос на поставку канального адреса для нужного IP-адреса Этот процесс показан на рис. 4 3. Каждая станция в этой логической сети проверяет запрос и, если запрашиваемый IP-адрес совпадает с ее адресом, она отвечает своим MAC-адресом. Поэтому станции не надо знать, какие конкретно MAC-адреса действуют в ее логической сети, чтобы общаться с ними. Однако многие протоколы глобальных сетей не поддерживают динамического

отображения канального адреса на IP-адрес и требуют для взаимодействия с другими станциями в рамках интерфейса глобальной сети дополнительного конфигурирования IP-адресов.

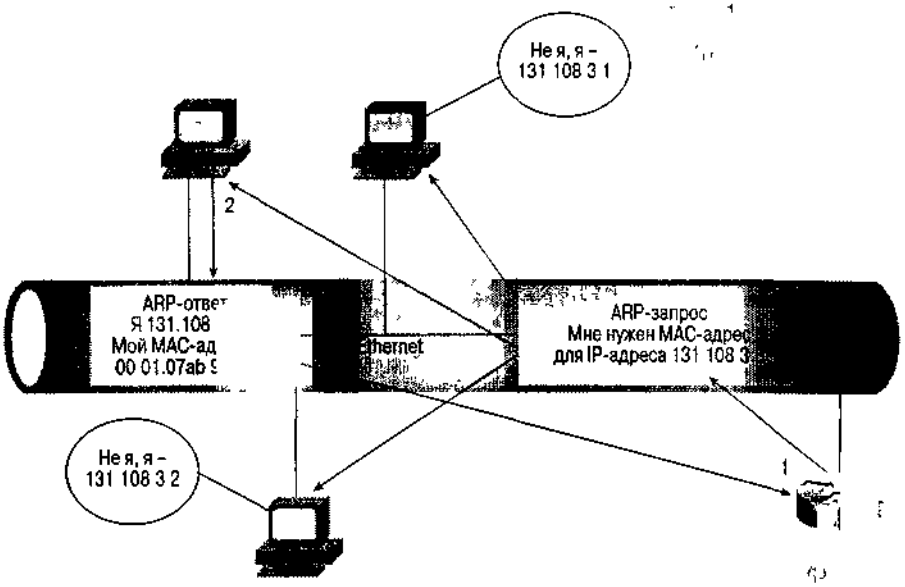


Рис. 4 3 IP-маршрутизатор посылает ARP-запрос о неизвестном MAC-адресе IP-получателя

Для проверки конфигурации интерфейса локальной сети воспользуемся сетевыми IP-адресами, выбранными для сети компании ZIP, которые сведены в табл. 4.1.

Таблица 4.1. Распределение сетевых IP-адресов сети компании ZIP

Сегмент сети	Назначенный сетевой IP-адрес и маска
Сводный маршрут сети	131.108.0.0/16
Сингапур, локальная сеть Ethernet	131.108.1.0/25
Куала-Лумпур, локальная сеть Ethernet	131.108.2 .0/25
Сеул, локальная сеть Ethernet	131.108.3.0/25
Сан-Франциско, локальная сеть Fast Ethernet	131.108.20.0/22
Сан-Хосе, локальная сеть Token Ring	131.108.100.0/24
SF-1, локальная сеть Ethernet	131.108.101.0/24
SF-2, первая локальная сеть Ethernet	131.108.110.0/24
SF-2, вторая локальная сеть Ethernet	131.108.120.0/24
SF-Core-1->Сан-Хосе, двухточечная глобальная сеть	131.108.240.0/30
SF-Core-2 ->Seoul-2, двухточечная глобальная сеть HDLC	131.108.240.4/30
Сан-Хосе ->Seoul-1, двухточечная глобальная сеть HDLC	131.108.241.0/30
Seoul-1 -> Куала-Лумпур, двухточечная глобальная сеть Frame Relay	131.108.242.0/30
Seoul-1 -> Сингапур, двухточечная глобальная сеть Frame Relay	131.108.242.4/30
Интерфейсы колец обратной связи отдельных маршрутизаторов	131.108.254.0/32
ZIPnet -> Провайдер Internet-сервиса, двухточечное HDLC-подключение к Internet в Сан-Франциско	192.7 .2.0/30 (выделено провайдером Internet-сервиса)
ZIPnet-> Провайдер Internet-сервиса, двухточечное HDLC-подключение к Internet в Сеуле, Корея	211.21.2.0/30 (выделено провайдером Internet-сервиса)

На рис. 4.4 показана топология логических IP-адресов всей сети компании ZIP.

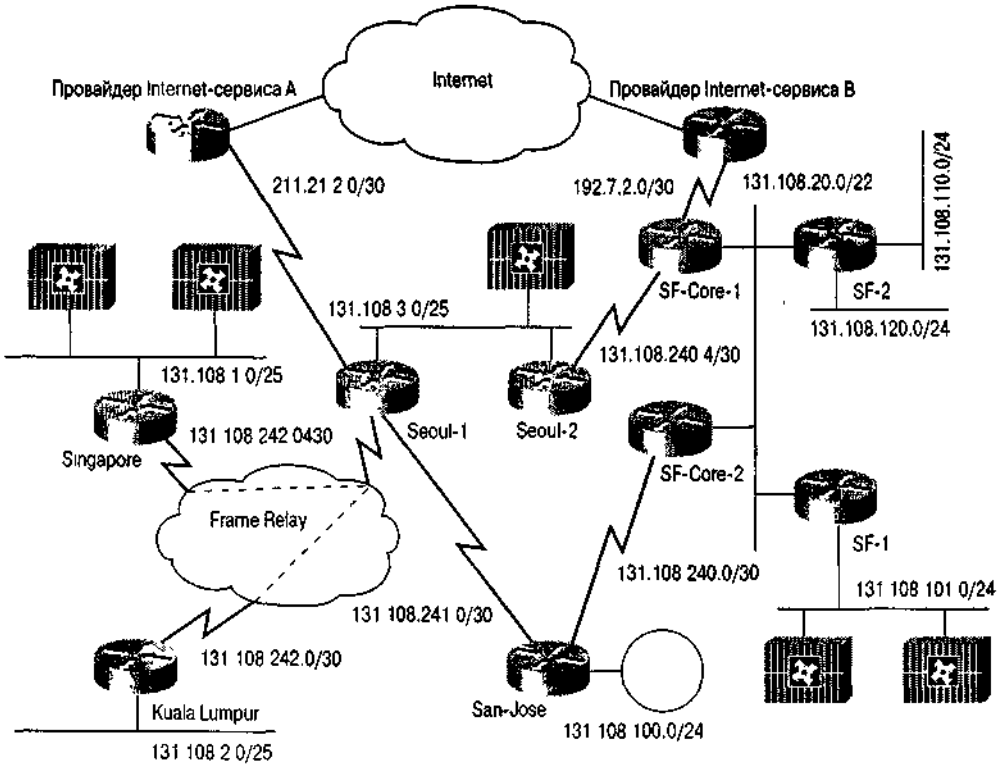


Рис. 4.4 Топология IP-адресов сети компании ZIP

В дополнение к назначению сетевых IP-адресов также были присвоены следующие IP-адреса рабочим станциям, выполняющим в сети компании ZIP указанные функции:

- 131.108.20. 45 — станция SNMP-управления;
- 131.108.21.70— корпоративный DHCP-сервер и WINS-сервер;
- 131.108.101.34— первичный почтовый SMTP-сервер и DNS-сервер;
- 131.108.101.35 — вторичный почтовый SMTP-сервер и DNS-сервер;
- 131.108.101.100 — WWW-и PTP-сервер;
- 131.108.110.33 — Syslog, TACACS+ и RADIUS-сервер.

Назначение IP-адресов интерфейсам локальных и глобальных сетей выполняется субкомандой конфигурирования интерфейса ОС IOS ip address. Эта команда требует, чтобы были указаны как IP-адрес, так и его сетевая маска.

В примере ниже выполняется конфигурирование маршрутизатора SF-2 на IP-адреса для каждого из трех его интерфейсов локальной сети. В каждом случае субкоманда ip address предваряется основной командой interface, что делается для того, чтобы указать интерфейс локальной сети, к которому должна будет применяться команда ip address.

```
SF-2#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-2(config)#interface ethernet 0
SF-2(config-if)#ip address 131.108.110.1 255.255.255.0
SF-2(config-if)#interface ethernet 1
SF-2(config-if)#ip address 131.108.120.1 255.255.255.0
SF-2(config-if)#interface fastethernet 0
SF-2(config-if}#ip address 131.108.20.2 255.255.252.0
SF-2(config-if)#^Z
```

Совет
Рекомендуется резервировать в начале или в конце сетевого адресного пространства каждой локальной сети некоторое количество IP-адресов для маршрутизаторов и других устройств инфраструктуры сети. Наличие постоянной группы адресов для различных сетевых устройств в сегментах всех локальных сетей помогает в процессе устранения неисправностей, обеспечивая более быстрое распознавание конкретного IP-адреса.

В некоторых случаях выделенный сети объем пространства IP-адресов может потребовать использования подсети, являющейся первой в диапазоне адресов. Эту первую подсеть принято называть *нулевой подсетью,* поскольку все биты подсетевой части сетевой маски имеют значение 0. Более старые протоколы маршрутизации испытывали трудности при попытке отличить основную сеть, например 131.108.0.0, и подсеть 131.108.0.0. Поэтому маршрутизаторы, как правило, не допускают использования первой подсети. Ниже показан пример попытки использования в маршрутизаторе SF-1 нулевой подсети:

```
S F-1# configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-1(config)#interface ethernet 1
SF-1(config-if)#ip address 131.108.0.1 255.255.255.128
Bad mask 255.255.255.128 for address 131.108.0.1
SF-1(config-if)#^Z
```

В данном примере маршрутизатор предупреждает пользователя о том, что сетевая маска неудачна, поскольку тот сделал попытку использования нулевой подсети. В сети компании ZIP имеется достаточный объем пространства IP-адресов, и поэтому первая подсеть из выделенного адресного пространства 131.108.0.0/25 не использовалась.

Новые протоколы маршрутизации более эффективны в распознавании различий между основной сетью и нулевой подсетью. Поэтому маршрутизатор имеет конфигурационные команды, которые разрешают пользователю использовать нулевую подсеть. Если в будущем окажется необходимым применить в маршрутизаторе SF-1 нулевую подсеть, то перед командой ip address потребуется ввести команду глобального конфигурирования ОС IOS ip subnet-zero.

```
S F-1# configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-1{config)#ip subnet-zero
SF-1(config)#interface ethernet 1
SF-1(config-if)#ip address 131.108.0.1 255.255.255.128
SF-1(config-if)#^Z
```

Конфигурирование интерфейса глобальной сети

IP-адресация интерфейсов глобальных сетей во многом подобна адресации интерфейсов локальных сетей, но со следующими исключениями.

- Интерфейсы глобальных сетей с двухточечной маршрутизацией могут не нумероваться.
- Интерфейсы глобальных сетей с многоточечной маршрутизацией, например, Frame Relay, X.25, ISDN и ATM, требуют отображения канальных адресов на IP-адреса.

Адресация интерфейса глобальной сети с двухточечной маршрутизацией

Интерфейс глобальной сети с двухточечной маршрутизацией — это просто интерфейс, в

котором связываются два устройства — одно на каждом конце линии. Интерфейсы подобного типа обычно встречаются при использовании выделенных арендуемых каналов передачи данных или при соединении двух маршрутизаторов друг с другом кабелями или с помощью заменителей модема. Двухточечные соединения также могут эмулироваться в многоточечной среде, например, Frame Relay или ATM, путем применения подинтерфейсов. На противоположных концах линии всегда имеется только одно устройство, так что при размещении пакета на двухточечном интерфейсе не возникает вопроса относительно того, по какому адресу или какой станции он посылается. Благодаря подобному свойству такой тип интерфейса
(или подинтерфейса) не требует IP-адресов, как интерфейсы локальной сети или многоточечные интерфейсы глобальной сети. В большинстве случаев для управления сетью и облегчения устранения неисправностей администраторы сети предпочитают присваивать адрес своим двухточечным интерфейсам глобальной сети. Однако, если сетевое адресное пространство ограничено, ненумеруемые интерфейсы определенно являются плюсом.

Если двухточечному интерфейсу глобальной сети, например, PPP, HDLC или Frame Relay, присваивается IP-адрес, то делается это с помощью команды ip address (аналогично тому, как это имеет место при адресации интерфейса локальной сети). Как и для интерфейса локальной сети, команда ip address требует указания сетевой маски и фактического IP-адреса. Каждому отдельному двухточечному соединению глобальной сети (или двухточечному подинтерфейсу) следует назначать отдельный сетевой IP-адрес. Заметим, что для обозначения того, какой интерфейс глобальной сети адресуется, команде ip address предшествует основная команда конфигурирования ОС IOS interface. Ниже показан пример IP-адресации для двухточечного HDLC-интерфейса и двух двухточечных подинтерфейсов Frame Relay маршрутизатора Seoul-1 компании ZIP:

```
Seoul-l#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul-1(config)#interface serial 0.16 point-to-point
Seoul-1(config-if)#ip address 131.108.242.1 255.255.255.252
Seoul-1(config-if)#interface serial 0.17 point-to-point
Seoul-1(config-if)#ip address 131.108.242.5 255.255.255.252
Seoul-1(config-if)#interface serial 1
Seoul-1(config-if)#ip address 131.108.241.2 255.255.255.252
Seoul-1(config-if)#^Z
```

Хотя на текущий момент в сети компании ZIP нет ненумерованных интерфейсов, рассмотрим процесс конфигурирования, когда в маршрутизаторе Seoul-2 появится интерфейс глобальной сети. Ненумеруемый двухточечный IP-интерфейс глобальной сети конфигурируется с помощью интерфейсной субкоманды ОС IOS ip unnumbered. Для того чтобы протоколы IP-маршрутизации, работая через ненумерованный интерфейс, имели для использования фактический IP-адрес, эта команда требует указать параметр базового интерфейса. Базовый интерфейс может быть физическим интерфейсом, например Ethernet или Token Ring, или виртуальным интерфейсом, например интерфейсом петли возврата. Ненумерованными должны быть оба конца канала глобальной сети, т.е. не может быть, чтобы одному концу линии адрес был присвоен, а другой конец остался ненумерованным. Ниже приводится пример добавления ненумерованного интерфейса маршрутизатору Seoul-2 сети компании ZIP:

```
Seoul-2#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul-2(config)#interface serial 1
Seoul-2(config-if)#ip unnumbered loopback 0
Seoul-2(config-if)#^Z
```

Ненумерованные IP-интерфейсы обладают двумя недостатками. Во-первых, невозможно сформировать соединение виртуального терминала (например, по протоколу Telnet)

непосредственно с последовательным интерфейсом или воспользоваться протоколом SNMP для опроса маршрутизатора через последовательный интерфейс. (SNMP — это протокол управления; более подробно он обсуждается в главе 7, "Основы администрирования и управления".) Можно организовать соединение с IP-адресом локальной сети или виртуальным интерфейсом в устройстве, осуществляя управление путем отправки запросов по этому адресу. Во-вторых, если ненумерованный интерфейс имеет ссылку на интерфейс локальной сети, и этот интерфейс выводится из активного состояния или отказывает, то связь с устройством может оказаться невозможной. По этой причине рекомендуется, чтобы ненумерованные интерфейсы имели в качестве базового какой-либо виртуальный интерфейс, например, интерфейс петли возврата.

Адресация многоточечного интерфейса глобальной сети

Многоточечный интерфейс глобальной сети — это такой интерфейс, который позволяет связываться с несколькими устройствами через одно соединение со средой глобальной сети. Посылаемый в многоточечный интерфейс глобальной сети пакет не знает, для какой станции он предназначается, поэтому для осуществления IP-обмена данными такие интерфейсы должны иметь адреса. Более того, такие технологии многоточечных глобальных сетей, как X.25, ISDN, ATM и Frame Relay, реализуют в себе методологии канальной адресации, используемые для того, чтобы различать станции в глобальной сети. Поэтому должен присутствовать механизм отображения IP-адреса на канальный адрес, который во многом аналогичен тому, как в интерфейсах локальной сети IP-адреса отображаются на MAC-адреса. Большинство технологий многоточечных глобальных сетей не имеют динамического метода отображения IP-адреса на канальный адрес. Следовательно, для обеспечения правильной адресации в такого типа интерфейсах требуются дополнительные команды. Исключение составляет технология Frame Relay, обладающая методом динамического отображения, называемым протоколом обратного разрешения адресов (Inverse ARP).

Хотя в сети компании ZIP и присутствуют многоточечные интерфейсы Frame Relay, все они с помощью подинтерфейсов сконфигурированы под работу в качестве двухточечных соединений. В сети компании ZIP отсутствуют интерфейсы X.25, ISDN или ATM.

Как описывалось в главе 3, в технологии Frame Relay для того, чтобы различать виртуальные каналы в сети Frame Relay, используются DLCI-идентификаторы. На многоточечном интерфейсе Frame Relay замыкается несколько вирту-1льных каналов, поэтому с ним ассоциируется также и несколько DLCI-идентификаторов. Для осуществления обмена данными между IP-устройствами, стоящими на концах этих виртуальных каналов, их IP-адреса должны отображаться на LCI-идентификаторы. Такое отображение позволяет многоточечному устройству идентифицировать в сети Frame Relay нужный виртуальный канал назначения каждого пакета, посылаемого через один физический интерфейс. После этого пакеты могут перемещаться по сети Frame Relay. На многоточечном интерфейсе Frame Relay отображение можно делать вручную с помощью субкоманды конфигурирования интерфейса ОС IOS frame-relay map или можно положиться на функцию протокола обратного разрешения адреса Inverse ARP. При адресации многоточечных интерфейсов глобальной сети необходимо прописать все устройства одной многоточечной сети, которые адресуются из одного логического сетевого IP-номера. Ниже приведен пример конфигурирования многоточечного интерфейса Frame Relay маршрутизатора SF-Core-1 с использованием команды frame-relay map:

```
SF-Core-l#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-Core-1(config)#interface serial 1/1
SF-Core-1(config-if)#encapsulation frame-relay ietf
SF-Core-1(config-if}#no inverse-arp
SF-Core-1(config-if)#ip address 131.108.130.1 255.255.255.0
SF-Core-1(config-if)#frame-relay map ip 131.108.130.17 30 Cisco broadcast
SF-Core-1(config-if)#frame-relay map ip 131.108.130.20 50 broadcast
SF-Core-1(config-if)#frame-relay map ip 131.108.130.35 62 broadcast
```

```
SF-Core-1(config-if)#^Z
```

В этом примере функция динамического отображения Inverse ARP отключена субкомандой конфигурирования интерфейса ОС ЮС no inverse-arp. Три IP-адреса отображаются на три виртуальных канала и их соответствующие номера DLCI-идентификаторов. Дополнительно виртуальный канал DLCI 30 с IP-адресом 131.108.101.17 использует для инкапсуляции не метод по умолчанию, а разработанный компанией Cisco метод "четверок" (gang of four). (Методом инкапсуляции по умолчанию определен метод IETF, что сделано с помощью команды конфигурирования интерфейса ОС IOS encapsulation frame-relay ietf.) Ключевое слово broadcast в конце команды frame-relay map инструктирует маршрутизатор направлять широковещательные пакеты для этого интерфейса в данный конкретный виртуальный канал.

Опционные ключевые слова и команды

Как и большинство команд ОС IOS, команды отображения IP-адресации на канальный уровень имеют опционные ключевые слова, которые изменяют поведение виртуального канала или активируют/деактивируют специальные функции, выполняемые в этом виртуальном канале, например, уплотнение данных. В данной книге освещаются только самые распространенные ключевые слова. Полное объяснение всех ключевых слов и команд ОС IOS можно найти на компакт-диске Cisco Connection Documentation или в интерактивном варианте на Web-сервере www.cisco com.universd/home/home.

Если бы в приведенном выше примере выполнение функции динамического отображения IP-адресов на DLCI-номера было разрешено, то необходимости в командах frame-relay map не было бы. В этом случае интерфейс посылал бы запросы функции Inverse ARP в каждый виртуальный канал, идентифицированный сетью Frame Relay на этом интерфейсе в качестве активного. Эти запросы, в свою очередь, привели бы к тому, что находящиеся на противоположном конце устройства ответили бы информацией о своих IP-адресах в данном виртуальном канале и запрашиваемых DLCI-идентификаторах. Использование функции Inverse ARP свело бы пример до следующих размеров:

```
SF-Core-l#configure
Configuring  from terminal,   memory,   or network   [terminal]?
Enter  configuration  commands,   one per  line.   End with  CNTL/Z.
SF-Core-1(config)#interface serial 1/1
SF-Core-1(config-if)#encapsulation frame-relay ietf
SF-Core-1(config-if)#ip address  131.108.130.1  255.255.255.0
SF-Core-1(config-if)#^Z
```

Совет

Конфигурирование интерфейса Frame Relay требует некоторой осторожности. Полагаясь в отображении IP-адресов на DLCI-идентификаторы на функцию Inverse ARP, следует иметь в виду, что ошибки в конфигурировании могут привести к отображению неизвестно откуда взявшихся виртуальных каналов на неизвестные устройства. Также следует помнить, что одновременное применение на одном интерфейсе Frame Relay иинкапсуляции по методу IETF и с помощью разработанного компанией Cisco метода “четверок" требует использования команд frame-relay map.

Статическая адресация интерфейсов глобальных сетей X.25 выполняется во многом так же, как и статическая адресация интерфейсов Frame Relay, т.е. с помощью команды static map. IP-адреса интерфейса X.25 должны отображаться на адреса 121, которые используются для установления виртуальных каналов связи между ситемами, стоящими в сети X.25. Каждый виртуальный канал идентифицируется ад-сом X.121, используемым для установления соединения. Ниже показан пример конфигурирования нового интерфейса X.25 маршрутизатора

компании ZIP с именем -Core-1 путем применения субкоманды конфигурирования интерфейсов x25 map:

```
SF-Core-1# configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF -Core-1(config)#interface serial 1/2
SF -Core-1(config-if)#encapsulation x25
SF -Core-1(config-if)#x25 address 44598631
SF -Core-1(config-if)#ip address 131.108.102.1 255.255.255.0
SF -Core-1(config-if)#x25 map ip 131.108.102.15 44593389 broadcast
SF -Core-1(config-if}#x25 map ip 131.108.102.29 44591165 broadcast
SF -Core-1(config-if)#tx25 map ip 131.108.102.176 44590712 broadcast
SF -Core-1(config-if)#^Z
```

Адресация интерфейсов ISDN требует команд отображения, подобных тем, которое используются при адресации интерфейсов Frame Relay и X.25. Однако команды отображения нужны только в тех случаях, когда устройство хочет установить соединение по коммутируемой линии с другим устройством. Если устройство принимает только входные звонки, то IP-адреса могут отображаться на принимающее устройство телефонный номер динамически. Для обеспечения отображения между IP-адресами именами систем и телефонными номерами, которые используются для установления соединений в сети ISDN, применяется субкоманда конфигурирования интерфейса ОС IOS dialer map. Чтобы установить правильное отношение между IP-адресом и телефонным номером удаленной системы, в команде dialer map должно вводиться ключевое слово name. Кроме того, ключевое слово name используется в качестве части процесса аутентификации после установления соединения с удаленной системой. Ниже показан пример конфигурирования нового интерфейса ISDN BRI (Basic Rate Interface — интерфейс передачи данных с номинальной скоростью) на маршрутизатора компании ZIP с именем Seoul-1:

```
Seoul-l#configure
Configuring from terminal, memory, or network  [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul -1(config)#interface bri 0
Seoul -1(config-if)#ip address 131.108.103.3 255.255.255.0
Seoul-1(config-if)#dialer map ip 131.108.103.1 name SF-Core-1 broadcast 14085551212
Seoul-1(config-if)#dialer map ip 131.108.103.2 name SF-Core-2 broadcast 14085551313
Seoul-1(config-if)#^Z
```

Адресация интерфейсов ATM, как и всех уже рассмотренных нами интерфейсов, требует ввода основной команды ip address. Однако для интерфейсов ATM тип команд, используемых для отображения IP-адресов на канальный уровень, зависит от типа применяемых ATM-протоколов, а также типа используемых виртуальных каналов. Существуют следующие возможные типы протоколов.

- **Инкапсуляция по протоколу управления логическим каналом/доступа к подсетям в постоянных виртуальных каналах (Logical link control/Subnetwork Access Protocol** (LLC/SNAP) **encapsulation with PVC's).** В этой модели в сети ATM устанавливается постоянный виртуальный канал. Пакеты идентифицируются как направленные на IP-адрес на противоположном конце конкретного виртуального канала.
- **Инкапсуляция по протоколу LLC/SNAP в коммутируемых виртуальных каналах. В** этой модели IP-пакеты идентифицируются как направленные на конкретный статически заданный ATM-адрес канального уровня. ATM-коммутаторы устанавливают виртуальный канал по требованию, когда маршрутизатор запрашивает соединение с ATM-адресом для конкретного IP-адреса.
- **Протокол IP с автоматическим разрешением адресов ARP.** В этой модели ATM-адрес канального уровня для конкретного IP-адреса автоматически поставляется станцией, называемой ATM ARP-сервером.

В случае применения инкапсуляции по протоколу LLC/SNAP в постоянных виртуальных

каналах для отображения IP-адресов на конкретные постоянные виртуальные каналы используется субкоманда конфигурирования интерфейса ОС IOS map-group и команда глобального конфигурирования map-list. В примере, показанном ниже, на маршрутизаторе SF-Core-1 выполняется конфигурирование адресации для нового ATM-интерфейса с инкапсуляцией по протоколу LLC/SNAP в постоянных виртуальных каналах:

```
SF-Core-l#configure
Configuring from terminal,  memory,   or network     [terminal]?
Enter configuration commands,   one per  line.   End with CNTL/Z.
SF-Core-1(config)#interface atm 1/0
SF-Core-1(config-if)#atm pvc 3 0 21 aalSsnap
SF-Core-1(config-if)#atm pvc 5 0 22 aalSsnap
SF-Core-1(config-if)#ip address  131.108.104.1  255.255.255.0
SF-Core-1 (config-if)#map-group zipl
SF-Core-1(config-if)#map-list zipl
SF-Core-1(config-map-list)#ip 131.108.104.2  atm-vc  3 broadcast
SF-Core-1(config-map-list)tip 131.108.104.7  atm-vc 5 broadcast
SF-Core-1(config-map-list)#^Z
```

В случае применения инкапсуляции по протоколу LLC/SNAP в коммутируемых виртуальных каналах для отображения IP-адресов на адреса точек доступа к сетевым Услугам (network service access point — NSAP), которые применяются для идентификации удаленных устройств в ATM-сети, используется субкоманда конфигурирования Интерфейса ОС IOS map-group и команда глобального конфигурирования map-list.

Ниже в примере показано, как осуществляется конфигурирование адресации для нового ATM-интерфейса с инкапсуляцией по протоколу LLC/SNAP в коммутируемых виртуальных каналах на маршрутизаторе SF-Core-1:

```
SF-Core-l#configure
Configuring from terminal, memory, or network  [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-Core-1(config)tinterface atm 1/0
SF-Core-1(config-if)ttatm nsap
FE.DCBA.01.987654.3210.ABCD.EF12.3456.7890.1234.12
SF-Core-1(config-if)#ip address 131.108.104.1 255.255.255.0
SF-Core-1(config-if)#map-group zipl
SF-Core-1(config-if)#map-list zipl
SF-Core-1(config-map-list)#ip 131.108.104.2 atm-nsap
Al.9876.AB.123456.7890.FEDC.BA.1234.5678.ABCD.12
SF-Core-1(config-map-list)#ip 131.108.104.7 atm-nsap
B2.9876.AB.123456.7890.FEDC.BA. 1234.5678.AB12.12
SF-Core-1(config-map-list)#^Z
```

Классический протокол IP с функцией ARP для конфигурирования IP-адресов интерфейса требует использования субкоманды ip address. Субкоманда конфигурирования ATM-интерфейса идентифицирует адрес ATM ARP-сервера, который способен преобразовать IP-адреса в ATM NSAP-адреса, которые необходимы для установления виртуальных каналов. Ниже приводится пример конфигурирования адресации нового ATM-интерфейса с классическим протоколом IP и функцией ARP в маршрутизаторе с именем SF-Core-1:

```
SF-Core-l# configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-Core-1(config)#interface atm 1/0
SF-Core-1(config-if)#atm nsap
FE.DCBA.01.987654.3210.ABCD.EF12.3456.7890.1234.12
SF-Core-1(config-if)#ip address 131.108.104.1 255.255.255.0
SF-Core-1(config-if)#atm arp-server nsap
01.ABCD.22.030000.0000.0000.0000.0000.0000.0000.00
SF-Core-1(config-if)#^Z
```

Проверка конфигурации IP-адресов

Верификация IP-адресов и других IP-атрибутов, которые были назначены интерфейсам, может быть выполнена с помощью одной из трех команд режима EXEC. Команда ОС IOS режима EXEC show interface выводит общую информацию об интерфейсе, включая назначенные ему IP-адрес и сетевую маску. Если в качестве параметра этой команды вводится конкретный интерфейс, то выводится информация только об этом интерфейсе. Если не указывается ни один интерфейс, то показываются данные обо всех интерфейсах. Ниже показан пример информации, выводимой в результате исполнения команды the show interface ethernet 0 на маршрутизаторе компании ZIP с именем SF-2:

```
SF-2#show interface ethernet 0
   EthernetO is up, line protocol is up
   Hardware is Lance, address is 0000.Oc07.b627 (bia 0000.Oc07.b627)
   Internet address is 131.108.110.1 255.255.255.0
   MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,rely 255/255,load 1/255
   Encapsulation ARPA,loopback not set,keepalive set (10 sec)
   ARP type:ARPA,ARP Timeout 04:00:00
   Last input 00:00:00,output 00:00:00,output hang never
   Last clearing of "show interface " counters never
   Queuing 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 1000 bits/sec, 1 packets/sec
      716895 packets input, 69741733 bytes, 0 no buffer
      Received 76561 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
      5148972 packets output, 750393298 bytes, 0 underruns
      0 output errors, 68 collisions, 5 interface resets
      0 babbles, 0 late collision, 286 deferred
      0 lost carrier, 0 no carrier
      0 output buffer failures, 0 output buffers swapped out
```

Команда ОС IOS режима EXEC show ip interface обеспечивает возможность полного просмотра всех параметров, связанных с IP-конфигурацией интерфейса. Если в качестве параметра команды указывается конкретный интерфейс, то на экран выводится информация только об этом интерфейсе. Если конкретный интерфейс не указывается, то выдается информация обо всех интерфейсах. Ниже показан результат исполнения команды show ip interface ethernet 0 на маршрутизаторе SF-2 компании ZIP:

```
SF-2#show ip interface ethernet 0
EthernetO is up, line protocol is up
Internet address is 131.108.110.1 255.255.255.0
Broadcast address is 255.255.255.255
Address determined by non-volatile memory
MTU is 1500 bytes
Helper address is not set
Directed broadcast forwarding is enabled
Multicast reserved groups joined: 224.0.0.1 224.0.0.2 224.0.0.10
Outgoing access list is not set
Inbound access list is not set
Proxy ARP is enabled
Security level is default
Split horizon is enabled
ICMP redirects are always sent
ICMP unreachables are always sent
ICMP mask replies are never sent
IP fast switching is enabled
IP fast switching on the same interface is disabled
IP multicast fast switching is enabled
Router Discovery is disabled
```

```
IP output packet accounting is disabled
IP access violation accounting is disabled
TCP/IP header compression is disabled
Probe proxy name replies are disabled
Gateway Discovery is disabled
Policy routing is disabled
Network address translation is disabled
```

Команда show ip interface имеет вариант формы, которая позволяет увидеть краткую сводную информации об IP-адресах и статусе всех имеющихся на устройстве интерфейсов. Такой краткий вариант отчета получается при использовании команды show ip interface brief.

Ниже показан результат исполнения команды show ip interface brief на маршрутизаторе SF-2 компании ZIP:

```
SF-2#show ip  interface brief
Interface          IP-Address          OK?Method       Status       Protocol
EthernetO         131.108.110.1        YES  NVRAM       up           up
Ethernetl         131.108.120          YES NVRAM        up           up
FastEthernet      0131.108.20.2        YES  NVRAM       up           up
```

В дополнение к проверке IP-конфигурации самого интерфейса имеется возможность просмотра как статических, так и динамических отображений IP-адресов на адреса канального уровня для различных сред многоточечных глобальных сетей. Для этого следует воспользоваться командами ОС IOS режима EXEC show frame-relay map, show atm map, show x25 map И show dialer maps. Ниже приведен пример информации, выводимой в результате исполнения команды show frame-relay map на маршрутизаторе компании ZIP Seoul-1:

```
Seoul-l#show frame-relay map
SerialO.16 (up): point-to-point dlci, dlci 16(0x10,0x400), broadcast,
status
defined, active
SerialO.17 (up): point-to-point dlci, dlci 17(0x11,0x410), broadcast,
status
defined, active
Seoul-l#
```

Другие команды отображения протоколов глобальных сетей выводят на экран информацию в виде, похожем на тот, в котором она выводится командой show frame-relay map.

Как обсуждалось ранее в разделе "TCP/IP - адресация", сетевые маски могут представляться как в десятичном формате с разделением точками, так и в формате с контрольной суммой или с прямой косой линией. По умолчанию маршрутизатор использует формат с контрольной суммой. Если при выполнении верификации сетевых масок вам удобнее пользоваться десятичным форматом с разделением точками, то для переключения форматов можно воспользоваться командой ОС IOS режима EXEC terminal ip netmask format decimal. Эта команда действует только в течение текущего сеанса виртуального терминала или консоли. Ниже показан пример использования команды в маршрутизаторе компании ZIP Seoul-1:

```
Seoul-lttterminal ip netmask-format decimal
Seoul-It
```

Чтобы сохранить этот формат для всех сеансов виртуального терминала или консоли, следует в режиме конфигурирования воспользоваться для нужных каналов субкомандой конфигурирования канала ip netmask-format decimal. Ниже показан пример изменения формата сетевой маски для всех сеансов виртуального терминала маршрутизатора компании ZIP Seoul-1:

```
Seoul-1#configure
Configuring from terminal, memory, or network [terminal] ?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul-1(config)#line vty 0 4
Seoul-1(config-line)#ip netmask-format decimal
Seoul-1(config-line)#^Z
```

Конфигурирование IP-маршрутизации

Назначение каждому устройству уникального IP-адреса является необходимым, но не достаточным условием для того, чтобы они могли обмениваться информацией друг с другом. Устройства в IP-сети, чтобы посылать друг другу пакеты данных, должны также знать путь или маршрут к другим устройствам, будь то в пределах одной автономной сети или во всей глобальной сети Internet. Однако, вместо того, чтобы каждое находящееся в сети устройство имело полный список всех других устройств сети с указанием их местонахождения, свою роль регулировщика потоков данных играет маршрутизатор, который выполняет в IP-сети две функции.

Первая функция состоит в том, что маршрутизатор принимает от станции пакеты, определяет оптимальный путь до пункта назначения и затем передает их в следующий сегмент локальной или глобальной сети, ведущий к пункту назначения. По мере продвижения пакета от одного маршрутизатора к другому в рамках сложной внутренней сети предприятия или в самой глобальной сети Internet этот процесс может повторяться несколько раз. Подобный процесс называется маршрутизацией, или коммутацией пакетов. Вторая функция маршрутизатора заключается в том, что он должен уметь определять, где находится другая IP-сеть или подсети (как внутри одной автономной сети, так и вне ее, например, в сети Internet). Чтобы определять местонахождение других сетей, маршрутизаторы используют таблицу маршрутизации, которая создается алгоритмами или протоколами маршрутизации.

По своей природе протоколы маршрутизации могут быть либо статическими, либо динамическими. В случае применения статических протоколов администратор сети вручную заносит в таблицу маршрутизации всю информацию о путях сетей. Статические протоколы подтверждены ошибкам, поскольку не способны реагировать на изменения в сети и должны реконфигурироваться вручную при каждом изменении. Динамические же протоколы маршрутизации полагаются в оповещении о подключенных к ним сетях и подсетях на сами маршрутизаторы. В разделе "Конфигурирование протоколов IP-маршрутизации" настоящей главы исследуются многочисленные протоколы динамической маршрутизации. В следующем разделе рассматриваются общие аспекты конфигурирования IP- и статической маршрутизации.

Команды конфигурирования IP-маршрутизации

Чтобы разрешить IP-маршрутизацию, используется команда глобального конфигурирования ОС IOS ip routing. По умолчанию в автономных маршрутизаторах ОС IOS сконфигурирована так, что IP-маршрутизация разрешена. Однако, если на таком устройстве функция IP-маршрутизации была отключена, до коммутации пакетов и активации алгоритмов маршрутизации ее выполнение следует снова разрешить. Некоторые устройства маршрутизации с интегрированным в них программным обеспечением компании Cisco не разрешают IP-маршрутизацию по умолчанию. В этом случае, чтобы в таких устройствах была возможной коммутация пакетов и обработка алгоритмов маршрутизации, снова необходимо использовать команду ip routing. В примере ниже показано, как разрешается IP-маршрутизация в маршрутизаторе сети компании ZIP с именем Seoul-1:

```
Seoul-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul-!(config)#ip routing
Seoul-1(config-line)#^Z
```

После разрешения IP-маршрутизации может строиться таблица маршрутизации, используемая для коммутации пакетов. По умолчанию после конфигурирования IP-адреса для интерфейса и его перевода в рабочее состояние сетевой адрес этого интерфейса заносится в таблицу маршрутизации. В таблицу маршрутизации заносятся данные всех интерфейсов, которые находятся в рабочем состоянии и подключены к маршрутизатору. Поэтому, если в сети есть только один маршрутизатор, то он обладает информацией обо всех других сетях или подсетях, и в конфигурировании статической или динамической маршрутизации нет необходимости. Записи в таблицу о статической или динамической маршрутизации необходимы только тогда, когда в сети находятся несколько маршрутизаторов.

Для просмотра таблицы IP-маршрутизации используется команда режима EXEC show ip route. При ее вводе без указания параметров на экран выводится вся информация, содержащаяся в таблице маршрутизации. Ниже показан пример таблицы маршрутизации маршрутизатора Seoul-1 сети компании ZIP, содержащей записи только о подключенных и находящихся в активном состоянии интерфейсах без каких-либо дополнительных записей, образовавшихся в результате конфигурирования или обучения:

```
Seoul-l#show ip route
         Codes:   C  -connected,   S  -static,   I  -IGRP,   R -RIP,   M -mobile,
         B  -BGP D -EIGRP,   EX -EIGRP external,   0 -OSPF,   IA -OSPF inter area
         N1   -OSPF NSSA external  type  1,   N2   -OSPF NSSA external  type  2 El
         -OSPF external  type  1,   E2  -OSPF external type 2,   E -EGP i -IS-IS,
         LI -IS-IS level-1,  L2 -IS-IS level-2,   * - candidate default U - per-
         user static  route,   o -ODR
Gateway of last  resort  is not  set
131.108.0.0/16 is variably subnetted,   4  subnets,   2 masks
C         131.108.3.0/25 is directly connected,   Ethernet0
C         131.108.242.0/30  is  directly connected,   Serial0.16
C         131.108.242.4/30   is  directly connected,   Serial0.17
C         131.108.241.0/30   is  directly  connected,   Seriall
```

Команда show ip route дает администратору сети огромное количество данных. Она является ключевым инструментом, используемым для определения пути, по которому пакет проходит по сети. Первый раздел выводимой информации представляет собой комментарий к первому столбцу самой таблицы. В нем говорится о том, где был взят маршрут. В данном примере буква С свидетельствует о том, что маршрут взят из непосредственно подключенного и функционирующего интерфейса. Шлюз последней надежды — это сетевой адрес маршрутизатора, куда должны будут посылаться пакеты, имеющие пункт назначения вне данной сети, если отсутствует конкретная информация о маршрутизации до этого пункта назначения. В приведенном примере маршрутизатор не был нацелен на шлюз последней надежды, так как не было сконфигурировано никаких статических маршрутов и никакие протоколы динамической маршрутизации не исполняются.

Последний раздел выводимой информации представляет собой собственно таблицу маршрутизации. По замечанию о переменной разбивке на подсети видно, что принадлежащая компании ZIP сеть класса В 131.108.0.0 была сконфигурирована с несколькими сетевыми масками. Выводимая информация также показывает, что данный маршрутизатор был нацелен на четыре подсетевых маршрута, которые имеют только две связанные с ними сетевые маски. Перечисляются все номера сетей, связанные с введенными на соответствующие интерфейсы IP-адресами, а также сетевые маски в формате с контрольной суммой и соответствующие названия интерфейсов. Важно отметить, что перечисляемые в таблице маршрутизации адреса сетей и подсетей не являются IP-адресами отдельных устройств. Сетевой адрес может представлять маршрут всего до двух хост-машин (как при использовании сети с сетевой маской /30) или до 65536 машин (при использовании сети, которая имеет сетевую маску /16), или даже до большего количества хост-машин, что зависит от сетевой маски.

Команда show ip route также имеет параметры по выбору, которые могут использоваться для осуществления запросов о маршрутах только определенного типа. Например, если таблица маршрутизации полностью заполнена маршрутами по прямым подключениям, статическими

маршрутами и маршрутами, полученными в результате динамического обучения, то команда ОС IOS режима EXEC show ip route connected может быть использована для показа только тех маршрутов, которые были определены из непосредственно подключенных и активных интерфейсов. Аналогично, команда show ip route static выводит только те маршруты, которые были получены из команд ручного конфигурирования сетевых путей. При вводе в качестве параметра команды конкретного сетевого адреса выводится информация только относительно этого конкретного маршрута. Ниже приводится пример ввода на маршрутизаторе Seoul-1 сети компании ZIP параметра конкретного маршрута с помощью команды show ip route 131.108.3.0:

```
Seoul-l#show ip route 131.108.3.0
Routing entry for 131.108.3.0/25
     Known via  "connected ",   distance 0,   metric  0   (connected) Routing
Descriptor Blocks:
     *  directly connected,   via EthernetO
Route metric is  0,   traffic  share count  is  1
```

Другие параметры по выбору команды show ip route исследуются в разделе "Конфигурирование протоколов IP-маршрутизации". Там же приводится и пояснение термина "метрика маршрута".

Конфигурирование статической маршрутизации

Как отмечалось ранее, для построения таблицы маршрутизации и, следовательно, получения информации о сетевых путях может использоваться информация как статической, так и динамической маршрутизации. Исторически статические маршруты были первым имеющимся в распоряжении сетевых администраторов средством для построения таблиц сетевых путей маршрутизаторов и некоторых оконечных устройств. Статические маршруты обладают определенными недостатками. Например, они не адаптируются к случаям, когда канал данных выходит из строя или изменяется топология сети. Однако все еще существует множество ситуаций, в которых статические маршруты нужны и желательны.

- Канал особенно ненадежен и постоянно разрывает передачу данных. В таких обстоятельствах протокол динамической маршрутизации может внести слишком много нестабильности, тогда как статический маршрут не изменяется.
- Сеть доступна по соединению через коммутируемую линию связи. Такая сеть не способна обеспечить постоянные обновления информации о себе, которые требуются протоколом динамической маршрутизации.
- Существует одно соединение с одним Internet-провайдером. В этом случае спользуется один-единственный статический маршрут по умолчанию. То же самое относится и к корпоративному офису, имеющему одно подключение к внутренней сети корпорации.
- Клиент или другая подсоединенная сеть не хотят обмениваться информацией динамической маршрутизации. Чтобы обеспечить наличие данных о том, каким бразом возможно выйти на такую сеть, может использоваться статический маршрут.

Конфигурирование статических маршрутов выполняется с помощью команды глобального конфигурирования ip route. Эта команда имеет несколько параметров, включая адрес сети и соответствующую сетевую маску, а также информацию о том, куда маршрутизатор должен посылать пакеты, предназначенные для этой сети. Информация о пункте назначения может иметь одну из нескольких форм.

- Конкретный IP-адрес следующего маршрутизатора в пути.
- Адрес сети другого маршрута из таблицы маршрутизации, куда должны переадресовываться пакеты.
- Непосредственно подключенный интерфейс, на котором размещена сеть на начения.

Первый вариант достаточно очевиден и представляет собой доминирующий способ, с помощью которого осуществляется ввод статических маршрутов. Ниже показан пример ввода статического маршрута для маршрутизатора компании ZIP SF-Core-1. В соответствии с этим маршрутом пакеты, адресованные на сетевой адрес 131.108.230.0/24, направляются по

последовательному соединению в маршрутизатор в Сан-Хосе, который имеет адрес 131.108.240.2:

```
SF-Core-l#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-Core-l(config)#ip route 131.108.230.0 255.255.255.0 131.108.240.2
SF-Core-1(config)#^Z
```

Второй вариант, при котором в качестве пункта назначения задается маршрут к другой сети, используется, если существует несколько путей выхода на нужный сетевой адрес. Преимуществом такого варианта является разделение трафиковой нагрузки между несколькими равнозначными по стоимости путями. Вторым преимуществом является то, что отказ одного из путей приводит к перенаправлению трафика на один из альтернативных путей. Ниже показан пример для маршрутизатора компании ZIP SF-Core-1. Пакеты, направленные на сетевой адрес 131.108.231.0/24, используют маршрут, ведущий к сети в Сан-Хосе 131.108.100.0/24:

```
SF-Core-I#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-Core-1(config)lip route 131.108.231.0 255.255.255.0 131.108.100.0
SF-Core-1(config) #^Z
```

Заметим, что для того, чтобы пакетам добраться до сети 131.108.231.0/24, в таблице маршрутизации должен присутствовать маршрут до сети 131.108.100.0/24. Пакеты для сети 131.108.231.0/24 посылаются через тот же интерфейс, через который отправляются и пакеты, адресованные сети 131.108.100.0/24.

Последний вариант задания адреса получателя — непосредственно подключенный интерфейс — является, вероятно, наименее используемым. Задавая в качестве пункта назначения непосредственно подключенный интерфейс, сетевой администратор по сути определяет, что устройства с IP-адресами из этой сети размещаются на заданном интерфейсе. В результате, IP-адреса пакетов из этой сети должны преобразовываться в адреса канального уровня для интерфейса конкретного типа (в таблице маршрутизации маршрут будет показан как маршрут через прямое подключение). При использовании интерфейса типа Ethernet IP-адрес должен будет преобразовываться в MAC-адрес. Для интерфейса типа Frame Relay либо должна существовать статическая карта преобразования для интерфейса Frame Relay, либо должна активироваться функция обратного разрешения адресов Inverse ARP, которые поставляют DLCI-информацию для отображения IP-адресов. При работе с интерфейсом ISDN необходима карта преобразования в вызов по номеру, которая отображает IP-адрес на имя системы и телефонный номер.

Совет

Задание для статического маршрута интерфейса в качестве пункта назначения является одной из основных ошибок, совершаемых при использовании команды ip route. Некоторые сетевые администраторы ошибочно полагают, что, если указать маршрут на конкретный интерфейс, пакеты будут должным образом перенаправляться следующему стоящему в пути маршрутизатору. Пакеты перенаправляются маршрутизатору в следующем узле перехода только тогда, когда указан IP-адрес этого маршрутизатора или задан маршрут к другой сети, проходящий через маршрутизатор следующего узла перехода.

Ниже показан пример задания непосредственно подключенного интерфейса в качестве пункта назначения в команде ip route. В этом примере сетевой адрес 131.108.232.0/24 конфигурируется как такой, на который можно выйти прямо через интерфейс Fast Ethernet маршрутизатора компании ZIP SF-Core-1:

```
SF-Core-l#configure
Configuring from terminal,   memory,   or network   [terminal]?
Enter configuration commands,   one per line.   End with CNTL/Z.
SF-Core-1(config)#ip route 131.108.232.0 255.255.255.0 fastethernet 0/0
SF-Core-1(config)#^Z
```

Давайте проверим таблицу маршрутизации маршрутизатора SF-Core-1 на предмет непосредственно подключенных интерфейсов и новых записей статических маршрутов:

```
SF-Core-l#show ip route
        Codes:   C -connected,   S -static,   I -IGRP,   R -RIP,   M -mobile,   B
        -BGP D  -EIGRP,   EX -EIGRP external,   0 -OSPF,   IA -OSPF  inter  area
        N1  -OSPF NSSA external  type  1,   N2   -OSPF NSSA external  type  2 El
        -OSPF external  type  1,   E2  -OSPF external  type  2,   E  -EGP i -IS-
        IS,  LI -IS-IS level-1,  L2 -IS-IS level-2,   * - candidate default U -
        per-user static route,   o -ODR
Gateway of last resort  is not set
131.108.0.0/16 is variably subnetted,   5  subnets,   3 masks
C     131.108.20.0/22   is  directly connected,   FastEthernetO/0
C     131.108.240.0/30  is directly connected,   Seriall/0  , ,,,
S     131.108.230.0/24  [1/0]   via  131.108.240.2
S     131.108.231.0/24  [1/0]   via  131.108.100.0
S     131.108.232.0/24  [1/0]   is  directly connected,   FastEthernetO/0
```

В момент прохождения пакетов через маршрутизатор и при поиске в таблице маршрутизации сетей назначения поведение маршрутизатора по умолчанию состоит в том, чтобы в рамках класса сети IP-адреса назначения подобрать наиболее близкую пару из конкретно заданного сетевого адреса и сетевой маски. Например, если пакет направлен на IP-адрес 131.108.21.6, и есть маршрут до сети 131.108.20.0/22, то пакет будет пересылаться через интерфейс для этого маршрута. Если для этого же пункта назначения имеются маршруты к сетям 131.108.20.0/22 и 131.108.21.0/24, то пакет будет пересылаться через интерфейс маршрута к адресу 131.108.21.0/24, так как это более точно заданный маршрут (с более длинной сетевой маской), чем маршрут до сети 131.108.20.0/22. В сети компании ZIP наименее конкретным маршрутом является маршрут 131.108.0.0/16, а наиболее конкретно заданными маршрутами являются сетевые адреса с маской /30.

Конфигурирование бесклассовой маршрутизации

В соответствии с традиционной классовой системой адресации сети компании ZIP была назначена сеть класса B. Однако, если был введен CIDR-блок (блок бесклассовой междоменной маршрутизации) адресов, то, чтобы дать маршрутизатору возможность подбирать в таблице маршрутизации маршруты, которые выходят за границы классов, потребуются дополнительные команды. Предположим, что сети был назначен CIDR-блок 206.220.224.0/24 (который состоит из адресов класса C с 206.220.224.0/24 по 206.220.227.0/24), и она была разбита на подсети и распределена по интерфейсам маршрутизатора следующим образом:

- интерфейсу Ethernet 0 назначен адрес 206.220.224.0/24;
- интерфейсу Ethernet 1 назначен адрес 206.220.225.0/24;
- интерфейсу Ethernet 2 назначен адрес 206.220.226.0/23.

По умолчанию маршрутизатор работает в классовом режиме. Пакеты, направленные на адрес 206.220.224.5, должным образом маршрутизируются в интерфейс Ethernet 0, поскольку сетевой адрес является адресом класса C и согласуется с IP-адресом пункта назначения. Это же справедливо для интерфейса Ethernet 1 и пакетов с адресом получателя 206.220.225.9. Однако пакеты, направленные на адреса 206.220.226.8 или 206.220.227.12, не согласуются с маршрутом для интерфейса Ethernet 2 206.220.226.0/23, и данные адреса пунктов назначения оказываются недосягаемыми. Это происходит из-за того, что сетевой адрес интерфейса Ethernet 2 представляет собой CIDR-блок из двух адресов класса C. Для того чтобы маршрутизатор работал в бесклассовом режиме и совмещал IP-адреса назначения с этим сетевым CIDR-адресом, сначала должна быть введена команда глобального конфигурирования ОС IOS ip classless. Ниже показан пример

ввода команды ip classless на маршрутизаторе компании ZIP SF-Core-1:

```
SF-Core-1#configure
Configuring from terminal, memory, or network  [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-Core-1(config)#ip classless SF-Core-1(config)#^Z
```

Конфигурирование сводных маршрутов и маршрутов по умолчанию

Во многих ситуациях не рекомендуется вести полную таблицу маршрутизации внутренней корпоративной сети предприятия или глобальной сети Internet. Это справедливо для небольшого офиса, имеющего одно подключение к корпоративной внутренней сети предприятия, маршрутизатора, конфигурируемого в условиях недостаточности памяти или наличия одного подключения к одному провайдеру Internet-сервиса. В подобных ситуациях конечным пользователям необходимо связываться с определенными адресами назначения, которые не имеют прямых записей в таблице маршрутизации. В нормальных условиях подобные пакеты были бы отброшены как имеющие недостижимые пункты назначения. Однако благодаря использованию сводных маршрутов и маршрутов по умолчанию маршрутизатор получает информацию о выходе на такие адреса назначения. Как сводный маршрут, так и маршрут по умолчанию предоставляют информацию об альтернативном пути, когда отсутствует маршрут, согласующийся с IP-адресом назначения.

Сводный маршрут дает информацию о пути выхода на пункт назначения по умолчанию внутри заданного адресного пространства. Как правило, сводный маршрут, который обычно укладывается в границы классов сети, используется для предоставления информации по умолчанию о пути выхода на подсети, явно не обнаруживаемые в таблице маршрутизации, но существующие в рамках внутренней корпоративной сети. В сети компании ZIP, например, маршрут 131.108.0.0/16 рассматривался бы как сводный. Если маршрутизатор в сети компании ZIP сталкивается с пакетом, имеющим адрес назначения 131.108.99.5, и не находит конкретного маршрута, к примеру, 131.108. S9.0/24, то он обычно отбрасывает этот пакет. Если в данной ситуации в таблице маршрутизации был бы сводный маршрут 131.108.0.0/16, то пакет переправился бы в интерфейс в направлении следующего перехода к пункту назначения по сводному маршруту.

Обычно сводный маршрут указывает направление к другой подсети в рамках внутренней корпоративной сети предприятия, но он также может указывать и конкретный IP-адрес следующего узла перехода. В любом случае целью сводного маршрута является направление пакетов друг им маршрутизаторам в рамках внутренней корпоративной сети предприятия, которые имеют более полную информацию маршрутизации. Сводный маршрут может конфигурироваться с помощью команд глобального конфигурирования ОС 1OS ip default-network или ip route.

При использовании команды ip default-network в качестве параметра указывается неподсоединенная подсеть, которая существует во внутренней корпоративной сети предприятия. Если же используется команда ip route, то в качестве параметров указываются сводный маршрут, сетевая маска и неподсоединенная подсеть. Ниже приведены примеры обеих команд, конфигурируемых на маршрутизаторе компании ZIP с именем Singapore. В этих примерах сводным маршрутом является маршрут 131.108.0.0/16, а неподсоединенной подсетью, используемой по умолчанию для Достижения пункта назначения, является сеть 131.108.20.0, которая обнаруживается на маршрутизаторах SF-Core-1 и SF-Core-2.

```
Singapore # configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Singapore(config)#ip default-network 131.108.20.0
Singapore(config)#^Z
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Singapore(config)#ip route 131.108.0.0 255.255.0.0 131.108.20.0
Singapore (config) #^Z
```

После того как сводный маршрут будет сконфигурирован, он появляется в таблице маршрутизации в виде наименее конкретно заданного сетевого маршрута с сетевой маской короче, чем у других сетевых и подсетевых маршрутов в таблице. В примере исполнения команды show ip route на маршрутизаторе компании ZIP Singapore маршрут 131.108.0.0/16 является сводным маршрутом, сконфигурированным в предыдущем примере. Заметим, что информация о пункте назначения сводного маршрута 131.108.20.0 была получена от маршрутизатора компании ZIP Seoul-1.

```
Singapore#show ip route
       Codes: C - connected, S - static, I - IGRP, R - RIP, M -mobile, B - BGP D -
       EIGRP, EX - EIGRP external, 0 - OSPF, IA -OSPF inter area N1 - OSPF NSSA
       external type 1, N2 - OSPF NSSA external type 2 El - OSPF external type 1,
       E2 - OSPF external type 2,E - EGP i - IS-IS, LI -IS-IS level-1, L2 -IS-IS
       level-2, * - candidate default U - per-user static route, o - ODR
Gateway of last resort is not set
131.108.0.0/16 is variably subnetted, 4 subnets, 4 masks
C   131.108.1.0/25 is directly connected, EthernetO
C   131.108.242.4/30 is directly connected, SerialO.100
D   131.108.20.0/22 [1/0] via 131.108.242.5, 20:10:45, SerialO.100
S   131.108.0.0/16 [1/0] via 131.108.20.0
```

Примечание

Если неподсоединенная подсеть находится внутри того же классового сетевого адресного пространства, что и непосредственно подключенный интерфейс маршрутизатора, то ОС IOS заменяет команду ip default-network командой ip route в версии со сводным маршрутом.

Когда IP-станция обменивается данными с другой компанией, университетом или организацией (через соединения по частной сети или через глобальную сеть Internet), она посылает пакеты, которые должны попасть на станции, размещенные в пространстве IP-адресов, отличающихся от ее собственного. Например, если станция сети компании ZIP общается с популярным Web-сервером www.yahoo.com, то пакеты, порождаемые в сетевом адресном пространстве сети компании ZIP 131.108.0.0/16, адресуются в сетевое адресное пространство компании Yahoo! 216.32.74.55/22. Чтобы переправлять должным образом пакеты, маршрутизаторы компании ZIP должны либо иметь точный маршрут 216.32.74.55/22, либо менее конкретный CIDR-маршрут, дающий им общее направление к сети компании Yahoo!.

Как уже объяснялось ранее, маловероятно, чтобы каждый маршрутизатор в сети компании ZIP или даже маршрутизатор подсоединения к глобальной сети Internet имел этот маршрут в своей таблице маршрутизации. Если сеть компании ZIP имеет одного-единственного провайдера Internet-услуг или не обменивается с ним информацией динамической маршрутизации, то, вероятнее всего, маршрутизаторы сети компании ZIP полагаются на сетевой маршрут по умолчанию, который обеспечивает необходимую информацию о выходе на Web-сервер компании Yahoo!, а также на другие серверы глобальной сети Internet (или, потенциально, на серверы внутри собственной внутрикорпоративной сети).

Базовая концепция маршрута по умолчанию заключается в том, что маршрутизатор, не имеющий конкретной информации о маршрутизации до пункта назначения, воспользуется путем по умолчанию до конкретной сети, содержащей маршрутизаторы с более полной информацией. Хотя маршрут по умолчанию подобен сводному маршруту, он используется для направления пакетов в IP-пункты назначения, которые находятся как вне автономной внутренней сети корпорации, так и вне границ адресов класса сети для данной организации. В глобальной сети Internet провайдер Internet-услуг компании или провайдер услуг выхода на центральный узел для самого провайдера Internet-услуг, вероятнее всего, обмениваются информацией динамической маршрутизации о местонахождении и достижимости всех сетей в рамках глобальной сети Internet с другими провайдерами Internet-услуг. Используя сетевой IP-адрес, принадлежащий сети провайдера Internet-услуг в качестве маршрута по умолчанию, маршрутизатор соединения с глобальной сетью Internet компании переправит пакеты, адресованные в неизвестные пункты назначения, провайдеру

Internet-услуг и, в конечном итоге, маршрутизаторам, которые имеют более полные таблицы маршрутизации и картины глобальной сети Internet.

Рассмотрим некоторые методы конфигурирования сетей по умолчанию с помощью ОС IOS.

- Конфигурирование сети по умолчанию с использованием динамически узнаваемого внешнего маршрута.
- Конфигурирование сети по умолчанию с использованием статически конфигурируемого внешнего маршрута.
- Конфигурирование сети по умолчанию с использованием зарезервированного адреса 0.0.0.0.

Основное различие между методами конфигурирования сети по умолчанию заключается в том, извлекается ли информация динамической маршрутизации из внешнего источника, например от провайдера Internet-услуг. Если маршруты к внешним сетевым адресам извлекаются из внешнего источника, то просто обозначьте один из этих внешних адресов как сеть по умолчанию, воспользовавшись командой глобального конфигурирования ОС IOS ip default-network. Параметром этой команды является маршрут со следующими характеристиками: он существует в таблице маршрутизации, не подключен к конфигурируемому маршрутизатору и попадает в адресное пространство класса сетей, сконфигурированное на каком-либо интерфейсе маршрутизатора. Ниже показан пример конфигурирования маршрутизатора компании ZIP SF-Core-1 с помощью команды ip default-network на сеть 140.222.0.0, информация о которой была взята у провайдера Internet-услуг компании ZIP:

```
SF-Core-1#configure
Configuring from terminal,   memory,   or network   [terminal]?
Enter configuration commands,   one per line.   End with CNTL/Z.
SF-Core-1 (config)#ip default-network 140.222.0.0
SF-Core-1 (config)#^Z
```

После конфигурирования маршрутизатор показывает, что он воспринял эту сеть в качестве сети по умолчанию, и на маршрут можно выйти, о чем говорит выводимая командой show ip route информация о шлюзе последней надежды. Возле маршрута маршрутизатор ставит звездочку, обозначая, что он *является* кандидатом на сеть по умолчанию, поскольку может быть сконфигурировано несколько поведений по умолчанию. Ниже показан пример с установленным для маршрутизатора компании ZIP SF-Core-I шлюзом последней надежды:

```
SF-Core-l#show ip route
          Codes:   C  -  connected,   S  -static,   I  -IGRP,   R -RIP,M -mobile,
          B –BGP D  -  EIGRP,   EX  -EIGRP external,   0 -OSPF,   IA -OSPF inter
          area     N1   -  OSPF NSSA external  type  1,   N2  -OSPF NSSA external
          type  2 El  - OSPF external type  1,   E2  -OSPF external  type  2,   E
          -EGP i  -IS-IS,   LI -IS-IS level-1,   L2  -IS-IS level-2,   * -
          candidate default U  -  per-user  static  route,   o  -ODR
Gateway of last  resort is  192.72.2.1 to network 140.222.0.0
131.108.0.0/16 is variably subnetted,   5 subnets,   3 masks
C    31.108.20.0/22  is directly connected,   FastEthernetO/0
C     131.108.240.0/30  is  directly connected,   Seriall/0
S     131.108.230.0/24   [1/0]   via  131.108.240.2
S     131.108.231.0/24   [1/0]   via  131.108.100.0
S    131.108.232.0/24   [1/0]   is directly connected, FastEthernetO/0
C     192.7.2.2/30  is  directly connected,   Seriall/1
B*   140.222.0.0/16   [20/19]   via  192.7.2.1,,3d08h
```

Если процесс динамической маршрутизации не ведет обмен с внешним провайдером, для указания на внешний сетевой адрес, используемый в качестве сети по умолчанию, применяется статический маршрут. Ниже показан предыдущий пример, но на этот раз для вывода информации о достижимости сетевого адреса по умолчанию через соединение с провайдером Internet-услуг используется статический маршрут.

```
SF-Core-l#configure
Configuring  from terminal,   memory,   or network   [terminal]?
Enter configuration commands,   one per line.   End with CNTL/Z.
```

```
SF-Core-1(config)#ip route 140.222.0.0 255.255.0.0  192.7.2.1
SF-Core-1(config)#ip default-network 140.222.0.0
SF-Core-1(config) #^Z
```

Как и в предыдущем примере, проверка данных, выводимых командой show ip route, показывает, что маршрутизатор установил сеть по умолчанию, но теперь отметил происхождение маршрута 140.222.0.0 буквой S (статический), поскольку он был сконфигурирован вручную.

```
SF-Core-l#show ip route
       Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP D -
       EIGRP, EX - EIGRP external, 0 - OSPF, IA -OSPF inter area N1 - OSPF NSSA
       external type 1, N2 - OSPF NSSA   external type 2 El - OSPF external type 1,
       E2 - OSPF external  type 2, E -EGP i -IS-IS, LI -IS-IS level-1, L2 -IS-IS
       level-2, * - candidate default U -per-user static route, o - ODR
Gateway of last resort is 192.72.2.1 to network 140.222.0.0
131.108.0.0/16 is variably subnetted, 5 subnets, 3 masks
C  131.108.20.0/22 is directly connected, FastEthernetO/0
C 131.108.240.0/30 is directly connected, Seriall/0
S  131.108.230.0/24 [1/0] via 131.108.240.2
S 131.108.231.0/24 [1/0] via 131.108.100.0
S 131.108.232.0/24 [1/0] is directly connected, FastEthernetO/0
C 192.7.2.2/30 is directly connected, Seriall/1
S* 140.222.0.0/16 [20/19] via 192.7.2.1
```

Последний метод конфигурирования сети по умолчанию должен быть знаком тем, кто работал в среде операционной системы UNIX (или ее производных) или в среде протокола маршрутной информации (Routing Information Protocol — RIP). Этот метод связан с установкой статического маршрута к специальному сетевому адресу — 0.0.0.0. Этот адрес считается зарезервированным. В среде UNIX или RIP он обозначает маршрут ко всем неизвестным IP-пунктам назначения.

В ОС IOS на маршрутизаторе сетевой адрес 0.0.0.0 является наименее конкретно определенным сетевым адресом. С подразумеваемой маской 0.0.0.0 или 0 бит этот маршрут отвечает любому IP-пункту назначения вне пределов адресного пространства класса. Если введена команда ip classless, маршрут подходит для любого IP-адреса неизвестного пункта назначения как внутри, так и вне адресного пространства класса. Ниже приведен пример использования команды ip route для конфигурирования в маршрутизаторе компании ZIP SF-Core-1 сети по умолчанию 0.0.0.0/0 с соединением с провайдером Internet-услуг в качестве IP-адреса следующего узла перехода.

```
SF-Core-1#configure
Configuring  from terminal,   memory,   or network   [terminal]?
Enter  configuration  commands,   one per  line.   End with CNTL/Z.
SF-Core-l(config)#ip route 0.0.0.0  0.0.0.0  192.7.2.1
SF-Core-1(config)#^Z
```

В выводимой командой show ip route информации указывается, что маршрутизатор установил маршрут сети по умолчанию 0.0.0.0. Как и в предыдущем методе, происхождение маршрута 0.0.0.0 показывается буквой S, так как он был сконфигурирован вручную.

```
SF-Core-1#show ip route
            Codes:   C  -connected,   S  -static,   I  -IGRP,   R -RIP,   M -
            mobile,   B -BGP D  -EIGRP,   EX  -EIGRP  external,   0  -OSPF,   IA -
            OSPF  inter area N1  -OSPF NSSA external  type  1,   N2   -OSPF NSSA
            external  type  2 El  -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
Gateway of last resort  is  192.72.2.1  to network 0.0.0.0
131.108.0.0/16 is variably subnetted,   5  subnets,   3 masks
C    131.108.20.0/22  is directly    connected,   FastEthernetO/0
C    131.108.240.0/30 is directly connected,   Seriall/0
S     131.108.230.0/24   [1/0]   via   131.108.240.2
S     131.108.231.0/24   [1/0]   via  131.108.100.0
```

```
S    131.108.232.0/24   [1/0]   is directly connected,   FastEthernetO/0
C    192.7.2.2/30 is directly connected,   Seriall/1
S*   0.0.0.0   [1/0]   via  192.7.2.1
```

Примечание

Если не была введена команда ip classless, и все маршруты до IP-пунктов назначения внутри и вне внутренней корпоративной сети неизвестны, то должны быть сконфигурированы как классовый сводный маршрут, так и маршрут сети по умолчанию. Это требование вытекает из предположения, что все маршрутизаторы внутри пространства класса IP-адресов обладают полнотой знаний о подсетях внутри этого адресного пространства. При работе в бесклассовом режиме по команде ip classless для обработки пунктов назначения, располагающихся как во внутренних подсетях, так и во внешних сетях, хватает одного маршрута по умолчанию к сети 0.0.0.0, поскольку он походит для всех неизвестных IP-пунктов назначения.

Конфигурируя маршрут сети по умолчанию для своей сети, выполняйте следующие важные рекомендации.

- При отсутствии обмена информацией динамической маршрутизации с внешней организацией, например, провайдером Internet-услуг, использование статического маршрута 0.0.0.0/0 является самым легким способом генерации маршрута по умолчанию.
- При обмене информацией динамической маршрутизации с одним или несколькими провайдерами Internet-услуг наиболее подходящим способом назначения одного или нескольких возможных маршрутов к сетям по умолчанию является использование команды ip default-network.
- При наличии одного или нескольких подсоединений к глобальной сети Internet через провайдера Internet-услуг маршрутизатор(ы) подсоединения к сети Internet должны распространять данные о сети по умолчанию во внутренней корпоративной сети предприятия с помощью протокола динамической маршрутизации.
- Приемлемо конфигурирование нескольких маршрутизаторов, стоящих во внутренней корпоративной сети предприятия, *с* помощью команды ip default-network. При этом маршрутом по умолчанию обозначается маршрут, получаемый в результате динамического обучения. Не допускается конфигурирование нескольких маршрутизаторов внутренней корпоративной сети на маршрут по умолчанию с адресом 0.0.0.0/0, если этот маршрутизатор не обеспечивает подсоединения к глобальной сети Internet через провайдера Internet-услуг. Поскольку это может привести к тому, что маршрутизаторы, не имеющие средств связи с неизвестными пунктами назначения, будут притягивать пакеты на себя. Это, в свою очередь, приведет к невозможности их доставки в такие пункты назначения. Исключение составляют те маршрутизаторы, которые не обмениваются информацией динамической маршрутизации или имеют только редкие соединения со внутренней корпоративной сетью в таких средах, как коммутируемые линии связи с протоколом ISDN или коммутируемые виртуальные каналы с протоколом Frame Relay.
- Как отмечалось ранее, в маршрутизаторах, которые не обмениваются информацией динамической маршрутизации или работают с соединениями по коммутируемым линиям связи, например, ISDN или коммутируемые виртуальные каналы Frame Relay, должен быть сконфигурирован маршрут сети по умолчанию либо маршрут на резервный адрес 0.0.0.0/0.
- Если внутренняя корпоративная сеть предприятия не подключена к каким-либо внешним сетям, например к глобальной сети Internet, то конфигурирование сети по умолчанию следует выполнять на маршрутизаторе или маршрутизаторах, которые стоят в центре сети и знают полную топологию маршрутов данной сети.

Совет

Если сеть по умолчанию конфигурируется с помощью статического маршрута с сетевым адресом 0.0.0.0/0, и маршрутизатор после команды ip classless работает в бесклассовом режиме, очень легко создать петлю маршрутизации между провайдером Internet-услуг и вашей сетью, если не все адреса в ней распределены. Например, если *в* адресном пространстве компании ZIP сетевой адрес 131.108.227.1 не был присвоен какому-либо конкретному сегменту сети или устройству, то маршрутизатор перешлет пакеты с таким пунктом назначения в сеть по умолчанию. Маршрутизатор соединения с глобальной сетью Internet не знает об этом адресе, так как он никому не присвоен. Однако адрес пункта назначения подходит под маршрут о . о. о. о/о, и поэтому маршрутизатор переадресует пакеты провайдеру Internet-услуг.

В свою очередь, провайдер Internet-услуг определит, что адрес 131.108.227.1 находится в сети компании ZIP (вероятно, благодаря маршруту 131.108.0. 0/16), и переправит пакеты назад маршрутизатору соединения с глобальной сетью Internet компании ZIP. Этот маршрутизатор снова не найдет конкретного маршрута, но данный адрес подходит под маршрут 0.0. 0.0/0, и он отошлет их назад провайдеру Internet-услуг. А тот повторит предыдущий шаг.

Этот процесс будет повторяться до тех пор, пока не истечет время жизни пакета. Если подобная петля возникает для большого количества пакетов, то результатом будет бесполезное расходование полосы пропускания соединения с глобальной сетью Internet и огромное количество заторов для пользователей из компании ZIP, пытающихся выйти в сеть Internet. На рис. 4.5 изображена подобная нежелательная ситуация.

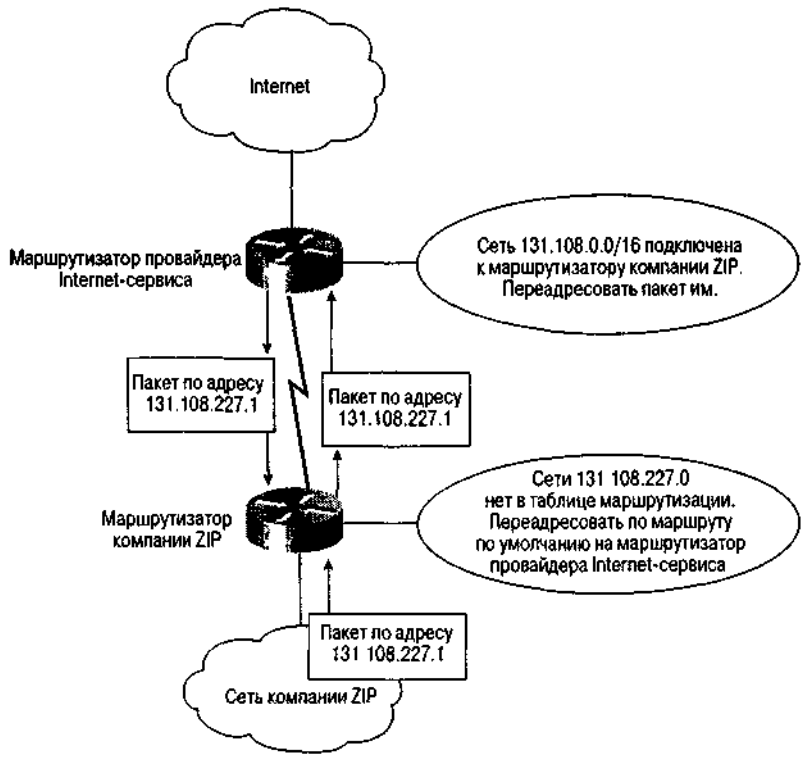


Рис. 4.5. Петля маршрутизации возникла из-за того, что пакет с неназначенным адресом был послан в бесклассовой IP-сети на адрес по умолчанию

Совет

Чтобы избежать возникновения подобной петли, необходимо обеспечить для адресного пространства компании ZIP сводный маршрут, который бы отбрасывал пакеты, адресованные на неназначенные IP-адреса, принадлежащие адресному пространству сети компании ZIP. Этого можно добиться, установив пунктом назначения несуществующий интерфейс Nail 0. Для сети компании ZIP сводным маршрутом, который будет отбрасывать пакеты в непрописанные пункты назначения, будет IP-маршрут 131.108 . 0. O 255.255.0.0 Null 0. Этот маршрут обычно устанавливается на маршрутизаторе соединения с глобальной сетью Internet, который является последним получающим пакеты маршрутизатором до их отправки провайдеру Internet-услуг.

Проверка конфигурации IP-маршрутизации

Как уже упоминалось в данной главе, основной командой для проверки конфигурации IP-маршрутизации является команда ОС IOS режима EXEC show ip route. В этом разделе рассматривается несколько других команд, которые помогают в верификации и управлении конфигурацией таблицы IP-маршрутов.

Команда show ip route является инструментом, который используется для просмотра состояния таблицы IP-маршрутизации. Сконфигурированы ли статические маршруты или работают протоколы динамической маршрутизации, эта команда показывает, присутствуют ли действительно в маршрутизаторе те маршруты, которые были сконфигурированы или, как ожидается, будут узнаны в процессе обучения. Ниже приводится часть результата исполнения команды show ip route на маршрутизаторе компании ZIP SF-Core-1:

```
SF-Core-l<fshow ip route
            Codes:   C  -  connected,   S  -  static,   I  -  IGRP,   R - RIP,   M
            -    mobile,   B  -BGP D -EIGRP,   EX -  EIGRP external,   0 -  OSPF,
            IA -  OSPF  inter area N1  - OSPF NSSA external  type  1,   N2  -  OSPF
            NSSA external  type 2 El  - OSPF external  type  1,   E2   - OSPF
            external  type  2,   E  - EGP i - IS-IS,LI -IS-IS level-1,  L2 -IS-IS
            level-2,* - candidate default U - per-user static  route, o -ODR
Gateway of last resort is  192.72.2.1 to network 0.0.0.0
131.108.0.0/16 is variably subnetted,8 subnets,3 masks
C  131.108.20.0/22  is directly connected,   FastEthernetO/0
C  131.108.240.0/30  is directly connected,   Seriall/0
S  131.108.230.0/24   [1/0]   via  131.108.240.2
S  131.108.231.0/24   [1/0]   via  131.108.100.0
S  131.108.232.0/24  [1/0] is directly connected, FastEthernetO/0
C  192.7.2.0/30 is directly connected,   Seriall/1
D  131.108.240.4/30 [90/307200] via 131.108.20.4,IdOOh,   FastEthernetO/0
D  131.108.241.0/30 [90/3182080] via 131.108.240.2,IdOOh,   Seriall/0
D  131.108.100.0/24 [90/3182080] via 31.108.240.2,IdOOh,   Seriall/0
S  131.108.0.0/16 is directly connected,   NullO
S* 0.0.0.0   [1/0]   via  192.7.2.1
```

Такой результат дает следующую информацию.

- Список всех сетевых маршрутов и масок, находящихся на текущий момент в таблице маршрутизации.
- IP-адрес следующего узла перехода и выходной интерфейс для таких маршрутов (или только выходной интерфейс для маршрутов с непосредственным соединением).
- Если маршрут возникает в процессе динамического обучения, то в зависимости от конкретного протокола маршрутизации приводится продолжительность времени (в секундах), которое маршрут находится в таблице, или продолжительность времени с момента последнего обновления информации.
- Административное расстояние и метрика протокола маршрутизации для всех маршрутов, кроме маршрутов с непосредственным соединением. Административное расстояние — число, стоящее слева от косой линии в квадратных скобках следом за

сетевым маршрутом и маской в формате с контрольной суммой. Метрика протокола маршрутизации — число, стоящее справа от косой линии в квадратных скобках.

Административное расстояние представляет собой числовое значение, отражающее степень доверительности источника обновления информации о маршрутизации. Каждому типу маршрута и протоколу маршрутизации приписывается конкретное административное расстояние. Чем меньше значение, тем доверительнее источник. В табл. 4.2 показаны административные расстояния, используемые в текущей версии ОС IOS. Метрика протокола маршрутизации представляет собой число, используемое для ранжирования маршрутов по предпочтению в тех случаях, когда существует несколько маршрутов до одного и того же пункта назначения. Часто метрика является составным числом, отражающим несколько характеристик маршрута, например, длину и стоимость пути. Каждый протокол динамической маршрутизации имеет свой алгоритм определения числового выражения метрики.

Таблица 4.2. Значения административных расстояний по умолчанию

Источник маршрута	Значение расстояния по умолчанию
Подключенный интерфейс	О
Статический маршрут	1
Усовершенствованный сводный IGRP-маршрут	5
Внешний протокол BGP	20
Внутренний усовершенствованный протокол IGRP	90
Протокол IGRP	100
Протокол OSPF	110
Протокол IS-IS	115
Протокол RIP	120
Протокол EGP	140
Внутренний протокол BGP	200
Неизвестный источник	255

Другим инструментом, позволяющим быстро определить состояние таблицы маршрутизации, является команда ОС IOS режима EXEC show ip masks. При задании сетевого адреса в качестве параметра эта команда выводит список масок, используемых совместно с конкретным сетевым адресом, и количество маршрутов, которые имеют хотя бы одну из таких масок. Эта команда часто используется при выявлении ошибок адресации и конфигурирования статических маршрутов, позволяя высвечивать упущенные или неожиданные сетевые маски, которые появляются в таблице маршрутизации. Ниже приведен пример результата исполнения команды show ip masks 131.108.0.0 на маршрутизаторе компании ZIP SF-Core-1, в котором показаны шесть различных подсетей сети 131.108.0.0.

```
SF-Core-l#show ip masks 131.108.0.0
Mask                          Reference count
255.255.255.255                      9
255.255.255.252                      5
255.255.255.128                      3
255.255.255.0                        4
255.255.252.0                        1
255.255.0.0                          1
SF-Core-l#
```

Большинство протоколов динамической маршрутизации автоматически посылают I пакеты актуализации информации о маршрутах, хранящейся в маршрутизаторах. Эти пакеты актуализации содержат последние сведения относительно добавления или удаления маршрутов из таблицы маршрутизации и информацию для поддержания свежими данных о маршрутах, находящихся на данный момент в таблице. Однако иногда может возникнуть

необходимость вручную убрать конкретную запись в таблице маршрутизации или очистить всю таблицу. Например, может понадобиться убрать динамический маршрут, который был отмечен как недостоверный, но не был удален из таблицы маршрутизации по возрасту естественным образом. Или в целях отладки, возможно, может понабиться актуализировать конкретный маршрут или всю таблицу. Для удаления одного конкретного маршрута или очистки всей таблицы маршрутизации можно воспользоваться командой ОС IOS режима EXEC clear ip route. Команда вводится либо со звездочкой, что приводит к очистке всей таблицы маршрутизации, либо с парой значений, представляющих собой сетевой адрес и маску, и тогда удаляется только этот конкретный маршрут.

Будьте осторожны при принятии решения об очистке всей таблицы маршрутизации. Актуализация всей содержащейся в ней информации может занимать от нескольких секунд до нескольких минут. В течение этого промежутка времени может пропасть связь для пакетов, проходящих через маршрутизатор, и пакетов, поступающих из сеанса терминала в сам маршрутизатор. Более того, в зависимости от конкретного используемого протокола динамической маршрутизации и размера самой *таблицы* маршрутизации очистка всей таблицы может вызвать чрезмерную загрузку центрального процессора. Ниже приведен пример очистки всей таблицы маршрутизации маршрутизатора компании ZIP SF-Core-1:

```
SF-Core-l#clear ip route *
SF-Core-l#
```

Далее показан пример удаления из маршрутизатора компании ZIP SF-Core-1 маршрута 131.108.3.0/25:

```
SF-Core-l#clear ip route 131.108.3.0 255.255.255.128
SF-Core-l#
```

Конфигурирование протоколов IP-маршрутизации

В предыдущем разделе исследовались вопросы организации среды маршрутизации и создания таблицы маршрутизации посредством статических маршрутов. Если бы все сети работали, используя только статические маршруты, они были бы сложны в управлении и не очень быстро реагировали на неисправности и изменения топологии, которые случаются довольно часто.

Для решения данных вопросов и были разработаны протоколы динамической маршрутизации. Протоколы динамической маршрутизации представляют собой алгоритмы, позволяющие маршрутизаторам сигнализировать об информации о путях в IP-сетях, которая требуется для построения таблицы маршрутизации. Эти алгоритмы также определяют критерии выбора маршрута, по которому будет следовать пакет, будучи представленным маршрутизатору для коммутации. Протоколы маршрутизации позволяют выбирать оптимальный путь через сеть, быстро реагировать на изменения в сети и делать это простейшим образом с наименьшими накладными расходами для маршрутизатора.

Протоколы маршрутизации делятся на два основных класса: протоколы внутренних шлюзов (Interior Gateway Protocols — ЮР) и протоколы внешних шлюзов (Exterior Gateway Protocols — EGP). Протоколы класса IGP проектировались для обмена информацией о сетях и подсетях между маршрутизаторами в пределах одной автономной системы, т.е. между маршрутизаторами, работающими с одним протоколом маршрутизации и в одном административном домене. Протоколы же класса EGP проектировались только для обмена сетевой информацией между маршрутизаторами и: различных автономных систем.

Наиболее широко используемым сегодня EGP-протоколом является протокол граничной маршрутизации версии 4 (Border Gateway Protocol version 4 — BGP-4). Это доминирующий протокол маршрутизации, используемый для обмена информацией о сетевых путях между компаниями, провайдерами Internet- и сетевых услуг в глобально? сети Internet. Основы конфигурирования протокола BGP-4 описываются в раздел* "Конфигурирование протокола пограничной маршрутизации Border Gateway Protocol".

Двумя основными признаками, отличающими один протокол IGP от другого, являются методология распространения информации и то, классовый это протокол или нет. Существуют две общепринятые методологии распространения информации: метод вектора расстояния и метод учета состояния каналов связи. В методе вектора расстояния каждый маршрутизатор через равные промежутки времени посылает соседним маршрутизаторам в так называемых сообщениях обновления всю или часть своей таблицы маршрутизации. По мере распространения информации о маршрутизации по сети маршрутизаторы могут вычислять расстояния до всех сетей и подсетей в пределах внутрикорпоративной сети предприятия.

В методе учета состояния каналов связи каждый маршрутизатор посылает полную информацию о локальных соединениях остальным маршрутизаторам во внутрикорпоративной сети предприятия. Поскольку каждый маршрутизатор получает информации обо всех локальных соединениях, то, обработав эту информацию о соединениях по сложному алгоритму, называемому алгоритмом выбора первым кратчайшего пути (Shortest Path First — SPF), он способен построить полную картину всей внутренне) корпоративной сети предприятия.

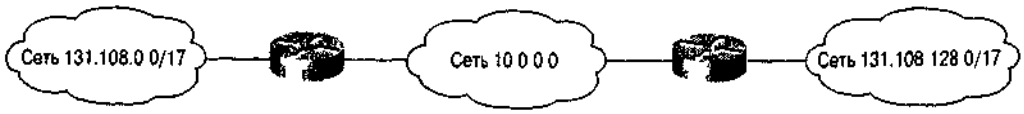


Рис. 4.6. Бесклассовые протоколы маршрутизации допускают наличие разрывного адресного пространства сети

На выбор протокола динамической маршрутизации оказывает влияние множество факторов. И хотя проблеме проектирования сетей и выбора протокола маршрутизации может быть посвящена целая книга, ниже приведено несколько ключевых моментов, влияющих на решение администратора сети.

- **Топология сети.** Некоторые протоколы маршрутизации для достижения соответствующего уровня масштабируемости и распространения информации о сетевых путях подразумевают наличие логической иерархии. Такие протоколы, как OSPF и IS-IS, требуют организации магистральной и логических областей (рис. 4.7). Эти протоколы могут потребовать перепроектирования физической топологии сети или хорошей инженерной проработки начального проекта сети для оптимальной ее работы.
- **Суммирование адресов и маршрутов.** В большой внутренней корпоративной сети пред приятия уменьшение количества записей в таблицах маршрутизации, поддерживаемых узлами маршрутизации, снижает относительную сложность сети и уменьшает на грузку на маршрутизаторы. Суммирование требует, чтобы протокол маршрутизации поддерживал маски подсетей переменной длины и был способен распространять ин формацию о сетевых масках вместе с информацией о сетевых маршрутах. Хорошо подходят для реализации функции суммирования такие бесклассовые протоколы, как OSPF и EIGRP.
- **Скорость сходимости.** Скорость, с которой протокол маршрутизации определяет, что путь недоступен, выбирает новый путь и распространяет информацию о новом пути, может быть одним из наиболее важных критериев.

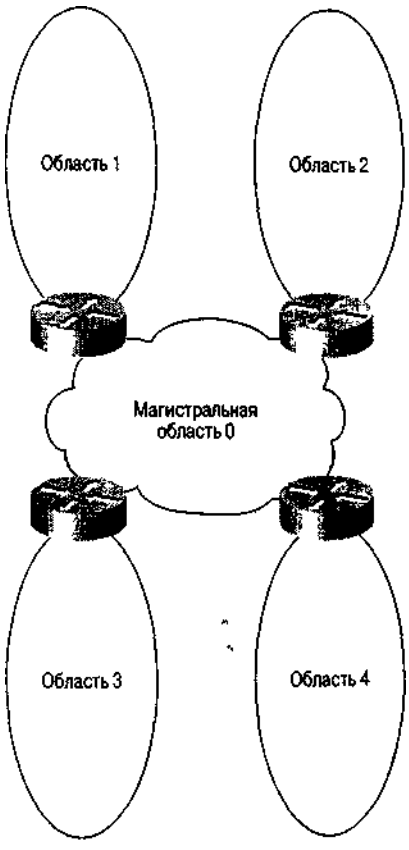


Рис. 4.7. Иерархическая топология сети

Если сеть поддерживает очень важные приложения, то вполне вероятно, что администратору сети захочется иметь протокол маршрутизации с быстрой сходимостью. Как правило, протоколы, основанные на методе вектора расстояния, требуют большего времени для сходимости, чем протоколы с выбором по состоянию канала связи, так как информация о новом пути должна проходить сегмент за сегментом к каждому следующему маршрутизатору внутренней корпоративной сети предприятия. Таким образом, протоколы RIP версии 1 и IGRP медленнее покрывают сеть, чем протоколы EIGRP и OSPF.

- **Критерий выбора маршрута.** Атрибуты пути, используемые протоколом маршрутизации для формирования своей метрики маршрута, играют ключевую роль в определении, какой протокол динамической маршрутизации реализовывать. Протокол, который при выборе маршрута полагается строго на количество переходов от маршрутизатора к маршрутизатору, например протокол RIP, не рекомендуется использовать в тех случаях, когда во внутрикорпоративной сети множество путей содержат участки сред локальных и глобальных сетей различных типов. В частности, для протокола RIP переход по сегменту Fast Ethernet стоит столько же, сколько переход по каналу в технологии глобальных сетей с полосой 56 Кбит/с. К атрибутам сетевого пути, используемым различными протоколами для вычисления его метрики, могут относиться длина пути, надежность, время задержки распространения, ширина полосы пропускания и уровень загрузки.
- **Масштабируемость.** Относительная масштабируемость протокола маршрутизации зависит от типов маршрутизаторов, стоящих во внутрикорпоративной сети предприятия, и размера сети. Протоколы, основанные на методе вектора расстояния, потребляют меньше тактов центрального процессора, чем протоколы с выбором по состоянию каналов связи с их сложными SPF-алгоритмами. С другой стороны, протоколы с выбором по состоянию каналов связи занимают меньшую часть полосы пропускания локальной или глобальной сети, так как распространяется " только информация об изменениях, а не вся таблица маршрутизации.
- **Легкость реализации.** Если сеть не слишком сложна, то протоколы, которые не требуют реинжениринга сети или хорошо структурированной и организованной топологии, проще внедрять. Например, протоколы RIP, IGRP и EIGRP для эффективной работы не требуют больших затрат на планирование или организацию топологии. С другой стороны, протоколы OSPF и IS-IS требуют, чтобы топология сети и схемы адресации были хорошо продуманы еще до развертывания.
- **Безопасность.** Если сеть может обмениваться IGP-информацией с компанией-партнером или подразделениями внутри корпорации, то, вероятно, понадобится аутентификация источника маршрутной информации. Такие протоколы, как OSPF и EIGRP, поддерживают эффективные методы аутентификации, например, аутентификацию по ключу MD5.

Примечание
Развернутая оценка работы и технических характеристик различных протоколов приводится в документе *Technology Overview Briefs* (Краткий обзор технологий), который размещен на странице сертифицированных пользователей устройств компании Cisco ПО адресу www.cisco.com/univercd/ac/td/doc/cisintwk/ito_doc/index.htm.

Выбор протокола маршрутизации для любой сети в значительной степени зависит от следующих факторов.

- Добавляется ли маршрутизатор в уже существующую топологию сети.
- Каков конкретный проект сети.
- Наличие существующих маршрутизаторов и протоколов маршрутизации.
- Чувство комфорта и опыт работы в области TCP/IP-маршрутизации у администратора сети.
- Необходимость обмена маршрутной информацией с оконечными устройствами системы, например серверами.

Если вы не знаете, какой протокол маршрутизации выбрать, обсудите различные варианты с техническими специалистами торгового представительства, внешними консультантами по

вопросам создания сетей или другими лицами, которые имеют опыт в развертывании IP-сетей. Кроме того, компанией Cisco Systems создано подробное руководство по проектированию под названием *Designing Large-Scale IP Internetworks* (Проектирование крупномасштабных IP-сетей), в котором подробно описываются проектирование и критерии выбора протоколов динамической маршрутизации при развертывании сети передачи данных. Это руководство можно найти на странице зарегистрированных пользователей устройств компании Cisco по адресу www.Cisco.com/univercd/cc/td/doc/cisintwk/idg4/nd2003.htm.

Для сети компании ZIP был выбран протокол EIGRP. В нем сочетаются характеристики, свойственные протоколам, работающим по методу вектора расстояния, и характеристики, свойственные протоколам на основе метода выбора пути по состоянию канала связи. Он относительно легок в конфигурировании, не требует специфической физической топологии, поддерживает функцию суммирования и подсетевые маски переменной длины и дает быструю сходимость. Основы конфигурирования протокола EIGRP обсуждаются в разделе "Конфигурирование усовершенствованного IP-протокола IGRP компании Cisco". Хотя в сети компании ZIP не используются другие популярные протоколы класса IGP, основы их конфигурирования описываются в последующих разделах. При обсуждении этих вопросов описываются действия, выполняемые при конфигурировании каждого из протоколов. Дополнительные команды, применяемые для управления информацией протоколов динамической маршрутизации, описываются в разделе "Управление информацией протоколов динамической маршрутизации".

Конфигурирование протокола маршрутной информации Routing Information Protocol

Протокол маршрутной информации (Routing Information Protocol — RIP) является одним из самых старых протоколов, используемых в IP-ориентированных устройствах. Первый вариант его реализации был сделан в начале 1980-х в рамках протокола PUP компании Xerox. Протокол RIP стал популярным после того, как был распространен в составе UNIX-версии продукта Berkley Systems Distribution (BSD) в качестве протокола маршрутизации в этой реализации TCP/IP-системы. Официальная спецификация протокола RIP как протокола TCP/IP-маршрутизации содержится в Запросе на комментарий № 1058.

RIP является протоколом на основе метода вектора расстояния, использующим в качестве метрики подсчет количества переходов между маршрутизаторами. Максимальное количество переходов в протоколе RIP — 15. Любой маршрут, длиннее 15 переходов, снабжается тэгом недостижимого с помощью установки счетчика переходов в значение 16. Маршрутная информация в протоколе RIP распространяется от маршрутизатора к его соседям путем отправки широковещательных IP-пакетов с использованием протокола UDP и порта 520.

Протокол RIP версии 1 представляет собой классовый протокол и не поддерживает распространение информации о сетевых масках. А вот протокол RIP версии 2 уже является бесклассовым протоколом и может поддерживать CIDR-адресацию, маски переменной длины, суммирование маршрутов и имеет средства защиты в виде аутентификации по ключу MD5 и открытого (нешифрованного) текста.

Хотя протокол RIP не используется в сети компании ZIP, предположим, что он конфигурируется на маршрутизаторе с условным названием RIProuter. Конфигурирование протокола маршрутизации RIP состоит из трех основных этапов: разрешение маршрутизатору исполнять протокол RIP, выбор версии этого протокола и задание сетевых адресов и интерфейсов, которые должны включаться в пакеты обновления данных маршрутизации. Для выдачи разрешения маршрутизатору на исполнение протокола RIP используется основная команда конфигурирования ОС IOS router rip. Выбор исполняемой версии осуществляется с помощью субкоманды конфигурирования маршрутизации ОС IOS version. Для задания версии команда version имеет параметр, принимающий значения 1 или 2. Если номер версии не задается, то ОС IOS по умолчанию конфигурирует применение протокола RIP версии 1 для отправки пакетов обновления, но прием осуществляется как пакетов версии 1, так и пакетов версии 2. В примере ниже разрешается исполнение протокола

маршрутизации RIP версии 2:

```
RIProuter#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
RIProuter (config)#router rip
RIProuter (config-router)#version 2
RIProuter (config-router)#^Z
```

Для определения интерфейсов и сетевых адресов, подлежащих включению в извещения с маршрутной информацией протокола RIP, используется субкоманда конфигурирования маршрутизации ОС IOS network. В качестве параметра этой команды выступает классовый сетевой адрес сети, подлежащий включению в пакеты обновления маршрутной информации. Команда network должна использоваться для идентификации только тех сетевых IP-адресов, которые непосредственно подключены к конфигурируемому маршрутизатору и предназначены для включения в пакеты обновления маршрутной информации. В пакеты обновления включаются только те интерфейсы, которые имеют IP-адреса, принадлежащие идентифицированной сети.

Предположим, что маршрутизатор имеет два интерфейса с IP-адресами 131.108.4.5 и 131.108.6.9, соответственно, и третий интерфейс с IP-адресом 172.16.3.6. В этом случае ввод субкоманды network 131.108.0.0 приведет к тому, что объявления с маршрутной информацией будут посылаться только с данными о подсетях сети 131.108.0.0 и только в интерфейсы, которые адресуются в сети 131.108.0.0. Чтобы включить в пакеты обновления маршрутной информации интерфейс, находящийся в адресном пространстве 172.16.0.0, должна быть введена дополнительная команда network 172.16.0.0.

Ниже приводится пример конфигурирования команды network на включение подсетей и интерфейсов сети 131.108.0.0:

```
RIProuter#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
RIProuter(config)#router rip
RIProuter(config-router)#network 131.108.0.0
RIProuter(config-router)#^Z
```

Примечание

Возможно сочетание в одной сети протоколов RIP версии 1 и RIP версии 2, хотя версия 1 и не поддерживает многие из функций версии 2. Смешение версий может вызвать проблемы взаимодействия. Отмена глобально сконфигурированной версии и задание версии поинтерфейсно выполняются субкомандами конфигурирования интерфейса ОС IOS ip rip send version И ip rip receive version.

Конфигурирование протокола внутренней маршрутизации между шлюзами компании Cisco Systems Interior Gateway Routing Protocol

Протокол внутренней маршрутизации между шлюзами компании Cisco Systems (Cisco Systems Interior Gateway Routing Protocol — IGRP) представляет собой усовершенствованный протокол на основе метода вектора расстояния, разработанный компанией в середине 1980-х годов. Он был спроектирован для устранения некоторых недостатков протокола RIP и обеспечения лучшей поддержки крупных сетей с каналами связи, обладающими разными характеристиками полосы пропускания.

Протокол IGRP рассчитывает метрику на основе нескольких задаваемых пользователем атрибутов сетевого пути, которые включают межсетевую задержку, полосу пропускания, надежность и загрузку. Каждый интерфейс глобальной и локальной сети имеет предварительно сконфигурированные значения ширины полосы пропускания и времени задержки, получаемые на

основе относительного быстродействия и возможностей интерфейса. Атрибуты надежности и загрузки вычисляются на основе производительности интерфейса при обработке реального трафика в сети, хотя в ОС IOS компании Cisco функция их расчета не активируется по умолчанию при принятии решений относительно маршрутизации.

Как и в протоколе RIP, в протоколе IGRP для обмена маршрутной информацией между маршрутизаторами используются широковещательные IP-пакеты. Однако протокол IGRP имеет свой собственный протокол транспортного уровня. Для обмена информацией о сетевых маршрутах в нем не применяются протоколы UDP или TCP. (Поскольку в протоколе IGRP нет механизмов обратной связи, он работает подобно протоколу UDP.)

Протокол IGRP по сравнению с протоколом RIP имеет три основных усовершенствования. Во-первых, его метрика может поддерживать сеть с количеством переходов между маршрутизаторами до 255. Во-вторых, его метрика способна различать разные типы сред связи и их соответствующие стоимости. В-третьих, благодаря применению пакетов обновления с мгновенной рассылкой, протокол IGRP обеспечивает более быструю сходимость. Мгновенная рассылка пакетов обновления предусматривает отправку информации об изменениях в сети, как только она становится доступной, а не при наступлении заданного времени обновления.

Давайте рассмотрим задачу конфигурирования протокола 1GRP на примере маршрутизатора с условным названием IGRProuter. Процесс конфигурирования маршрутизации в соответствии с протоколом IGRP состоит из двух этапов, а именно: разрешения маршрутизатору исполнения протокола 1GRP и идентификации сетевых адресов и интерфейсов, которые должны включаться в пакеты обновления данных маршрутизации. Чтобы выдать маршрутизатору разрешение на исполнение протокола IGRP, используется основная команда конфигурирования ОС IOS router igrp. Эта команда требует параметра, известного под названием "идентификатор процесса" (process-fd). Идентификатором процесса может быть целое число в диапазоне значений от 1 до 65535. Поскольку на одном маршрутизаторе может исполняться множество процессов, чтобы их различать и нужны номера идентификаторов процесса. Несколько IGRP-процессов могут исполняться, например, на маршрутизаторе, который соединяет два подразделения компании, каждое из которых хочет иметь отдельное администрирование своей сети. Все маршрутизаторы в пределах подразделения используют один и тот же идентификатор IGRP-процесса.

Как и при использовании протокола RIP, задание интерфейсов и сетевых адресов, подлежащих включению в извещения с маршрутной информацией протокола IGRP выполняется с помощью субкоманды конфигурирования маршрутизации ОС IOS network. В качестве параметра этой команды выступает классовый сетевой адрес сети, подлежащий включению в пакеты обновления маршрутной информации. Команда network должна использоваться для идентификации только тех сетевых IP-адресов, которые непосредственно подключены к конфигурируемому маршрутизатору и предназначены для включения в процесс IGRP-маршрутизации. В пакеты обновления включаются только те интерфейсы, которые имеют IP-адреса, принадлежащие идентифицированной сети.

Например, если существуют два интерфейса с IP-адресами 131.108.4.5 и 131.108.6.9, соответственно, и третий интерфейс с IP-адресом 172.16.3.6, то ввод субкоманды network 131.108.0.0 приведет к тому, что объявления с маршрутной информацией будут посылаться только с данными о подсетях сети 131.108.0.0 и только в интерфейсы, которые адресуются в сети 131.108.0.0. Чтобы включить в пакеты обновления маршрутной информации интерфейс, находящийся в адресном пространстве 172.16.0.0, вводится дополнительная команда network 172.16.0.0.

Ниже приводится пример конфигурирования процесса IGRP-маршрутизации для сети 131.108.0.0:

```
IGRProuter#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
IGRProuter(config)#router igrp 25000
IGRProuter(config-router)#network 131.108.0.0
IGRProuter(config-router)#^Z
```

Конфигурирование открытого протокола выбора первым наикратчайшего пути Open Shortest Path First Protocol

Открытый протокола выбора первым наикратчайшего пути (Open Shortest Path! First Protocol— OSPF) был спроектирован в конце 1980-х годов рабочей группой) OSPF Комитета по инженерным проблемам Internet (IETF). Его проектирование было осуществлено в ответ на потребности IP-ориентированных сетей в поддержке масок подсетей переменной длины, аутентификации источников маршрутов, быстрой сходимости, тэгирования маршрутов, получаемых через внешние протоколы маршрутизации и многоадресной рассылки извещений о маршрутах. Протокол OSPF версии 2 (в настоящее время наиболее часто используемый) специфицирован в Запросе на комментарий №1583.

Протокол OSPF работает, деля большую внутрикорпоративную сеть или автономную систему на мелкие иерархические блоки. Каждая из таких областей соединена с магистральной областью через маршрутизатор границы области (см. рис. 4.7). Все пакеты, адресованные рабочей станцией в одной области рабочей станции из другой области, проходят через магистральную область вне зависимости от того, существует ли прямое соединение одной области с другой. Хотя возможна работа OSPF-сети при существовании только магистральной области, масштабируемость протокола OSPF хороша только тогда, когда сеть поделена на ряд более мелких областей.

Как уже описывалось ранее, OSPF представляет собой протокол, основанный на методе учета состояния каналов связи. В отличие от протоколов RIP и IGRP, которые извещают о своих маршрутах только соседние маршрутизаторы, OSPF-маршрутизаторы посылают свои извещения о состоянии каналов всем маршрутизаторам, находящимся в пределах одной иерархической области. Эти извещения содержат данные о подсоединенных интерфейсах, используемой метрике и другую информацию, необходимую для расчета сетевого пути и заполнения базы данных топологии. OSPF-маршрутизаторы накапливают информацию о состоянии каналов связи и затем обрабатывают ее с применением алгоритма SPF (также известного под названием алгоритма Дейкстры (Dijkstra), названного так в честь его автора), рассчитывая кратчайший путь до каждого узла.

Для определения интерфейсов, которые принимают извещения о состоянии каналов, маршрутизаторы исполняют протокол приветствия OSPF Hello. Соседние маршрутизаторы обмениваются сообщениями приветствия, определяя при этом, какие маршрутизаторы сидят на данном интерфейсе и подают сигнал "я живой", показывающий, что эти маршрутизаторы все еще доступны для обращений.

Как только обнаруживается соседний маршрутизатор, происходит обмен топологической информацией протокола OSPF. После того как маршрутизаторы синхронизируются, они сообщают, что соседство признано. Извещения о состоянии каналов посылаются и принимаются только после признания соседства. Информация извещений о состоянии каналов переносится в пакетах на транспортном уровне протокола OSPF. На транспортном уровне протокола OSPF определяется надежность извещения, пересылается подтверждение и реализуется процесс запроса, гарантирующего правильность рассылки извещений о состоянии каналов всем маршрутизаторам, находящимся в пределах области. Извещения о состоянии каналов подразделяются на четыре типа. К самым распространенным относятся те, которые извещают о каналах подключенных к маршрутизатору сетей, и те, которые извещают о сетях, имеющихся вне OSPF-областей.

Метрика маршрутизации в протоколе OSPF вычисляется как сумма OSPF-стоимостей вдоль пути, ведущего к сети. OSPF-стоимость рассчитывается на основе полосы пропускания интерфейса и конфигурируется пользователем.

Рассмотрим базовое конфигурирование протокола OSPF на примере маршрутизатора с именем OSPFrouter. Конфигурирование процесса OSPF-маршрутизации состоит из двух этапов. Во-первых, следует разрешить маршрутизатору исполнять протокол OSPF. Во-вторых, необходимо идентифицировать сетевые адреса и интерфейсы, которые должны включаться в пакеты обновления данных маршрутизации, и определить, каким областям эти интерфейсы принадлежат.

Чтобы разрешить маршрутизатору работать с протоколом OSPF, используется основная команда конфигурирования ОС IOS router ospf. Если на одном маршрутизаторе исполняется несколько OSPF-процессов, эта команда требует в качестве параметра целочисленное значение идентификатора процесса. Как и в других протоколах маршрутизации, необходимо сконфигурировать интерфейсы и адреса сетей, которые должны включаться в маршрутные извещения протокола OSPF. Кроме того, следует идентифицировать OSPF-область, в которой размешается интерфейс.

Для идентификации сетевых адресов и интерфейсов, включаемых в работу по протоколу OSPF, а также для идентификации областей, которым они принадлежат, используется субкоманда конфигурирования маршрутизации ОС IOS network area. Эта команда имеет два параметра. Первый параметр представляет собой сетевой адрес и подстановочную маску, используемую для сравнения с IP-адресами, назначенными интерфейсам. *Подстановочная маска* — это метод сопоставления IP-адресов и диапазонов IP-адресов. Он описывается в разделе "Конфигурирование IP-фильтрации с помощью списков доступа". Если подстановочная маска накладывается на IP-адрес интерфейса, и получившийся в результате сетевой адрес согласуется с сетевым адресом в команде network area, то интерфейс включается в процесс OSPF-маршрутизации для заданной области. Второй параметр, называемый идентификатором области, используется для идентификации области, к которой принадлежит интерфейс. Идентификатором области может быть целое число или число в десятичной форме с разделением точками, аналогичное IP-адресу.

Предположим, что наш используемый для примера маршрутизатор имеет три интерфейса. Интерфейсам присвоены IP-адреса 131.108.200.1, 131.108.201.1 и 131.108.202.1, соответственно. Первые два интерфейса приписаны к области 1, а третий — к области 0, или магистральной области. Основываясь на этих предположениях, составлен приведенный ниже пример конфигурирования протокола OSPF:

```
OSPFrouter#configure
Configuring from terminal,   memory,   or network   [terminal]?
Enter configuration commands,   one per  line.   End with CNTL/Z.
OSPFrouter(config)#router ospf 25000
OSPFrouter(config-router)#network 131.108.200.0  0.0.1.255 area 1
OSPFrouter(config-router)#network 131.108.202.0  0.0.0.255 area 0
OSPFrouter(config-router) #^Z
```

Как и в ранее обсуждавшихся протоколах маршрутизации, пакеты обновления маршрутной информации протокола OSPF включают только те сетевые адреса и интерфейсы, которые совпадают с адресами в команде network area.

Протокол OSPF работает, предполагая, что извещения о состоянии каналов могут представлять собой пакеты многоадресной рассылки в пределах автономной системы. Однако многие среды глобальных сетей, например, двухточечные последовательные линии связи, двухточечные каналы с интерфейсом Frame Relay или многоточечные каналы Frame Relay, не являются средой для широковещания и не поддерживают многоадресную рассылку. Не имея возможности осуществлять многоадресную рассылку маршрутной информации, администратор сети вынужден конфигурировать отношения соседства между маршрутизаторами на двухточечных и многоточечных сетевых интерфейсах вручную. Однако есть одно решение, которое исключает такое конфигурирование соседей. Протокол OSPF может рассматривать двухточечный интерфейс в качестве среды широковешания и многоточечный интерфейс в качестве сети с частичным широковещанием. Типом сети, которому (для протокола OSPF) принадлежит подключенная к интерфейсу сеть, управляет субкоманда конфигурирования интерфейса ОС IOS ip ospf network. Команда воспринимает в качестве параметра одну из следующих опций:

- broadcast (широковещательная) — дает указание рассматривать такую среду в качестве среды с широковещанием, предполагая, что многоадресная рассылка может отсылаться и приниматься;
- non-broadcast (нешироковешательная) — дает указание рассматривать среду как такую,

которая не допускает широковещания. Эта опция требует от администратора конфигурировать отношения соседства вручную с помощью субкоманды конфигурирования маршрутизации neighbour;

- point-to-multipoint (из точки многим) — дает указание рассматривать та кую среду в качестве среды с частичным широковещанием. В топологии "из точки многим" маршрутизатор, подключенный к концентратору, имеет виртуальные каналы к нескольким удаленным маршрутизаторам. Для сообщений о состоянии каналов и при маршрутизации между маршрутизаторами, не имеющими прямого подключения, он выполняет роль ретранслятора.

Ниже приведен пример конфигурирования двухточечного подынтерфейса Frame Relay в качестве интерфейса широковещательного типа для протокола OSPF и многоточечного интерфейса Frame Relay в качестве интерфейса типа "из точки многим":

```
OSPFrouter#configure
Configuring  from terminal,  memory,   or network   [terminal]?
Enter configuration commands,   one per line.   End with CNTL/Z.
OSPFrouter(config)#interface serial  0.1 point-to-point
OSPFrouter(config-int)#ip ospf network broadcast
OSPFrouter(config-int)#interface  serial  1
OSPFrouter(config-int)#ip ospf network point-to-multipoint
OSPFrouter(config-int)#^Z
```

В отличие от других протоколов маршрутизации класса IGP, протокол OSPF не генерирует маршрут по умолчанию при конфигурировании командой ip default-network. При использовании протокола OSPF в маршрутизаторе, стоящем на границе автономной системы, принудительная генерация маршрута по умолчанию в остальную часть OSPF-домена должна конфигурироваться вручную. Генерацию маршрута по умолчанию в рамках протокола OSPF вызывает субкоманда конфигурирования ОС IOS ip default-information originate. Ниже приведен пример конфигурирования с помощью команды ip default-information originate совместно С командой ip default-network маршрутизатора, стоящего на границе автономной системы, для принудительной генерации маршрута по умолчанию:

```
OSPFrouter#configure
Configuring  from terminal, memory, or    network [terminal]?
Enter  configuration commands, one per    line.   End with  CNTL/Z.
OSPFrouter(config)#ip default-network     140.222.0.0
OSPFrouter(config-router)#router ospf     25000
OSPFrouter(config-router)#ip default-information originate
OSPFrouter(config-router)#^z
```

Конфигурирование усовершенствованного IP-протокола IGRP компании Cisco

Усовершенствованный протокол внутренней маршрутизации между шлюзами (Enhanced Interior Gateway Routing Protocol — EGRP) представляет собой улучшенную версию исходного протокола IGRP, разработанного компанией Cisco Systems. В протоколе EIGRP сохранены тот же алгоритм вектора расстояния и метрическая информация, что и в исходном протоколе IGRP. Однако время сходимости и другие аспекты масштабируемости были значительно усовершенствованы. Теперь в нем имеются функции, которых не было у его предшественника — протокола IGRP, например, поддержка масок подсетей переменной длины и суммирование произвольных маршрутов. Кроме того, в протоколе EIGRP появились функции, которые можно найти в протоколах, подобных протоколу OSPF, включая пакеты актуализации с частичным инкрементальным обновлением информации и уменьшенное время сходимости. Другими словами, протокол EIGRP объединяет в себе лучшие свойства протоколов, работающих по методу учета состояния каналов связи, и протоколов, работающих по методу вектора расстояния.

Как и в протоколе IGRP, в протоколе EIGRP информация таблицы маршрутизации распространяется только между соседними маршрутизаторами. Однако в отличие протокола IGRP эти

соседи выявляются с помощью простого протокола приветствий, которыми маршрутизаторы обмениваются по той же физической сети. После выявления соседей протокол EIGRP использует надежный транспортный протокол, гарантирующий точную и упорядоченную доставку информации таблицы маршрутизации и пакетов актуализации. Маршрутизатор отслеживает не только свои собственные подключенные маршруты, но и все маршруты, о которых извещают его соседи. Основываясь на этой информации, протокол EIGRP может быстро и эффективно выбрать путь до пункта назначения с минимальной стоимостью и гарантировать, что этот путь не является частью петли маршрутизации. Храня маршрутную информацию соседей, алгоритм способен быстрее определить замещающий маршрут или жизнеспособного преемника в случае отказа канала или другого события, связанного с изменением топологии.

Приветствия протокола EIGRP и маршрутная информация переносятся транспортным протоколом, являющимся элементом протокола EIGRP. Этот транспортный протокол определяет порядок надежного осуществления извещений, получения подтверждений и процесса запросов, гарантирующих правильную передачу приветствий и маршрутной информации всем соседним маршрутизаторам.

Протокол EIGRP является протоколом динамической маршрутизации для выбранной в качестве примера сети компании ZIP. Рассмотрим его конфигурирование в данном контексте. Конфигурирование EIGRP-процесса маршрутизации состоит из двух этапов: разрешения маршрутизатору исполнять протокол EIGRP и идентификации сетевых адресов и интерфейсов, включаемых в пакеты актуализации маршрутной информации.

Для выдачи разрешения маршрутизатору на исполнение протокола EIGRP используется основная команда конфигурирования ОС IOS router eigrp. При исполнении на одном маршрутизаторе нескольких процессов протокола EIGRP эта команда требует указывать в качестве параметра целочисленный идентификатор процесса. Как и в протоколе IGRP, задание интерфейсов и сетевых адресов, включаемых в извещения с маршрутной информацией протокола EIGRP, выполняется с помощью субкоманды конфигурирования маршрутизации ОС IOS network. В качестве параметра этой команды выступает классовый сетевой адрес сети, подлежащий включению в пакеты обновления маршрутной информации. Команда network должна использоваться для идентификации IP-адресов только тех сетей, которые непосредственно подключены к конфигурируемому маршрутизатору и предназначаются для включения в процесс EIGRP-маршрутизации. В пакеты обновления включаются также только те интерфейсы, которые имеют IP-адреса, принадлежащие идентифицированной сети.

Например, на маршрутизаторе компании ZIP SF-Core-1 имеются интерфейсы, стоящие в сети 131.108.0.0 и в сети 192.7.2.0. Команда network 131.108.0.0 свидетельствует о том, что извещения посылаются с маршрутной информацией о подсетях сети 131.108.0.0, и посылаются они интерфейсам, которые адресуются в сети 131.108 .0.0. Для того чтобы включить пакеты актуализации маршрутной информации в состав данных, передаваемых для интерфейса, находящегося в адресном пространстве 192 . 7 . 2 . О, будет необходима дополнительная команда network 192 .7.2.0. В нашем случае сеть 192.7.2.0 подключена к провайдеру Internet-услуг. Она не включается в процесс EIGRP-маршрутизации, так как провайдер Internet-услуг не использует протокол EIGRP.

Ниже приведен пример конфигурирования протокола EIGRP на маршрутизаторе компании ZIP SF-Core-1 для сети 131.108.0.0:

```
SF-Core-l#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands,one per line.End with CNTL/Z.
SF-Core-1(config)#router eigrp 25000
SF-Core-1(config-router)#network 131.108.0.0
SF-Core-1(config-router)#^Z
```

Конфигурирование протокола пограничной маршрутизации Border Gateway Protocol

Протокол пограничной маршрутизации (Border Gateway Protocol — BGP) относится к классу протоколов внешней маршрутизации (Exterior Gateway Protocol — EGP). В отличие от

протоколов внутренней маршрутизации или протоколов класса IGP, которые обмениваются информацией о сетях и подсетях внутри одного домена маршрутизации или автономной системы, протоколы класса EGP спроектированы для обмена маршрутной информацией между доменами маршрутизации или автономными системами. Протокол BGP является основным методом обмена сетевой информацией между компаниями, провайдерами Internet и сетевых услуг в рамках глобальной сети Internet. Этот протокол обеспечивает ряд преимуществ по сравнению со своим предшественником — протоколом внешней маршрутизации EGP. Наиболее явным преимуществом является то, что он гарантирует отсутствие зацикливания обмена информацией между автономными системами. Самой последней модификацией протокола BGP является его версия 4. Она в отличие от предыдущих версий может работать с CIDR-блоками. Стандарт BGP, который был утвержден комитетом IETF, определен в Запросах на комментарий с номерами 1163, 1267 и 1771. В этих документах описываются протоколы BGP версий 2, 3 и 4, соответственно.

BGP-маршрутизаторы конфигурируются информацией соседей, так что они могут формировать надежное TCP-соединение, по которому и осуществляют транспортировку информации о сетевых маршрутах и путях автономной системы. В отличие от некоторых протоколов класса ЮР протокол BOP использует в качестве транспортного протокола протокол TCP, а не вводит свой собственный. После установки BGP-сеанса между соседями этот сеанс остается открытым все время, если только не закрывается специально или не происходит отказ канала связи Когда два соседних маршрутизатора обмениваются маршрутной информацией в рамках BGP-сеанса, говорят, что они являются *одноранговыми BGP-узлами.* Информация о маршрутах, которой обмениваются одноранговые узлы, включает пары номер сети/путь автономной системы и другие атрибуты маршрута. Путь автономной системы представляет собой цепочку из номеров автономных систем, через которые можно попасть на сообщаемый маршрут.

Первоначально одноранговые BGP-узлы обменивались всем содержимым своих таблиц BGP-маршрутизации. Впоследствии между одноранговыми узлами стали пересылаться инкрементальные пакеты актуализации, сообщающие только о новых или аннулированных маршрутах В отличие от IGP-таблиц маршрутов BGP-таблицы маршрутов не требуют периодического обновления информации. Вместо этого каждый BGP-маршрутизатор сохраняет номер последней версии таблицы, которую он объявил своим одноранговым соседям, а также версию своей внутренней таблицы. Когда от однорангового узла принимается изменение, внутренняя таблица инкрементируется и сравнивается с версиями объявленных таблиц от одноранговых узлов. Этот процесс гарантирует, что каждый из одноранговых маршрутизаторов синхронизируется со всеми происходящими изменениями. В рамках протокола BOP также ведется отдельная таблица маршрутов, которая содержит все возможные пути до объявленных сетей. В первичной таблице выбора маршрута маршрутизатора хранится только оптимальный путь, и только он объявляется одноранговым BGP-узлам.

Одноранговые BGP-узлы делятся на две категории- внешние одноранговые BGP-узлы (EBGP) и внутренние одноранговые BGP-узлы (IBGP). Одноранговые BGP-узлы, которые находятся в разных административных доменах и обмениваются маршрутной информацией, называют *одноранговыми EBGP-узлами.* Обычно одноранговые EBGP-узлы — это другие организации, провайдеры Internet или сетевых услуг, с которыми автономная система хочет совместно пользоваться информацией о маршрутах внутри автономной системы или маршрутах, сведения о которых были получены из других внешних источников.

Одноранговые BGP-узлы, которые находятся в одном административном домене и обмениваются маршрутной информацией, называют *одноранговыми IBGP-узлами.* Одноранговыми lBGP-узлами являются маршрутизаторы, находящиеся в пределах одной автономной системы, которым необходимо коллективно использовать информацию о полученных извне BGP-маршрутах, чтобы иметь полную картину всех возможных маршрутов к внешним пунктам назначения и переобъявлять их другим одноранговым EBGP-узлам. Взаимодействие между внутренними одноранговыми BGP-узлами является типичным решением в случае, когда автономная система имеет несколько связей с внешним одноранговым BGP-узлом. Например, это может быть наличие двух подключений к глобальной сети Internet через провайдеров Internet или сетевых услуг. Организация взаимодействия

одноранговых IBGP-узлов является более простым и гибким методом коллективного использования маршрутов, полученных от одноранговых EBGP-узлов.

Альтернативой организации IBGP-взаимодействия является редистрибуция информации о маршрутах, полученных из EBGP-узлов, в какой-нибудь протокол класса IGP, например EIGRP или OSPF, для транспортировки по автономной системе с последующей редистрибуцией маршрутов из IGP-протокола в протокол BOP для их объявления через EBGP-узел другим внешним одноранговым BGP-узлам. Как описывается в разделе "Управление информацией протоколов динамической маршрутизации", редистрибуция маршрутов может спровоцировать потерю информации о метрике маршрутизации и возникновение петель маршрутизации. Кроме защиты от опасностей редистрибуции маршрутов, организация взаимодействия одноранговых IBGP-узлов обеспечивает полный административный контроль, реализацию функций взвешивания и фильтрации, связанных с протоколом BOP. Вдобавок, она позволяет поддерживать непротиворечивость маршрутной информации, объявляемой внешнему миру с помощью протокола BOP.

На рис. 4.8 показано различие между одноранговыми IBGP- и EBGP-узлами на примере сети компании ZIP. EBGP-узлы — это пары маршрутизаторов Seoul-1 и ISP-A, SF-Core-1 и ISP-B. IBGP-узлы — это пара маршрутизаторов Seoul-1 и SF-Core-1. Являясь одноранговыми IBGP-узлами, маршрутизаторы Seoul-1 и SF-Core-1 будут совместно использовать маршрутную информацию, полученную от маршрутизаторов ISP-A и ISP-B, чтобы определить наилучший маршрут до пункта назначения, находящегося за пределами сети компании ZIP.

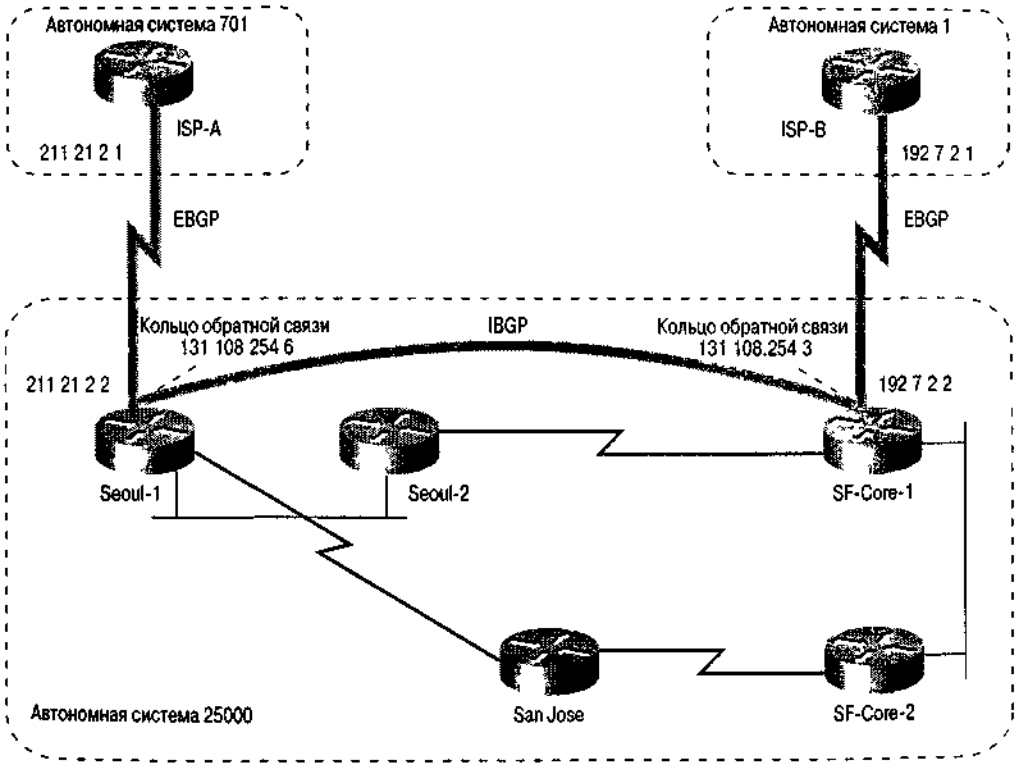


Рис. 4.8 Одноранговые EBGP- и IBGP-узлы в сети компании ZIP

Если не применяется административное регулирование, выбор оптимального маршрута в BGP-протоколе основывается на длине пути автономной системы для сетевого маршрута. Длина определяется как количество отдельных автономных систем, требующихся для того, чтобы достигнуть сети. Чем короче длина, тем лучше путь. Благодаря наличию административного регулирования протокол BOP представляет собой наиболее гибкий, с высокой степенью конфигурируемости протокол маршрутизации. С помощью различных атрибутов маршрута, например, метрики на основе многовыходного дискриминатора (Multi-Exit Discriminator— MED) и локального предпочтения (Local Preference), а также благодаря наличию функции фильтрации в виде списков доступа он предоставляет администратору сети возможность разнообразить политику маршрутизации.

Совет

Прежде чем приступать к реализации в рамках протокола BGP той или иной политики маршрутизации с помощью многовыходных дискриминаторов, локальных предпочтений и других атрибутов, убедитесь, что точно можете предвидеть последствия этих модификаций. Рекомендуется просмотреть 2-е издание книги *принципы маршрутизации в Internet* и подготовленный компанией Cisco Systems аналитический материал *Using the Border Gateway Protocol for Interdomain Routing (Применение протокола пограничной маршрутизации для маршрутизации между доменами).* Аналитический материал можно найти на странице сертифицированных пользователей устройств компании Cisco по адресу www.Cisco.com/univercd/cc/td/doc/cisintwk/ics/icsbgp4.htm.

Если сеть имеет несколько соединений с провайдером Internet-услуг, обычно исполняется протокол BGP, который позволяет выбрать оптимальный путь к внешним сетям. Использовать протокол BGP, если существует только одно соединение с провайдером Internet-услуг, в общем случае не требуется, так как все пути к внешним сетям проходят через этого провайдера. Однако некоторым провайдерам нравится обмениваться по протоколу BGP, чтобы знать путь к сетям своего клиента и обеспечить сетевые маршруты для конфигурирования маршрутов по умолчанию.

Рассмотрим конфигурирование протокола BGP на маршрутизаторах компании ZIP SF-Core-1 и Seoul-1, каждый из которых соединен с глобальной сетью Internet через провайдера Internet-услуг. Конфигурирование процесса BGP-маршрутизации состоит из трех этапов: разрешения маршрутизатору исполнять протокол BOP, идентификации одноранговых маршрутизаторов и задания сетевых адресов, которые подлежат объявлению одноранговым маршрутизаторам.

Для выдачи маршрутизатору разрешения на исполнение протокола BGP используется команда глобального конфигурирования ОС IOS router bgp. Эта команда имеет в качестве параметра целое число, которое представляет собой номер автономной системы (ASN), назначаемый сети одним из реестров (RIPE, APNIC или ARIN). Каждой автономной системе, подключающейся к глобальной сети Internet, одним из реестров должен присваиваться уникальный номер ASN, чтобы не допустить случайного дублирования. Дублирование номера ASN может привести к тому, что сеть не будет объявляться из-за обнаружения ложной петли маршрутизации. Если протокол BGP исполняется в частной сети, которая не подключается к глобальной сети Internet, то выбор номера ASN должен осуществляться из блока частных номеров ASN с диапазоном значений от 32768 до 64511.

Примечание

Многие сетевые администраторы используют номер ASN в качестве идентификатора процесса для других протоколов динамической маршрутизации, например EIGRP. Сеть компании ZIP следует этой традиции.

Идентификация одноранговых маршрутизаторов выполняется с помощью использования субкоманды конфигурирования маршрутизации ОС IOS neighbor remote-as. Эта команда имеет два параметра: IP-адрес соседнего маршрутизатора и номер ASN. Если номер ASN, заданный после слова remote-as, отличается от номера ASN, заданного в команде глобального конфигурирования router bgp, то этот сосед считается внешним одноранговым BGP-узлом (EBGP). Обычно IP-адрес соседнего маршрутизатора, являющегося одноранговым EBGP-узлом, представляет собой адрес на непосредственно подключенном сетевом интерфейсе.

Если номер ASN, заданный после слова remote-as, совпадает с номером ASN, заданным в команде глобального конфигурирования router bgp, то этот сосед считается внутренним одноранговым BGP-узлом (IBGP). IP-адресом соседнего маршрутизатора, являющегося одноранговым IBGP-узлом, может быть любой достоверный и достижимый IP-адрес такого одноранговой узла. Одноранговые IBGP-узлы могут размещаться как на непосредственно подключенном сетевом интерфейсе (как при нескольких соединейиях с провайдером Internet-

услуг в одном физическом месте), так и в сети, не имеющей прямого подключения, а подсоединенной к удаленному маршрутизатору, входящему в состав автономной системы (как при нескольких соединениях с провайдером Internet-услуг в разных физических местах).

Поскольку IP-адреса одноранговых IBGP-узлов не обязательно находятся на непосредственно подключенном сетевом интерфейсе, то при организации взаимодействия IBGP-узлов рекомендуется использовать в качестве адреса источника и получателя адрес интерфейса кольца обратной связи. Интерфейс кольца обратной связи не связывается с каким-либо физическим интерфейсом, поэтому он всегда находится в рабочем состоянии и достижим, пока существует путь к связанному с ним IP-адресу через IGP-маршрутизацию или статические маршруты. Чтобы сконфигурировать интерфейс кольца обратной связи в качестве IP-адреса источника для организации взаимодействия одноранговых IBGP-узлов, используется субкоманда конфигурирования маршрутизации ОС IOS neighbor вместе с ключевым словом update-sourоe. За ключевым словом update-source должны следовать имя и номер соответствующим образом адресованного и сконфигурированного интерфейса кольца обратной связи конфигурируемого маршрутизатора.

Если маршрутизатор имеет много соседних взаимодействующих с ним одноранговых BGP-узлов, часто бывает трудно запомнить, какой IP-адрес и какой номер ASN принадлежат какому одноранговому узлу. Применяя в качестве параметра субкоманды конфигурирования маршрутизации ОС IOS neighbor ключевое слово description, могут быть введены комментарии, помогающие в предоставлении этой информации сетевому администратору.

Идентификация сетей автономной системы, которые будут объявляться одноранговым EBGP-узлам, выполняется с помощью субкоманды конфигурирования маршрутизации ОС IOS network. Эта команда воспринимает в качестве параметра адрес сети, подлежащий объявлению одноранговым маршрутизаторам, и имеет необязательное ключевое слово mask, после которого указывается сетевая маска для этого адреса. Если сетевая маска не указывается, то предполагается классовый адрес сети. Благодаря использованию сетевой маски протокол BGP позволяет объявлять одноранговым маршрутизаторам как подсети, так и CIDR-блоки. Обмен полученной через EBGP-узлы информацией о сетях из других автономных систем будет осуществляться между одноранговыми IGBP-узлами внутри автономной системы.

Примечание

Следует помнить, что BGP-маршрутизатор объявляет полученные от однорангового BGP-узла маршруты всем своим одноранговым BGP-узлам. Например, маршруты, данные о которых были получены от EBGP-узла, принадлежащего одному провайдеру Internet-услуг, будут объявляться IBGP-узлам, которые, в свою очередь, оповестят о них других провайдеров Internet-услуг с помощью механизма EBGP-обмена. Переобъявляя информацию о маршрутах, сеть может стать транзитной сетью между провайдерами, к которым она подключена. Подобный результат может расстроить провайдеров, а также вызвать массовые перегрузки сети. Если образование таких транзитных сетей нежелательно, следует воспользоваться субкомандами фильтрации distribute-lists и route-maps. Эти субкоманды позволяют управлять переобъявлением поступающих извне маршрутов. Списки рассылки (distribute lists) более подробно обсуждаются в следующем разделе.

Наконец, поскольку сеть компании ZIP не будет передавать данные между провайдерами IPS-A и IPS-B, и BGP-маршруты не будут редистрибутироваться в процесс IGP-маршрутизации, то BGP-синхронизация будет отключена субкомандой конфигурирования маршрутов ОС IOS no synchronization. При разрешении синхронизации маршрут не будет объявляться одноранговому EBGP-узлу, если он не появится в первичной таблице выбора маршрутов для однорангового узла, и информация о нем не поступит через процесс IGP-маршрутизации. Поскольку сеть компании ZIP хочет объявлять маршруты только для своей собственной автономной системы, то запрещение синхронизации приведет к уменьшению времени сходимости протокола BGP.

Ниже приведен пример конфигурирования BGP-маршрутизации на маршрутизаторе компании ZIP SF-Core-1 таким образом, что он будет объявлять сеть 131.108.0.0 провайдеру Internet-сервиса

через EBGP-процесс. Маршрутизатор SF-Core-I имеет номер ASN 25000. У провайдера номер ASN равен 1 и IP-адрес однорангового узла — 192.7.2.1. Дополнительно в качестве однорангового IBGP-узла для маршрутизатора SF-Core-1 конфигурируется маршрутизатор Seoul-1 с IP-адресом однорангового узла 131.108.254.6. При этом в качестве адреса источника для соединения одноранговых узлов используется IP-адрес интерфейса кольца обратной связи loopback 0.

```
SF-Core-l#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-Core-1(config)#router bgp 25000
SF-Core-1(config-router)#no synchronization
SF-Core-1(config-router)#network 131.108.0.0
SF-Core-1(config-router)#neighbor 192.7.2.1 remote-as 1
SF-Core-1(config-router)#neighbor 192.7.2.1 description Internet Connection to ISP-B
SF-Core-1(config-router)#neighbor 131.108.254.6 remote-as 25000
SF-Core-1(config-router)#neighbor 131.108.254.6 description IBGP to Seoul-1
SF-Core-1(config-router)#neighbor 131.108.254.6 update-source loopback 0
SF-Core-1(config-router)#^Z
```

Ниже приведен пример конфигурирования BGP-маршрутизации уже на маршрутизаторе компании ZIP Seoul-1, когда он будет объявлять сеть 131.108.0.0 своему провайдеру Internet-услуг через EBGP-процесс. Маршрутизатор Seoul-1 имеет номер ASN 25000. У провайдера номер ASN равен 701 и IP-адрес однорангового узла — 211.21.2.1. Дополнительно в качестве однорангового IBGP-узла для маршрутизатора Seoul-1 конфигурируется маршрутизатор SF-Core-1 с IP-адресом однорангового узла 131.108.254.3. При этом в качестве адреса источника для соединения одноранговых узлов используется IP-адрес интерфейса кольца обратной связи loopback 0:

```
Seoul-l# configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul-1(config)#router bgp 25000
Seoul-1(config-router)#no synchronization
Seoul-1(config-router)#network 131.108.0.0
Seoul-1(config-router)#neighbor 211.21.2.1 remote-as 701
Seoul-1(config-router)#neighbor 211.21.2.1 description Internet Connection to ISP-A
Seoul-1(config-router)#neighbor 131.108.254.3 remote-as 25000
Seoul-1(config-router)#neighbor 131.108.254.3 description IBGP to SF-Core-1
Seoul-1(config-router)#neighbor 131.108.254.3 update-source loopback 0
Seoul-1(config-router)#^Z
```

После того как оба маршрутизатора будут сконфигурированы на работу с протоколом BGP и определятся одноранговые узлы, маршрут сети 131.108.0.0 объявляется провайдерам ISP-A и ISP-B маршрутизаторами Seoul-1 и SF-Core-1, соответственно. Используя команды режима EXEC ОС IOS, описываемые в следующем разделе "Просмотр информации протоколов динамической маршрутизации", администратор сети может проверить задание одноранговых узлов и правильность объявления и приема информации о сетевых маршрутах.

Когда одноранговые IBGP-узлы обмениваются маршрутной информацией, полученной от одноранговых EBGP-узлов, важно запомнить, что IBGP-узел должен иметь маршрут до узла следующего перехода маршрута, информация о котором поступила от EBGP-узла. Например, если маршрутизатор SF-Core-1 узнает о маршруте 140.222.0.0/16 от провайдера Internet-сервиса ISP-B, то адресом узла следующего перехода будет для него адрес 192.7 .2.1. Когда этот маршрут переобъявляется одноранговому IBGP-узлу Seoul-1, он не может быть занесен в BGP-таблицу маршрутов маршрутизатора Seoul-1, если тот не имеет маршрута до адреса узла следующею перехода 192.7.2.1. Не будучи занесенным в BGP-таблицу маршрутизатора Seoul-1, маршрут не сможет быть выбранным в качестве лучшего и занесенным в первичную таблицу выбора маршрутов. Не сможет он также и оцениваться в сравнении с таким же маршрутом, информация о котором могла бы поступить от провайдера ISP-A. Если адрес узла следующего перехода не входит в диапазон сетевых адресов, для которых ваш IGP-протокол обеспечивает

маршрутную информацию (например, адреса, назначаемые провайдером Internet-услуг), то для объявления в вашем процессе IGP-марщрутизации непосредственно подключенных или статических маршрутов с такими адресами используйте команду redistribute, описываемую в следующем разделе.

Управление информацией протоколов динамической маршрутизации

Сетевые администраторы часто используют политику административного регулирования, чтобы управлять потоком информации о сетевых маршрутах как внутри, так и снаружи своих сетей. Такая политика включает задание маршрутизаторов, участвующих в процессе маршрутизации, определение того, передается ли информация о подсетях

Между различными основными пространствами сетевых адресов, и того, какие маршруты должны коллективно использоваться маршрутизаторами. Реализуя ту или иную политику, можно управлять трафиком доступа к сети и ее защитой. В данном разделе исследуются пять популярных команд IOS, которые используются для управления протоколами динамической маршрутизации и реализации политики маршрутизации.

Одним из наиболее важных атрибутов управления динамическими протоколами маршрутизации является возможность разрешить или запретить распространять в сеть информацию о сетевых маршрутах от отдельно взятого маршрутизатора. Такая возможность фильтрации маршрутной информации позволяет ограничивать доступ к одной части сети из другой ее части. Для протокола BOP ограничение распространения и переобъявления информации о маршрутах в плане маршрутов для одноранговых узлов предотвращает в автономной системе непреднамеренный транзит пакетов между несколькими провайдерами Internet-услуг.

Основным средством фильтрации маршрутной информации является субкоманда конфигурирования маршрутизации ОС IOS distribute-list. Возможности по фильтрации команды distribute-list реализуются через использование стандартных списков IP-доступа. Списки доступа представляют собой общий инструментарий для задания критериев фильтрации. Совместно с субкомандами протоколов маршрутизации, списки доступа могут задавать разрешенные и запрещенные маршруты. Подробно списки доступа обсуждаются в разделе "Конфигурирование IP-фильтрации с помощью списков доступа". Команда distribute-list использует список доступа в конкретной ситуации управления распространением информации о маршрутах.

Эта команда имеет несколько параметров, включая имя или номер списка IP-доступа, ключевые слова in или out, управляющие направлением, в котором производится фильтрация, и необязательный идентификатор интерфейса. Этот идентификатор указывает, что должна производиться фильтрация только тех пакетов актуализации маршрутной информации, которые предназначены для конкретного интерфейса. Если идентификатор не указывается, а просто опускается, то список распространения (distribute list) применяется в отношении всех пакетов актуализации маршрутной информации, которые подпадают под действие списка доступа.

Ниже показан пример применения команды distribute-list на маршрутизаторе SF-Core-1 для запрещения передачи резервного сетевого адреса 10.0.0.0 в процессе BGP-маршрутизации и для разрешения передачи всех других адресов.

```
SF-Core-l# configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-Core-1(config)#router bgp 25000
SF-Core-1(config-router)#distribute-list 1 in
SF-Core-1(config-router)#access-list 1 deny 10.0.0.0 0.0.0.0
SF-Core-1(config)#access-list 1 permit any
SF-Core-1(config)#^Z
```

Примечание
Из-за лавинной природы адресации пакетов с информацией о состоянии каналов фильтрация входящей маршрутной информации в протоколах, основанных на методе учета состояния каналов, например OSPF или IS-IS, невозможна. Фильтрация входящей маршрутной информации осуществляется только в отношении внешних маршрутов.

Если команда distribute-list используется в качестве субкоманды процесса маршрутизации, фильтрация, определенная в ней, производится в отношении всех источников пакетов актуализации маршрутной информации. Во многих ситуациях желательно производить фильтрацию только одного источника маршрутной информации, например, конкретного однорангового BGP-узла. Фильтрация пакетов актуализации, поступающих и выходящих из конкретных одноранговых BGP-узлов, может осуществляться путем использования команды distribute-list в отношении нужного BGP-соседа в качестве опционного ключевого слова субкоманды конфигурирования протокола BGP neighbor.
Ниже приведен предыдущий пример, но переписанный так, что теперь команда distribute-list используется на маршрутизаторе SF-Core-1 в качестве опции субкоманды neighbor для того, чтобы только одноранговый EBGP-узел не мог получать информацию о резервном сетевом адресе 10.0.0.0.

```
SF-Core-1(config)#router bgp 25000
SF-Core-1(config-router)#neighbor 192.7.2.1 distribute-list 1 in
SF-Core-1(config-router)#access-list 1 deny 10.0.0.0 0.0.0.0
SF-Core-1(config)#aceess-list 1 permit any
SF-Core-1(config)#^Z
```

Иногда может понадобиться, чтобы маршрутизатор слушал обновления маршрутной информации на конкретном интерфейсе, но не объявлял ее другим маршрутизаторам, подключенным к этому интерфейсу. В тех случаях, когда желательная такая конфигурация, говорят, что интерфейс работает в пассивном режиме. Устанавливает пассивный режим субкоманда конфигурирования маршрутизации ОС IOS passive-interface. Параметром этой команды является идентификатор интерфейса, на котором подавляются выходящие пакеты актуализации маршрутной информации. Ниже приведен пример конфигурирования маршрутизатора компании ZIP San-Jose с помощью команды passive-interface таким образом, чтобы пакеты актуализации маршрутной информации не посылались в интерфейс маршрутизатора Token Ring.

```
San-Jose#configure
Configuring  from terminal,  memory,   or network   [terminal]?
Enter configuration commands,   one per line.   End with CNTL/Z.
San-Jose(config)#router eigrp 125000
San-Jose(config-router)#passive-interface tokenring 1/0
San-Jose(config-router)#^Z
```

Может возникнуть необходимость и в том, чтобы сконфигурировать на маршрутизаторе перечень конкретных соседних маршрутизаторов, с которыми он может обмениваться информацией динамической маршрутизации. Например, чтобы реализовать протокол OSPF в среде, не допускающей широковещания, для его правильной работы необходимо задать конкретные соседние маршрутизаторы. В качестве другого примера можно привести задачу организации среды с более высоким уровнем защиты, в которой только заданным маршрутизаторам-соседям разрешено обмениваться маршрутной информацией двухточечным образом.
Для задания IP-адреса соседнего маршрутизатора, с которым разрешен обмен маршрутной информацией, используется субкоманда конфигурирования маршрутизации ОС IOS neighbor. Если использовать одновременно и команду passive-interface, то обмен маршрутной информацией осуществляется только с заданными соседями и путем двухточечного (не широковещательного) обмена. В качестве параметра команды neighbor выступает IP-адрес соседнего маршрутизатора. Ниже приведен пример конфигурирования в маршрутизаторе

компании ZIP Seoul-2 двухточечного обмена маршрутной информацией с работающим под управлением операционной системы UNIX сервером, на котором исполняется протокол RIP в сегменте, подключенном к интерфейсу Ethernet. Команда passive-interface используется для того, чтобы запретить протоколу RIP делать объявления на последовательном интерфейсе.

```
Seoul-2# configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul-2(config)#router rip
Seoul-2(config-router)#passive-interface serial 0
Seoul-2(config-router)#passive-interface ethernet 0
Seoul-2(config-router)#neighbor 131.108.3.40
Seoul-2(config-router)#^Z
```

Иногда возникают ситуации, когда устройствам, работающим под управлением ОС IOS компании Cisco, необходимо обмениваться маршрутной информацией с другими устройствами, которые не поддерживают протокол маршрутизации, выбранный для сети. Например, в сети компании ZIP исполняется протокол EIGRP. Устройства на UNIX-платформе не могут принимать пакеты актуализации маршрутной информации протокола ЕЮ PR, поскольку они работают только с протоколом RIP. Чтобы справиться с такой ситуацией, ОС IOS имеет возможность передавать маршрутную информацию из одного протокола маршрутизации в другой. Этот процесс называется редистрибуцией маршрутов.

Чтобы разрешить редистрибуцию маршрутов, используется субкоманда конфигурирования маршрутизации ОС IOS redistribute. В качестве параметра этой команды выступает имя процесса маршрутизации, из которого необходимо осуществлять редистрибуцию. Вместо имени процесса маршрутизации также могут указываться ключевые слова static или connected. Использование слова static позволяет осуществлять объявление в процессе маршрутизации сконфигурированных вручную статических маршрутов. Ключевое слово connected позволяет процессу маршрутизации объявлять маршруты для непосредственно подключенных интерфейсов, которые не согласуются с адресом, заданным в субкоманде маршрутизации network. Поскольку каждый протокол динамической маршрутизации использует разные методы вычисления метрики, автоматическое преобразование метрики может оказаться невозможным. Ниже дан перечень поддерживаемых ОС IOS случаев автоматического преобразования метрики.

- Протокол RIP может автоматически редистрибутировать статические маршруты, присваивая им значение метрики 1 (непосредственно подключенный).
- Протокол IGRP может автоматически редистрибутировать статические маршруты и информацию из других автономных систем, использующих протокол IGRP. Этот протокол назначает статическим маршрутам метрику, которая идентифицирует их как непосредственно подключенные. Протокол IGRP не изменяет метрику маршрутов, полученных из IGRP-пакетов актуализации от других автономных систем.
- Любой протокол может редистрибутировать информацию из других протоколов маршрутизации, если установлено значение метрики по умолчанию.

Метрика по умолчанию задается субкомандой конфигурирования маршрутизации ОС IOS default-metric. Эта команда имеет аргументом один или несколько атрибутов метрики протокола маршрутизации, что зависит от конкретного конфигурируемого протокола маршрутизации. Ниже показан пример конфигурирования редистрибуции информации протокола EIGRP в протокол R1P на маршрутизаторе компании ZIP с именем Singapore. Заметим, что команда passive-interface используется для запрещения объявления информации протокола RIP на последовательном интерфейсе и что значение метрики по умолчанию устанавливается равным 3.

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Singapore(config)#router rip
Singapore(config-router)#tdefault-metric 3
Singapore(config-router)#redistribute eigrp 25000
```

```
Singapore(config-router)#passive-interface serial 0
Singapore(config-router)#^Z
]
```

Совет

Редистрибуция маршрутной информации из одного протокола в другой может оказаться коварной. Взаимная редистрибуция, при которой маршруты передаются из одного протокола в другой и наоборот, может привести к образованию петель маршрутизации, так как разумные методы проверки редистрибутируемых маршрутов отсутствуют. По возможности следует избегать взаимной редистрибуции. Если же она абсолютно необходима, то должны использоваться команды passive-interface и distribute-list, чтобы ограничить объявление конкретных маршрутов в конкретные протоколы маршрутизации.

Как уже обсуждалось ранее, при передаче маршрутной информации из одного адреса основной сети в другой протоколы маршрутизации класса IGP, поддерживающие маски подсетей переменной длины, автоматически суммируют подсети в единый классовый маршрут сети. Например, подсети сети компании ZIP 131.108.0.0 не объявляются в адресном пространстве 172.16.0.0 другого маршрутизатора, исполняющего протокол EIGRP. Если существуют подсети адресного пространства 131.108.0.0, которые подключаются за пределами сети 172.16.0.0, т.е. сеть 131.108.0.0 разрывна, то может оказаться необходимым распространять информацию о подсетях одной части сети 131.108.0.0 через сеть 172.16.0.0 в другой части сети 131.108.0.0. Понятно, что в такой ситуации суммирование нежелательно. Субкоманда конфигурирования маршрутизации ОС IOS no auto-summari предотвращает автоматическое суммирование адресов на границах класса сети \ разрешает передачу информации о подсетях.
Ниже показан пример выполнения деконфигурирования автосуммирования на маршрутизаторе компании ZIP SF-Core-1:

```
SF-Core-l#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-Core-1(config)#router eigrp 25000
SF-Core-1(config-router)#no auto-summary
SF-Core-1(config-router)#^Z
```

Просмотр информации протоколов динамической маршрутизации

Работа и конфигурация протоколов динамической маршрутизации могут проверяться с помощью команд режима EXEC ОС IOS. Эти команды подразделяются на две категории: протоколо-независимые и протоколо-зависимые. Давайте сначала рассмотрим протоколо-независимые команды.
Как уже показывалось в разделе "Конфигурирование протоколов IP-маршрутизации", для того чтобы выяснить, являются ли источниками информации о маршрутах какие-либо протоколы динамической маршрутизации и определить атрибуты таких маршрутов, может использоваться команда режима EXEC ОС IOS show ip route.
Выяснение того, какие протоколы маршрутизации исполняются, и определение различных атрибутов таких протоколов выполняется с помощью команды ОС IOS режима EXEC show ip protocols. Эта команда может иметь в качестве параметра ключевое слово summary. Вариант команды с ключевым словом summary выводит на экран только список имен протоколов маршрутизации и идентификаторы процесса, если таковые имеются. Ниже показан пример результата исполнения команды show ip protocols summary на маршрутизаторе компании ZIP SF-Core-1:

```
SF-Core-l#show ip protocols  summary
Index       Process Name
0           connected
1           static
```

```
2           eigrp 25000
3           bgp 25000
```

Стандартный вариант команды show ip protocols выводит перечень исполняемых протоколов маршрутизации и многочисленные атрибуты этих протоколов, включая источники пакетов актуализации маршрутной информации, используемые в списках распространения фильтры, метрическую информацию и данные об объявляемых сетях. Ниже приводится пример информации, выводимой командой show ip protocols при ее исполнении на маршрутизаторе компании ZIP SF-Core-1, работающем с протоколами EIGRP и BOP:

```
SF-Core-l#show ip protocols
   Routing Protocol is "eigrp 25000 "
   Outgoing update filter list for all interfaces is not set
   Incoming update filter list for all interfaces is not set
   Default networks flagged in outgoing updates
   Default networks accepted from incoming updates
   EIGRP metric weight K1=1,K2=0,K3=1,K4=0,K5=0
   EIGRP maximum hopcount 100
   EIGRP maximum metric variance 1
   Redistributing:connected,eigrp 1
   Automatic network summarization is not in effect
   Routing for Networks:
     131.108.0.0
   Routing Information Sources:
     Gateway       Distance       Last Update
     131.108.20.1        90       00:04:13
     131.108.20.2        90       00:04:13
     131.108.20.4        90       00:04:13
     Distance:internal 90 external 170
Routing Protocol is "bgp 25000 "
   Sending updates every 60 seconds,next due in 0 seconds
   Outgoing update filter list for all interfaces is 2
   Incoming update filter list for all interfaces is 1
   IGP synchronization is disabled
   Automatic route summarization is enabled
   Neighbor(s):
     Address   Filtln FiltOut Distln DistOut Weight RouteMap
     192.7.2.1               150
   Routing for Networks:
     131.108.0.0
   Routing Information Sources:
     Gateway       Distance          Last Update
     (this router)       200          Iw5d
     192.7.2.1            20          Iw3d
Distance: external 20 internal 200 local 200
```

В сложных протоколах маршрутизации, например EIGRP, OSPF и BGP, обеспечивается доступ ко многим атрибутам, таблицам и базам данных с информацией относительно их работы, конфигурации и топологии. В табл. 4.3, 4.4 и 4.5 показаны общие команды режима EXEC ОС IOS, которые используются для просмотра информации, относящейся к протоколам EIGRP, OSPF и BOP, соответственно.

Таблица 4.3. Команды ОС IOS режима EXEC для протокола EIGRP

Команды ОС IOS режима EXEC для протокола EIGRP	**Функция**
show ip eigrp interfaces	Выводит на экран информацию об интерфейсах, сконфигурированных для работы с протоколом IP EIGRP
show ip eigrp neighbors	Выводит на экран данные о соседях, обнаруженных протоколом

	IP EIGRP
show ip eigrp topology	Выводит на экран таблицу топологии протокола IP EIGRP
show ip eigrp traffic	Выводит на экран значение количества пакетов, отправленных и полученных процессом (процессами) протокола IP EIGRP

Таблица 4.4. Команды ОС IOS режима EXEC для протокола OSPF

Команды ОС IOS режима EXEC для протокола OSPF	Функция
show ip ospf	Показывает общую информацию о процессах OSPF-маршрутизации
show ip ospf database	Выдает перечень информации, связанной с базой данных протокола OSPF
show ip ospf database router	Показывает информацию базы данных протокола OSPF о связях маршрутизатора
show ip ospf database network	показывает информацию базы данных протокола OSPF о связях сети
show ip ospf database external	Показывает информацию базы данных протокола OSPF о внешних связях сети
show ip ospf database database-summary	Показывает сводную информацию относительно базы данных протокола OSPF
show ip ospf border-routers	Выводит записи внутренней таблицы маршрутизации протокола OSPF, находящиеся в столбцах "Маршрутизаторы границы области" (Area Border Routers или ABR) и "Маршрутизаторы границы автономной системы" (Autonomous System Boundary Routers—ASBR)
show ip ospf interface	Показывает связанную с протоколом OSPF информацию для конкретного интерфейса
show ip ospf neighbor	Показывает информацию об OSPF-соседях

Таблица 4.5. Команды ОС IOS режима EXEC для протокола BGP

Команды ОС IOS режима EXEC для протокола BGP	**Функция**
show ip bgp cidr-only	Показывает все BGP-маршруты, которые содержат сетевые маски подсетей и суперсетей
show ip bgp filter-list *номер списка доступа*	Показывает маршруты, которые входят в заданный список доступа путей автономных систем
show ip bgp regexp *регулярное выражение*	Показывает маршруты, которые удовлетворяют критерию регулярного выражения, введенного в командной строке
show ip bgp [network] [network-mask] [subnets]	Показывает содержимое таблицы BGP-маршрутизации
show ip bgp neighbors	Выдает подробную информацию о TCP-и BGP-соединениях с отдельными соседями
show ip bgp nieghbors *[адрес]* routes	Показывает маршруты, информация о которых поступила от конкретного BGP-соседа
show ip bgp bgp nieghbors *[адрес]* advertised	Показывает маршруты, объявляемые конкретному BGP-соседу

show ip bgp bgp nieghbors *[адрес]* paths	Показывает пути, информация о которых поступила от конкретного BGP-соседа
show ip bgp paths	Показывает все пути, содержащиеся в базе данных протокола BGP
show ip bgp summary	Показывает статус всех соединений с одноранговыми BGP-узлами

Конфигурирование IP-фильтрации с помощью списков доступа

С того самого времени, когда несколько систем были соединены вместе, образуя сеть, существовала необходимость в ограничении доступа к некоторым системам или частям сети по соображениям защиты, конфиденциальности данных или по другим причинам. Используя средства фильтрации пакетов ОС IOS компании Cisco, администратор сети может ограничить доступ к определенным системам, сегментам сети, диапазонам адресов и службам на основе разнообразных критериев. Важность функции ограничения доступа особенно возрастает, когда сеть компании начинает подключаться к внешним сетям, например, к сетям компаний-партнеров или к глобальной сети Internet.

Функции по фильтрации пакетов, реализованные в списках IP-доступа ОС IOS, позволяют ограничивать потоки пакетов на основе следующих критериев.

- IP-адрес источника.
- IP-адрес источника и пункта назначения.
- Тип IP-протокола, включая протоколы TCP (протокол управления передачей данных), UDP (протокол дейтаграмм пользователя) и ICMP (протокол управления сообщениями в сети Internet).
- Службы протокола TCP у источника или в пункте назначения, например, фильтруется служба отправки почтовых сообщений sendmail или служба Telnet.
- Службы протокола UDP у источника или в пункте назначения, например, фильтруются дейтаграммы службы начальной загрузки протокола bootp или службы NetBIOS.
- Службы протокола ICMP, например, фильтруются эхо-ответы и сигналы недостижимости порта протокола ICMP.

Приведенный выше список ни в коей мере не является исчерпывающим. Гибкость списков IP-доступа обеспечивает администратору сети свободу выбора объекта фильтрации.

Ключом к пониманию роли списков доступа в ОС IOS является четкое представление о том, что задача фильтрации пакетов разбивается на два разных этапа. На первом с помощью команд access-list и ip access-list задаются критерии фильтрации. На втором эти критерии фильтрации накладываются на желаемые интерфейсы. Один метод наложения фильтрации с помощью списков доступа уже рассматривался. Это их применение совместно с командой distribute-list для фильтрации маршрутной информации. Е следующих разделах упор делается на применение списков доступа совместно с командор ip access-group. Но сначала давайте рассмотрим установку критериев фильтрации.

Задание списка доступа

Критерии фильтрации задаются в списке операторов разрешения и запрета, называемом *списком доступа.* Строки списка доступа сравниваются с IP-адресами *t* Другой информацией пакета данных последовательно в том порядке, в котором были заданы, пока не будет найдено совпадение. При совпадении осуществляется выход из списка. При этом работа списка доступа напрямую зависит от порядка следования строк.

В первоначально разработанном варианте ОС IOS имела только одну команду для создания списков доступа — команду access-list. Используя эту команду и число из соответствующего диапазона значений, администратор сети мог задавать сетевой протокол, для которого создавался список доступа. Например, диапазон значений от 1 до 99 обозначал стандартный список IP-доступа, а диапазон значений от 900 до 999 обозначал фильтр для IPX-пакетов. (Списки IPX-доступа обсуждаются в главе 6, "Основы IPX".)

Для увеличения гибкости и количества списков доступа разработчики ОС IOS создали версии команды access-list для протоколов IP и IPX, которые позволяют формировать именованные списки доступа. Другими словами, новые команды могут использовать для идентификации списка доступа произвольную цепочку символов, а не только число. Командой для создания именованного списка IP-доступа является команда ip access-list. (Для именованных IPX-списков также предназначена команда ipx access-list.)

И нумерованные, и именованные списки IP-доступа подразделяются на две категории: стандартные и расширенные. Стандартный список IP-доступа выполняет сравнение только с IP-адресом источника пакета, тогда как расширенный список доступа может осуществлять сравнение с IP-адресами источника и пункта назначения, типом IP-протокола, портами источника и пункта назначения транспортного уровня.

Для создания нумерованного списка доступа используется команда глобального конфигурирования ОС IOS access-list. Как отмечалось ранее, параметром команды access-list является номер списка. Стандартным спискам IP-доступа присваиваются номера из диапазона 1-99. Расширенные списки IP-доступа обозначаются числом, лежащим в диапазоне 100-199. После номера списка на каждой строке списка доступа следует ключевое слово permit (разрешить) или deny (запретить), за которым указываются IP-адрес, подстановочная маска, протокол и номер порта фильтруемого протокола. Ниже приведен пример нумерованного стандартного списка IP-доступа на маршрутизаторе компании ZIP SF-1, который запрещает пакеты с IP-адресом источника 131.108.101.99, но разрешает остальные пакеты из сети 131.108 .101. 0/24.

```
SF-1#configure
Configuring from terminal,  memory,   or network   [terminal]?
Enter configuration commands,   one per line.   End with CNTL/Z.
SF-l(config)#accese-list 1 deny 131.108.101.99
SF-l(config)#access-list 1 permit 131.108.101.0  0.0.0.255
SF-l(config) #^Z
```

Порядок строк в списке доступа определяет его работу. В предыдущем примере перестановка операторов в списке доступа в обратном порядке полностью изменит его функционирование. Ниже показан вид такого списка доступа.

```
SF-1#configure
Configuring  from terminal,   memory,   or network   [terminal]?
Enter configuration commands,   one per line.   End with CNTL/Z.
SF-1(config)#access-list 1 permit 131.108.101.0  0.0.0.255
SF-1(config)#access-list 1  deny 131.108.101.99
SF-1(config)#^Z
```

Теперь, если пакет с IP-адресом 131.108.101.99 сравнивается с этим списком доступа, то он удовлетворяет критерию первого оператора и выходит из списка. Сравнение по оператору deny для адреса 131.108.101. 99 никогда выполняться не будет.

Совет

В списках доступа используется концепция подстановочной, или безразличной, маски. Подстановочная маска отличается тем, что в ней позиции битов, установленных в значение 1, подходят для любого адреса. Подстановочная маска 0.0.0.255 согласуется с любым числом в диапазоне от 0 до 255, которое стоит в четвертом октете IP-адреса. Подстановочная маска 0.0.3.255, если пересчитывать из двоичной системы, совпадает с любым IP-адресом, имеющим в третьем октете число 0, 1, 2 или 3 и любое число в четвертом октете. Подстановочная маска позволяет администратору сети задавать диапазоны адресов, лежащие на границах значений в двоичном исчислении.

Ниже приведен пример нумерованного расширенного списка IP-доступа для маршрутизатора компании ZIP SF-1, который разрешает достигать IP-адреса 131.108.101.99 только пакетам

протокола TCP Simple Mail Transfer Protocol (SMTP) (упрощенного протокола электронной почты) и службы имен доменов (DNS) протокола UDP. Заметим, что ключевое слово any может заменять собой сетевой адрес 0.0.0.0 с подстановочной маской 255.255.255.255.

```
SF-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-1(config)#access-list 100 permit tcp any host 131.108.101.99 eq smtp
SF-1(config)#access-list 100 permit udp any host 131.108.101.99 eq domain
SF-1(config)#access-list 100 deny ip any any log
SF-1(config)#^Z
```

Совет

Все списки доступа неявно имеют в конце оператор deny. Это означает, что любой пакет, который не удовлетворяет критериям фильтрации одной из строк списка доступа, запрещается. Чтобы облегчить устранение неполадок и административный контроль средств защиты, рекомендуется ставить оператор deny в конце списка в явном виде, сопровождая его опционным ключевым словом log. Такие действия приведут к тому, что все пакеты, которые не удовлетворяют критериям списка, будут заноситься в журнал нарушений консоли или, если активирована функция ведения журнала системы, будут попадать на сервер системного журнала. (Занесение в журнал более подробно обсуждается в главе 7.) Ключевое слово log может также ставиться в конце любой строки списка доступа, для которой администратор сети хочет иметь запротоколированные журнальные записи.

До сих пор рассматривались лишь примеры нумерованных списков доступа. Как отмечалось ранее, именованные списки доступа позволяют администратору использовать при ссылке на список IP-доступа произвольную цепочку символов. Например, можно давать спискам доступа легко запоминаемые имена, и название может относиться к выполняемой им задаче фильтрации.

Именованные списки IP-доступа создаются командой конфигурирования ip access-list. Эта команда в качестве параметров имеет ключевое слово **extended** (расширенный) или standard (стандартный), которым обозначается тип создаваемого именного списка доступа, и фактическое имя этого списка доступа.

Команда ip access-list вызывает переход ОС IOS в подрежим конфигурирования списков доступа. После перехода в подрежим конфигурирования списков доступа следует вводить только операторы permit и deny вместе с другими критериями фильтрации. Имя списка доступа повторять в каждой строке списка не надо. Преобразуем предыдущий пример стандартного нумерованного списка доступа, используя режим создания именованного списка.

```
SF-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-1(config)lip access-list standard sorrycharlie
SF-1(config-std-nacl)#deny 131.108.101.99
SF-l(config-std-nacl)#permit 131.108.101.0 0.0.0.255
SF-1(config)#^Z
```

Ниже показан предыдущий пример расширенного списка доступа, переписанный с применением режима создания именованных списков доступа.

```
SF-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-1(config)#ip access-list extended out-of-luck
SF-1(config-ext-nacl)#permit tcp any host 131.108.101.99 eq smtp
SF-1(config-ext-nacl)#permit udp any host 131.108.101.99 eq domain
SF-1(config-ext-nacl)#deny ip any any log
```

```
SF-1(config-ext-nacl)#^Z
```

И для нумерованных, и для именных списков важно знать, почему разрешался или запрещался доступ к определенным хост-машинам, сетям или службам.
В ранних версиях ОС IOS единственным способом сохранения информации о списках доступа (или о любой команде конфигурирования) было введение комментариев в копию конфигурационного файла запуска, который хранился на сервере. К сожалению, при загрузке конфигурационного файла в память маршрутизатора эти комментарии игнорируются, и поэтому документация в энергонезависимой или рабочей памяти фактически отсутствует.
В последних версиях в ОС IOS была введена возможность добавлять комментарии как в нумерованные, так и в именованные списки доступа. Добавление комментариев в нумерованные списки доступа осуществляется путем использования ключевого слова remark (примечание) вместо ключевых слов permit или deny после команды глобального конфигурирования ОС IOS access-list и номера списка. Примечания могут помещаться в любом месте списка доступа, и каждое может быть до 100 символов длиной. Ниже показан пример добавления примечаний в нумерованный расширенный список IP-доступа, заданный ранее для маршрутизатора компании ZIP SF-1:

```
SF-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-1 (config)#access-list 100 remark Allow smtp mail to John 's machine per Jane
SF-l(config)#access-list 100 permit tcp any host 131.108.101.99 eq smtp
SF-1 (config)#access-list 100 remark Allow DNS queries to John ' s machine per Jane
SF-1(config)#access-list 100 permit udp any host 131.108.101.99 eq domain
SF-1(config)#access-list 100 remark Nothing else gets through and gets logged
SF-1(config)#access-list 100 deny ip any any log
SF-1(config)#^Z
```

Для добавления комментариев в именованные списки доступа используется команда подрежима конфигурирования списков IP-доступа remark. Аналогично операторам permit и deny, используемым в этом подрежиме, команда remark применяется после перехода в подрежим конфигурирования списков доступа путем ввода команды ip access-list с последующим указанием имени списка. Как и примечания в нумерованных списках доступа, примечания в именованных списках доступа могут стоять в любом месте списка, и каждое может быть до 100 символов длиной. Ниже показан пример добавления примечаний в именованный расширенный список IP-доступа, ранее заданный для маршрутизатора компании ZIP SF-1:

```
SF-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-1(config)#ip access-list extended out-of-luck
SF-1(config-ext-nacl)#remark Allow smtp mail to John 's machine per Jane
SF-1(config-ext-nacl)#permit tcp any host 131.108.101.99 eq smtp
SF-l(config-ext-nacl)#remark Allow DNS queries to John 's machine per Jane
SF-1(config-ext-nacl)#permit udp any host 131.108.101.99 eq domain
SF-1(config-ext-nacl)#remark Nothing else gets through and gets logged
SF-1(config-ext-nacl)#deny ip any any log
SF-1(config-ext-nacl)#^Z
```

Наложение списков доступа

Созданные критерии фильтрации списка доступа должны быть наложены на один или несколько интерфейсов, чтобы могла выполняться фильтрация пакетов. Список доступа может накладываться как в направлении входа интерфейса, так и в направлении выхода. Когда пакеты перемещаются во входном направлении, они поступают в маршрутизатор из интерфейса, а когда в выходном — покидают маршрутизатор и затем поступают на интерфейс. Список доступа накладывается с

помощью субкоманды конфигурирования интерфейса ОС IOS ip access-group. Эта команда воспринимает в качестве параметра ключевое слово in (внутрь) или out (наружу). Если параметр не вводится, то подразумевается наличие ключевого слова out. В показанном ниже примере заданный ранее стандартный список доступа 1 накладывается на интерфейс Fast Ethernet маршрутизатора компании ZIP SF-1. Данная конфигурация запрещает пакетам, имеющим адрес происхождения 131.108.101.99, достигать пунктов назначения, находящихся за интерфейсом Fast Ethernet.

```
SF-l#configure
Configuring from terminal,   memory,   or network   [terminal]?
Enter configuration commands,   one per line.   End with CNTL/Z.
SF-1(config)#tinterface fastethernet 0
SF-1(config-if)#ip access-group 1 out
SF-1(config-if)#^Z
```

В следующем примере осуществляется наложение заданного ранее списка доступа с именем out-of-luck ("не повезло") на интерфейс Fast Ethernet маршрутизатора компании ZIP SF-1. Такая конфигурация запрещает прохождение выходящих из маршрутизатора пакетов с любым адресом за исключением тех, которые направлены хост-машине 131.108.101.99 для служб SMTP и DNS.

```
SF-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-1(config)#interface fastethernet 0
SF-1(config-if)#ip access-group out-of-luck out
SF-l(config-if)#^Z
```

После того как списки доступа будут сконфигурированы, их можно просмотреть и проверить с помощью команд режима EXEC ОС IOS show access-list и show ip access-list. Первая команда показывает все списки доступа, заданные для маршрутизатора, тогда как последняя показывает только заданные маршрутизатору списки IP-доступа (будь то нумерованные или именованные). Каждая команда может воспринимать в качестве параметра конкретный нумерованный или именованный список доступа и выводить на экран содержание только этого списка. Если параметр не указывается, то выводятся данные по всем спискам. Ниже показан результат исполнения команды show access-list на маршрутизаторе компании ZIP SF-1, свидетельствующий о том, что ранее заданные списки доступа были наложены на маршрутизатор.

```
SF-l#show access-lists
Standard IP access list 1
   deny 131.108.101.99 (50 matches)
   permit 131.108.101.0 0.0.0.255 (576 matches)
Standard IP access list sorrycharlie
   deny 131.108.101.99
   permit   131.108.101.0  0.0.0.255
Extended IP access  list  100
   permit tcp any host  131.108.101.99 eq smtp
   permit udp any host  131.108.101.99 eq domain
   deny ip any any log
Extended IP access  list out-of-luck
   permit tcp any host  131.108.101.99 eq smtp   (987 matches)
   permit udp any host  131.108.101.99 eq domain   (10987 matches)
   deny ip any any log   (453245 matches)
SF-1#
```

Как видно по выводимым данным, команды show access-list и show ip access-list выполняют подсчет количества раз, когда удовлетворялись критерии фильтрации, заданные в каждой строке списка доступа, и показывают результат такого подсчета в круглых скобках. Эта информация может быть полезной для определения эффективности и необходимости каждой строки списка доступа. Она также может помочь при устранении неполадок, выявляя возможные ошибки

конфигурирования списков доступа. Например, если показания счетчика разрешений пакетам службы имен доменов протокола UDP в списке out-of-luck не увеличиваются, и есть сообщения от пользователей о неполадках в работе этой службы, значит, доменные пакеты не проходят список доступа. Другим свидетельством может быть увеличение показаний счетчика в последней строке списка out-of-luck, в которой регистрируется количество пакетов, не соответствующих критериям списка доступа.
Показания счетчиков совпадений команд show access-list и show ip access-list сбрасываются командой режима EXEC ОС IOS clear ip access-list counters. Эта команда может иметь параметром номер или имя списка IP-доступа, для которого выполняется очистка счетчиков совпадений. Если параметр не указывается, то очищаются показания всех счетчиков совпадений во всех списках доступа.
Ниже показан пример очистки счетчиков совпадений в именованном списке IP-доступа out-of-luck маршрутизатора компании ZIP SF-1:

```
SF-l#clear ip access-list counters out-of-luck
SF-l#
```

Чтобы определить, где используется тот или иной список доступа, требуется некоторое мастерство. Если они применяются в качестве фильтров пакетов с командой ip access-group, то по результату, выводимому командой show ip interfaces, будет видно, какие списки доступа были наложены и на какой интерфейс. Если они применяются в качестве фильтров маршрутов с командой distribute-list, то уже результат исполнения команды show ip protocols покажет, стоят ли эти фильтры на входе или на выходе для каждого конкретного протокола маршрутизации. Приведенное здесь обсуждение команд для просмотра и верификации списков доступа ни в коей мере не является исчерпывающим, поскольку на списках доступа основаны многие функции фильтрации ОС IOS. И каждое конкретное применение списков доступа имеет свои соответствующие команды верификации.
Реализованные в ОС IOS компании Cisco функциональные возможности по фильтрации IP-пакетов являются очень мощным инструментарием для решения вопросов ограничения доступа к ресурсам как внутри, так и снаружи сети организации. Однако проектирование схемы межсетевой защиты является важной и сложной задачей. Проблеме обеспечения адекватной защиты сети посвящаются целые книги. Поэтому для получения дополнительной информации о защите сетевых ресурсов рекомендуем обратиться к соответствующей литературе (см. раздел "Дополнительная литература" в конце этой главы). Кроме того, компания Cisco Systems имеет великолепный аналитический материал под названием *Increasing Security on IP Networks* ("Увеличение безопасности в IP-сетях"), который можно найти на странице зарегистрированных сертифицированных пользователей устройств компании Cisco по адресу WWW.Cisco.com/univercd/cc/td/doc/cisintwk/ics/cs003.htm.

Конфигурирование основных IP-служб работы с коммутируемыми каналами передачи данных

До настоящего момента мы рассматривали возможности ОС IOS по коммутации пакетов и работе с протоколами маршрутизации. ОС IOS также позволяет маршрутизаторам и серверам доступа работать в режиме удаленного доступа. Удаленный доступ может осуществляться как по асинхронным коммутируемым каналам с применением внешних и интегрированных модемных модулей, так и по сети ISDN. Функция удаленного доступа обеспечивает удаленных пользователей и удаленные маршрутизаторы возможностью подключаться к службам IP-сети, не имея прямого подключения через интерфейс локальной или глобальной сети.
Службы удаленного IP-доступа поддерживают многие продукты, основанные на ОС IOS. Эти продукты предлагают многочисленные варианты конфигураций как аппаратной части, так и функций ОС IOS. Обсуждению служб удаленного доступа, как и другим сложным вопросам, рассматриваемым в этой главе, посвящаются целые книги. Мы же выбрали для изучения две широко распространенные базовые конфигурации удаленного IP-доступа, которые поддерживают пользователей рабочих станций с доступом по коммутируемым каналам

передачи данных. Многие из этих команд и концепций конфигурирования также применимы при реализации удаленного доступа типа маршрутизатор-маршрутизатор, который известен под названием маршрутизации с вызовом по запросу. Для получения дополнительной информации о концепции и конфигурировании маршрутизации с вызовом по запросу обратитесь к материалам следующих аналитических исследований, выполненных в компании Cisco Systems: *Dial-on-Demand Routing* ("Маршрутизация с вызовом по требованию") и *Scaling Dial-on-Demand Routing* ("Масштабирование маршрутизации с вызовом по требованию"). Эти материалы можно найти на странице зарегистрированных сертифицированных пользователей устройств компании Cisco по адресу www.cisco.com/univercd/cc/td/doc/cisintwk/ics/cs002.htm И www.Cisco.com/ univercd/cc/td/doc/cisintwk/ics/cs012.htm, соответственно.

Для обеспечения надежности соединения через службу удаленного доступа по коммутируемым каналам передачи данных (например, через службу модемной связи или через ISDN) IP-протокол переносится на протокол канального уровня, с которым такая служба работает. В службах удаленного доступа по коммутируемым каналам поддерживается несколько протоколов канального уровня, включая протоколы PPP, HDLC, SLIP (Serial Line IP) (протокол межсетевого взаимодействия по последовательным каналам) и Frame Relay. На время написания этой книги чаше всего в качестве протокола канального уровня для служб удаленного доступа по коммутируемым каналам используется протокол PPP.

Конфигурирование служб удаленного доступа по коммутируемым каналам может быть разделено на три основные области:

- конфигурирование линии или интерфейса;
- конфигурирование средств зашиты;
- конфигурирование IP-протокола.

Каждая из них будет рассмотрена на примере серверов доступа сети компании ZIP, расположенных в Сингапуре, для сценария с асинхронным удаленным доступом и доступом по сети ISDN. Службы асинхронного доступа обеспечиваются устройством Cisco 2511, которое поддерживает 16 асинхронных линий. ISDN-службы обеспечиваются устройством Cisco 4500 с интегрированным ISDN-интерфейсом BRI (Basic Rate Interface — интерфейс передачи данных с номинальной скоростью).

Конфигурирование асинхронного удаленного доступа по коммутируемым каналам

Асинхронный удаленный доступ по коммутируемым каналам связан с использованием аналоговых модемов, которые преобразовывают данные в информационные потоки, способные передаваться по телефонным линиям. Такие модемы могут либо быть встроенными в устройство (сервер доступа Cisco AS5200 AccessServer или маршрутизатор серии 3600), либо подключаться извне (сервер доступа 2511 AcessServer) через вспомогательный порт, имеющийся у большинства маршрутизаторов компании Cisco. На рис. 4.9 показана типовая схема организации удаленного доступа по коммутируемым каналам передачи данных для пользователя удаленной рабочей станции, обращающегося к сети через сервер доступа с внешними модемами.

Вне зависимости от того, подсоединены ли к модемам физические асинхронные линии, либо мы имеем дело с виртуальными линиями внутри интегрированных модемных модулей, эти линии и модемы должны быть соответствующим образом сконфигурированы, чтобы гарантировать нужную по качеству связь. Скорость передачи данных в линии, метод управления потоками данных, направление удаленного доступа и тип подключаемого модема — это наиболее важные параметры, которые подлежат установке. В главе 7 обсуждается конфигурирование виртуальных терминальных линий (vty) для управления удаленным доступом к маршрутизатору с помощью основных команд ОС IOS режима конфигурирования линий связи. Команды конфигурирования линий связи также будут использоваться для конфигурирования характеристик физических асинхронных линий (tty) подключений модемов.

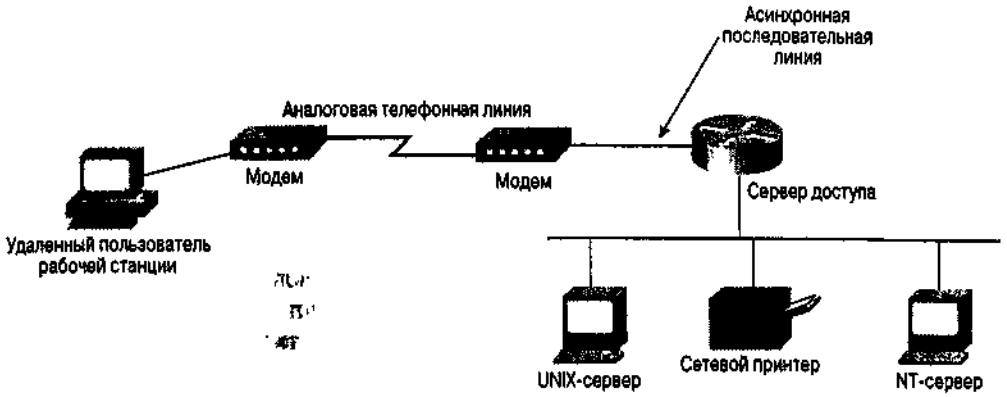


Рис. 4.9. Удаленный доступ по коммутируемым каналам к серверу доступа через модемы

Чтобы установить скорость, с которой сервер доступа будет обмениваться данными с модемами, необходимо воспользоваться субкомандой конфигурирования линий связи ОС IOS speed. Эта команда воспринимает в качестве параметра целое число, показывающее скорость передачи и приема данных в битах в секунду. Скорость передачи данных должна устанавливаться в значение, соответствующее максимальной скорости, поддерживаемой портом данных модема (максимальная скорость передачи данных, поддерживаемая сервером доступа, составляет 115200 бит/с).

Для определения метода, используемого для управления потоком информации от сервера доступа к модему, применяется субкоманда конфигурирования линий связи ОС IOS flowcontrol. Параметром данной команды является ключевое слово hardware (аппаратный) или software (программный). Эти слова отражают два поддерживаемых типа управления потоками. При скорости выше 9 600 бит/с рекомендуется использовать аппаратное управление потоками. Ниже приведен пример конфигурирования всех 16 асинхронных линий сервера доступа компании ZIP Singapore на использование аппаратного управления потоками со скоростью обмена данными 115200 бит/с. Заметим, что здесь используется основная команда конфигурирования line, в которой называются асинхронные линии с 1 по 16, в отношении которых и будут действовать субкоманды.

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Sing2511(config)#line 1 16
Sing2511(config-line)#speed 115200
Sing2511(config-line)#flowcontrol hardware
Sing2511(config-line)#^Z
```

После того как будет выбрана скорость обмена данными и метод управления потоками, в сервер доступа следует ввести информацию о типе подключаемого модема и направлении удаленного доступа по коммутируемой линии. Ввод информации о типе модема облегчает задачу конфигурирования удаленного доступа по коммутируемым линиям за счет исключения необходимости выполнения установок модема вручную. Кроме того, сервер доступа может сбрасывать установки модема после каждого соединения по вызову, чем гарантирует соответствующую работу пула удаленного доступа.

Параметр направления удаленного доступа дает серверу доступа распоряжение, как реагировать на сигналы модема, посылаемые ему во время соединения по вызову. Для конфигурирования как типа подключаемого модема, так и направления удаленного доступа используется субкоманда конфигурирования линии связи ОС IOS modem. Конфигурирование типа модема осуществляется путем применения команды modem autoconf igure. Параметром этой команды является ключевое слово discovery либо type. Ключевое слово discovery дает указание серверу доступа определить тип модема, чтобы выбрать соответствующие установки. Ключевое слово type с указанием после него одного из заранее описанных или задаваемых пользователем типов модемов отдает серверу доступа распоряжение о выборе модемных установок для названного типа модема.

ОС IOS поддерживает ряд популярных типов модемов, включая Courier и Sportster компании U.S. Robotics и T3OOO компании Telebit. Если тип модема предварительно не описан, то пользователь может внести дополнительные типы и соответствующие установки, воспользовавшись командой

конфигурирования ОС IOS modemcap. Для установки направления удаленного доступа используются ключевые слова dialin и inout, выступающие в роли параметра команды modem. Ниже показан пример конфигурирования сервера доступа компании ZIP Singapore на использование установок модема, соответствующих модему компании U.S. Robotics типа Courier. Направление удаленного доступа конфигурируется как работа на вход (dialin).

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Sing2511(config)#line 1 16
Sing2511(config-line)#modem autoconfigure type usr_courier
Sing2511(config-line)#modem dialin
Sing2511(config-line)#^Z
```

Совет

Даже если асинхронные линии работают только на вход, рекомендуется при начальном конфигурировании и при устранении неполадок устанавливать линии в режим двусторонней работы (inout). Это позволит виртуальному терминалу иметь прямой доступ к асинхронной линии по протоколу Telnet, в результате чего будет возможно конфигурирование и верификация установок модема вручную. Этот метод доступа через виртуальный терминал, также известный под названием метода обратного протокола Telnet, более подробно описан в материале *Configuring Modems* ("Конфигурирование модемов"), который можно найти на страничке зарегистрированных сертифицированных пользователей устройств компании Cisco по адресу www.cisco.com/univercd/cc/td/doc/product/software/iosll3ed/dsqcg/qcmodems.htm.

После асинхронных линий следует сконфигурировать средства защиты сервера доступа. Как обсуждается в главе 7, защита доступа осуществляется в два этапа. На первом выполняется аутентификация, т.е. идентификация того, кто пытается получить доступ. На втором происходит авторизация идентифицированного пользователя на выполнение конкретных задач или предоставление ему доступа к конкретным службам. Для удаленного IP-доступа по коммутируемым линиям вводятся понятия типа аутентификации и типа авторизации, которые используют локально конфигурируемую информацию пользователя. Этот аспект не обсуждается в главе 7. Команды аутентификации и авторизации используют локально конфигурируемую информацию пользователя. Как вариант, вместо локально конфигурируемой информации может использоваться сервер защиты, например, TACACS+ или RADIUS. Этот случай и рассматривается в главе 7.

Для пользователей, проходящих аутентификацию, которые пытаются получить доступ к IP-службам по протоколу PPP, используется аутентификация AAA типа ppp. Она запускается командой конфигурирования ОС IOS aaa authentication ppp. Параметрами этой команды выступают имя списка аутентификации или ключевое слово default и название одного или нескольких методов аутентификации, например, local (по локальной информации) или, как в обсуждаемом случае, TACACS+. После того как PPP-пользователь будет идентифицирован, он должен быть авторизован на использование сетевых служб (протокол PPP является одной из них). Пользование сетевыми службами авторизуется командой aaa authorization network. Эта команда воспринимает в качестве параметра название одного или нескольких типов авторизации. Ниже приведен пример конфигурирования сервера доступа компании ZIP в Сингапуре на аутентификацию PPP-пользователей по локально конфигурируемой информации пользователя и на авторизацию доступа к сетевым службам всех пользователей, прошедших аутентификацию:

```
Sing2511#configure
Configuring from terminal,   memory,   or network   [terminal]?
Enter  configuration  commands,   one per  line.   End with CNTL/Z.
Sing2511(config)#aaa authentication default ppp local
Sing2511(config)#aaa authorization network default if-authenticated
Sing2511(config)#^Z
```

Аутентификационная информация для PPP-пользователей конфигурируется локально, поэтому должны быть сконфигурированы реальные имена пользователей и пароли, используемые в процессе аутентификации. Эта информация конфигурируется с помощью команды глобального конфигурирования ОС IOS username. Параметрами этой команды являются идентификатор пользователя, который должен использоваться для аутентификации, ключевое слово password и сам пароль, используемый при аутентификации пользователя. Хотя пароль вводится в виде обычного читаемого текста, при активации режима шифрования пароля он преобразуется в шифрованную цепочку символов, что описывается в главе 7. Ниже приводится пример создания двух имен пользователей и паролей на сервере доступа компании ZIP в Сингапуре для двух пользователей: Джона (John) и Джейн (Jane).

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Sing2511(config)#username John password foo
Sing2511(config)#rusername jane password bar
Sing2511(config)#^Z
```

Последним шагом в конфигурировании служб асинхронного удаленного IP-доступа по коммутируемым линиям является предоставление информации IP-протокола, которая используется для открытия и поддержания IP-сеанса удаленного доступа. Вместо ввода информации IP-протокола субкомандами конфигурирования линий связи эта информация связывается с типом интерфейса, который представляет асинхронную линию, как это делается в любой другой среде локальной или глобальной сети. Этот тип интерфейса называется *асинхронным интерфейсом*, и каждая асинхронная линия на сервере доступа имеет соответствующий асинхронный интерфейс. Информация IP-протокола может вводиться отдельно для каждого асинхронного интерфейса, на котором возможен сеанс удаленного доступа по коммутируемой линии, или только один раз через коллективный асинхронный интерфейс, называемый *групповым асинхронным интерфейсом*.
Групповой асинхронный интерфейс может упростить конфигурирование в тех случаях, когда одни и те же команды конфигурирования будут повторно использоваться в отношении нескольких асинхронных интерфейсов. В этом случае также используется субкоманда конфигурирования интерфейса ОС IOS group-range, чтобы идентифицировать отдельные асинхронные интерфейсы, подлежащие включению в групповую структуру. Ниже показан пример добавления к трем асинхронным интерфейсам команды description:

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Sing2511(config)#interface async 1
Sing2511(config-if)#description dialup pool on Singapore 2511
Sing2511(config-if)#interface asyno 2
Sing2511(config-if)#desoription dialup pool on Singapore 2511
Sing2511(config-if)#interfaea async 3
Sing2511(config-if)#description dialup pool on Singapore 2511
Sing2511(config-if}#^Z
```

А вот этот же пример, но с использованием группового асинхронного интерфейса:

```
Sing2511#configure
Configuring from terminal,  memory,   or network   [terminal]?
Enter configuration commands,   one per line.   End with CNTL/Z.
Sing2511(config)#interface group-async 1
Sing2511(config-if)#description dialup pool on Singapore 2511
Sing2511(config-if)#group-range 1 3
Sing2511(config-if)#^Z
```

Предоставляемая асинхронным интерфейсам информация IP-протокола подразделяется на три категории.

- Конфигурация IP-адреса для асинхронного интерфейса.
- Информация об IP-адресах, отсылаемых пользователю, работающему по уда ленному доступу.
- Информация о том, как протоколы IP и PPP должны работать на асинхронном интерфейсе.

Начнем с рассмотрения команд работы протоколов IP и PPP. Сначала асинхронному интерфейсу сообщается об использовании протокола PPP в качестве метода инкапсуляции для таких служб, как IP. *Задание типа* инкапсуляции осуществляется субкомандой конфигурирования интерфейса ОС IOS encapsulation Параметром этой команды является ключевое слово (например, ppp или slip), которое описывает тип инкапсуляции, используемый в интерфейсе.

После того как в качестве метода инкапсуляции будет сконфигурирован протокол PPP, администратор сети может либо сконфигурировать асинхронную линию на работу только в качестве порта сетевых служб удаленного доступа по коммутируемым линиям (т.е. пользователю разрешается использовать только те сетевые службы, которые сконфигурированы на порте, например, протокол PPP или протокол SLIP), либо позволить пользователю принимать по удаленному доступу подсказки режима EXEC, чтобы тот мог вручную выбрать нужную службу. Для выбора способа работы используется субкоманда конфигурирования интерфейса async mode. Параметром команды выступают ключевые слова interactive или dedicated, которые и устанавливают нужный режим работы.

При принятии решения о выборе режима (интерактивного или выделенного) обычно учитывается уровень опытности удаленного пользователя и то, как используется асинхронный интерфейс. Конфигурирование на работу в выделенном режиме избавляет администратора сети от необходимости набирать телефонный номер и авторизоваться на использование команд режима EXEC. Интерактивный режим может поддерживать как исполнение команд режима EXEC, так и работу с сетевыми службами. Однако недостатком интерактивного режима является то, что неопытные пользователи могут неправильно сконфигурировать программное обеспечение вызова по телефонной линии и оказаться в командной строке режима EXEC, не подозревая об этом.

При работе в интерактивном режиме существует дополнительный набор команд, которые упрощают процесс вызова по номеру. Эти команды позволяют серверу доступа определять тип соединения, попытка установить которое имеет место, не требуя от пользователя задания службы в командной строке режима EXEC. Такой процесс называется *автовыбором*. Активируется он командой конфигурирования линий ОС IOS autoselect. В качестве параметра этой команды выступает ключевое слово, описывающее протокол канального уровня, который выбирается автоматически, или время, в течение которого выполняется автовыбор (обычно это время аутентификации пользователя). Применение автовыбора при конфигурировании асинхронных линий на работу в интерактивном режиме является для большинства пользователей простейшим методом доступа к PPP- и IP-службам сервера доступа.

Последняя команда, связанная с работой протокола PPP и необходимая для интерфейса, отдает протоколу PPP распоряжение выполнять аутентификацию и авторизацию удаленных и работающих на коммутируемых линиях пользователей до подключения к сетевым PPP- и IP-службам. Это гарантирует, что только авторизованные пользователи получат доступ к сетевым службам, имеющимся на сервере доступа. Данная команда также говорит серверу доступа о том, какой протокол аутентификации использовать между сервером доступа и удаленным клиентом, работающим по телефонным каналам. Возможны три протокола: протокол аутентификации по квитированию вызова (Challenge Handshake Authentication Protocol — CHAP), протокол аутентификации по квитированию вызова компании Microsoft (Microsoft Challenge Handshake Authentication Protocol — MS-CHAP) и протокол аутентификации по паролю (Password Authentication Protocol — PAP).

Распоряжение серверу доступа на выполнение аутентификации отдает команда конфигурирования интерфейса ОС IOS ppp authentication. В качестве параметра она воспринимает ключевые слова chap, ms-chap или pap, которыми задается протокол аутентификации. В одной конфигурационной команде можно задавать один или сразу несколько протоколов, если удаленные пользователи работают с несколькими. протоколами

аутентификации. Эта команда также воспринимает в качестве параметра ключевое слово callin, которое отдает серверу доступа распоряжение выполнять при аутентификации запрос на квитирование только для входных вызовов по номеру. По умолчанию запрос выполняется как для входящих, так и для выходящих звонков. В реализациях некоторых поставщиков ответ на запрос не производится, если принимается входящий звонок.

Описанные выше команды составляют тот минимум, который требуется для конфигурирования работы PPP-протокола для удаленных пользователей, работающих с коммутируемыми линями связи. Поскольку сегодня большое количество удаленных пользователей работает с программным обеспечением компании Microsoft, администратор сети, возможно, захочет добавить поддержку протокола сжатия данных при двухточечной маршрутизации компании Microsoft (Microsoft Point-to-Point Compression— MPPC), описанного в Запросе на комментарий № 2118 "Microsoft Point-to-Point Compression Protocol" {"Протокол сжатия данных при двухточечной маршрутизации компании Microsoft"). Сжатие данных оптимизирует передачу информации в такой среде, как коммутируемые линии связи, что позволяет передавать большие объемы информации, чем это было бы возможно в обычных условиях. При относительно медленных коммутируемых линиях, которые работают на скоростях от 28000 до 53000 бит/с, сжатие может увеличить скорость передачи данных до полутора раз.

Добавление поддержки сжатия данных для удаленных пользователей, работающих на коммутируемых линях связи, выполняется с помощью субкоманды конфигурирования интерфейса ОС IOS compress. Параметром команды compress являются ключевые слова mppc, stac или predictor, указывающие тип сжатия, согласование использования которого должно выполняться при установке соединения удаленным пользователем. Ключевые слова stac и predictor обозначают использование алгоритмов сжатия данных STAC или Predictor, соответственно. STAC представляет собой широко известный алгоритм сжатия, поддерживаемый многими клиентами, работающими с вызовом по номеру, включая системы Windows 95, и это будет хороший выбор, если поддерживается большая группа не Microsoft-ориентированных пользователей или пользователей систем, работающих под ОС Windows 95. Алгоритм Predictor менее известен. Выбор протокола сжатия данных при двухточечной маршрутизации компании Microsoft осуществляется с помощью ключевого слова mppc. Учитывая, что ОС Windows NT поддерживает только протокол MPPC, а ОС Windows 95/98 поддерживает как MPPC, так и STAC, выбор этого алгоритма обеспечивает наибольшую гибкость для тех администраторов сети, которые работают с несколькими операционными системами компании Microsoft.

Рассмотрим пример конфигурирования команд работы протоколов PPP и IP на сингапурском сервере доступа компании ZIP. В этом примере конфигурируются все 16 асинхронных линий с помощью метода группового асинхронного интерфейса. Интерфейсы определяются как использующие PPP-инкапсуляцию и работают в интерактивном режиме, позволяя асинхронным линиям выполнять автовыбор протокола PPP во время регистрации. Кроме того, протокол PPP конфигурируется на аутентификацию входящих вызовов по номеру с использованием протоколов аутентификации CHAP, MS-CHAP и PAP с последующим согласованием сжатия по алгоритму компании Microsoft.

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Sing2511(config)#interface group-async 1
Sing2511(config-if)#group-range 1 16
Sing2511(config-if)#encapsulation ppp
Sing2511(config-if)#async mode interactive
Sing2511(config-if)#ppp authentication chap ms-chap pap callin
Sing2511(config-if)#compress mppc
Sing2511(config-if)#line 1 16
Sing2511(config-line)#autoselect ppp
Sing2511(config-line)#autoselect during-login
Sing2511(config-line)#^Z
```

Задав рабочий режим протокола PPP, можно выполнить задание на асинхронных интерфейсах IP-адресов. В нормальных условиях каждый пользователь, работающий с коммутируемыми линиями связи, имеет только один IP-адрес, связанный с его рабочей станцией (в отличие от маршрутизатора, к которому подключен целый сегмент локальной сети и который для обеспечения нужной связи должен выполнять маршрутизацию на центральный узел сети компании). Так как каждый удаленный пользователь использует IP-адрес в отдельном соединении по коммутируемой линии и на отдельном асинхронном интерфейсе, то фактический IP-адрес самого асинхронного интерфейса не важен. По сути, каждый асинхронный интерфейс может рассматриваться как размещаемый в том же адресном пространстве, что и интерфейс подсоединенной локальной сети. Можно даже допустить, что IP-адрес удаленного пользователя назначается из этого адресного пространства. Если посмотреть с другой стороны, то логически удаленный пользователь подключается к сегменту локальной сети по длинному кабелю — телефонной линии. А телефонной линии не присваивается никакой IP-адрес, как и при подключении сетевой рабочей станции по кабелю 1OBaseT.
Рабочая станция принимает IP-адрес из того же пространства сетевых IP-адресов, которое назначено интерфейсу локальной сети сервера доступа. Сервер доступа отвечает за прием пакетов из локальной сети от имени удаленного пользователя. Затем он направляет эти пакеты в соответствующий телефонный звонок. Делает это сервер доступа, вводя в момент соединения по коммутируемой линии в свою таблицу маршрутизации маршрут хост-машины (сетевого маршрута с 32-разрядной маской) и отвечая на ARP-запросы об IP-адресах, назначенных сеансам удаленного доступа по коммутируемым линиям.
Согласно описанному выше методу сами асинхронные интерфейсы не имеют IP-адресов, поэтому для активации IP-обработки на асинхронных интерфейсах может быть использована субкоманда конфигурирования интерфейсов ОС IOS ip unnumbered. Эта команда уже рассматривалась в разделе "Адресация интерфейса глобальной сети с двухточечной маршрутизацией". Используется она таким же образом, как уже описывалось ранее, а именно: для задания интерфейса локальной сети сервера доступа в качестве базового интерфейса. Ниже показан пример конфигурирования асинхронных интерфейсов группового асинхронного интерфейса сервера доступа компании ZIP в Сингапуре в качестве ненумерованных:

```
Sing2511#configure
Configuring  from terminal,   memory,   or network   [terminal]?
Enter  configuration commands,   one per  line. End with CNTL/Z.
Sing2511(config)#interface group-async  1
Sing2511(config-if)#ip unnumbered ethernet 0
Sing2511(config-if)#^Z
```

Последним этапом в организации IP-взаимодействия по коммутируемым линиям на асинхронном интерфейсе является конфигурирование IP-адресов, назначаемых удаленному клиенту на время соединения. Метод, который используется для назначения IP-адреса удаленному клиенту, определяется субкомандой конфигурирования интерфейса ОС IOS peer default ip address. Задавая конкретный IP-адрес в качестве параметра этой команды, можно назначать каждому асинхронному интерфейсу отдельные IP-адреса. Однако при этом каждый асинхронный интерфейс должен быть сконфигурирован вручную на IP-адрес, который будет назначаться удаленным клиентам, подключающимся к этому интерфейсу.
Более гибкий метод заключается в том, чтобы назначать адреса из состава одного или нескольких адресных пулов, организованных на сервере доступа с помощью команды parameter pool. Этот метод также дает пользователям с постоянно закрепленными IP-адресами возможность дозваниваться в любой модемный порт, поскольку сервер доступа будет принимать предлагаемый IP-адрес удаленного клиента, если тот попадает в предварительно заданный пул адресов. При конфигурировании пулового метода указывается конкретное имя пула адресов.
Сами пулы адресов задаются путем применения команды глобального конфигурирования ОС IOS ip local pool. В качестве параметров этой команды выступают имя пула, начальный и конечный адреса этого пула. При этом необходимо, чтобы IP-адреса принадлежали той же IP-сети, что и интерфейс локальной сети сервера доступа. Конечно, эти адреса не должны принадлежать какой-

либо рабочей станции, стоящей в сегменте данной локальной сети. Ниже показан пример конфигурирования на сервере доступа компании ZIP в Сингапуре асинхронных интерфейсов ранее заданной структуры группового асинхронного интерфейса, которым назначаются IP-адрес из локального пула с названием modem-users ("пользователи модемов"). Заметим, что пул задается имеющим только 16 адресов, так как на сервере доступа существуетт всего 16 модемов и асинхронных интерфейсов.

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Sing2511(config)#interface group-async 1
Sing2511(config-if)#peer default ip address pool modem-users
Sing2511(config-if)#ip local pool modem-users 131.108.1.111 131.108.1.126
Sing2511(config-if)#^Z
```

Хотя адресные пулы и представляют собой наиболее гибкий метод назначения IP-адресов, не существует метода координации назначения адресов при наличии нескольких серверов доступа. В такой ситуации, возможно, будет лучше назначать адреса из центрального сервера адресов по типу сервера протокола динамического конфигурирования хост-машин (Dynamic Host Configuration Protocol — DHCP). Чтобы соответствовать этому методу, ОС IOS может выступать в качестве уполномоченного клиента, запрашивая у DHCP-сервера IP-адрес от имени удаленного клиента. Этот метод конфигурирования разрешается путем задания в команде peer default ip address параметра в виде ключевого слова dhcp. Но для того, чтобы сервер доступа посылал запросы по поводу адреса, он должен быть сконфигурирован на IP-адрес DHCP-сервера, что достигается командой глобального конфигурирования ОС IOS ip dhcp-server. Пулы адресов, заданные на DHCP-сервере, будут содержать адреса, входящие в сетевой IP-адрес интерфейса локальной сети сервера доступа. Ниже приведен пример конфигурирования сервера доступа компании ZIP в Сингапуре на назначение IP-адресов удаленным клиентам с использованием протокола DHCP:

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Sing2511(config)#interface group-async 1
Sing2511(config-if)#peer default ip address dhcp
Sing2511(config-if)#ip dhcp-server  131.108.21.70
Sing2511(config-if)#^Z
```

Во многих реализациях PPP-обмена для удаленных клиентов, работающих по коммутируемым линиям передачи данных, в процессе установки соединения по телефонному вызову используется нестандартный метод получения IP-адресов серверов имен служб DNS и NetBIOS/WINS. Этот метод описывается в информационном Запросе на комментарий № 1877 "PPP Internet Protocol Control Protocol Extensions" ("Расширения протокола контроля соединений для протокола двухточечной связи по протоколу межсетевого обмена"). Хотя этот метод не стандартизован, он достаточно широко применяется (особенно в реализациях удаленного доступа по коммутируемым линиям компании Microsoft). Сервер доступа тоже может поддерживать описанные в этом документе методы предоставления адресов серверов имен служб DNS и NetBIOS/WINS. Для конфигурирования этих опций в ранних версиях ОС IOS использовалась команда глобального конфигурирования async-bootp. При конфигурировании IP-адреса (адресов) DNS-серверов команда использует в качестве параметра ключевое слово dns-server, после которого указывается один или несколько IP-адресов. При конфигурировании IP-адреса (адресов) NetBIOS/WINS-серверов команда использует в качестве параметра ключевое слово nbns-server, после которого указывается один или несколько IP-адресов. Ниже показан пример конфигурирования сервера доступа компании ZIP в Сингапуре на поставку IP-адресов серверов имен служб DNS и NetBIOS/WINS по методу Запроса на комментарий № 1877 с использованием команды async-bootp:

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Sing2511(config)#async-bootp dns-server 131.108.101.34
131.108.101.35
Sing2511(config)#async-bootp nbns-server 131.108.21.70
Sing2511(config)#^Z
```

Примечание

Поставка адресов серверов имен служб DNS и NetBIOS/WINS имеет мало общего с протоколом BOOT, однако команда async-boot использовалась для активации этой функции ОС IOS благодаря вводу расширений в существующие команды протокола согласования SLIP BOOT. Это было сделано для того, чтобы не создавать отдельные команды и механизмы в протоколе PPP для реализации нестандартного метода из Запроса на комментарий № 1877.

Недостатком применения команды async-boot для генерации имен серверов служб DNS и NetBIOS/WINS является то, что эта команда представляет собой команду глобального конфигурирования ОС IOS. А это приводит к тому, что сконфигурированные через нее адреса поставляются всем удаленным пользователям, работающим с сервером доступа, вне зависимости от того интерфейса удаленного доступа, с которым они могут соединяться. Как оказалось, этот метод не подходит для тех сетевых администраторов, которые хотят поддерживать несколько типов соединений удаленного доступа или различные классы пользователей и присваивать адреса разных серверов для таких соединений или пользователей. В последующих версиях ОС IOS субкоманда конфигурирования интерфейса ppp ipcp позволила сетевым администраторам управлять этими опциями на поинтерфейсной основе. При конфигурировании IP-адреса (адресов) DNS-серверов команда использует в качестве параметра ключевое слово dns, после которого указывается один или два IP-адреса. При конфигурировании IP-адреса (адресов) NetBIOS/WINS-серверов команда использует в качестве параметра ключевое слово wins, после которого указывается один или два IP-адреса. Ниже показан пример конфигурирования сервера доступа компании ZIP в Сингапуре на поставку IP-адресов серверов имен служб DNS и NetBIOS/WINS по методу Запроса на комментарий № 1877 с использованием команды ppp ipcp:

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Sing2511(config)#interface group-async 1
Sing2511(config-if)#ppp ipcp dns 131.108.101.34 131.108.101.35
Sing2511(config-if)#ppp ipcp wins 131.108.21.70
Sing2511(config-if)#^Z
```

Удаленный ISDN-доступ

Как и асинхронный доступ по коммутируемым каналам, удаленный ISDN-доступ связан с использованием общественной телефонной сети, через которую пользователи удаленных рабочих станций получают доступ к сетевым службам, если они не имеют прямого подключения через интерфейс локальной или глобальной сети. ISDN-доступ отличается от асинхронного тем, что звонки передаются с использованием цифровых синхронных сигналов. Как говорилось в главе 3, данные преобразовываются в потоки цифровой информации либо интегрированными в маршрутизатор ISDN-интерфейсами, либо подсоединяемыми внешними ISDN-устройствами, называемыми терминальными адаптерами (TA). Удаленные пользователи рабочих станций также используют для подключения к ISDN-службе либо интегрированные в ПК ISDN-платы, либо внешние TA. На рис. 4.10 показан типовой сценарий доступа по коммутируемым каналам связи для удаленного пользователя рабочей станции, обращающегося к сети через сервер доступа с интегрированными ISDN BRI-интерфейсами.

Многие из задач конфигурирования, необходимые для настройки служб асинхронного IP-

доступа по коммутируемым каналам, должны быть выполнены и при настройке служб ISDN IP-доступа. Однако, в отличие от конфигурирования асинхронной связи, они не требуют ввода команд конфигурирования линии, так как маршрутизатор имеет непосредственно интегрированный в него ISDN-интерфейс, а TA подключается прямо к синхронному последовательному интерфейсу. Если маршрутизатор имеет интегрированный ISDN-интерфейс, то все команды, которые управляют его взаимодействием с
ISDN-сетью, применяются непосредственно к интерфейсу. В главе 3 показан пример использования идентификаторов профилей ISDN-служб (ISDN SPIDs) для конфигурирования ISDN-интерфейса BRI. Если маршрутизатор подключается к ISDN-сети через внешний TA, то для соответствующего взаимодействия с ISDN-сетью он конфигурируется своими методами. Это сводит конфигурирование ISDN-служб удаленного IP-доступа по коммутируемым каналам связи к двум задачам: конфигурирование средств защиты и задание IP-информации.

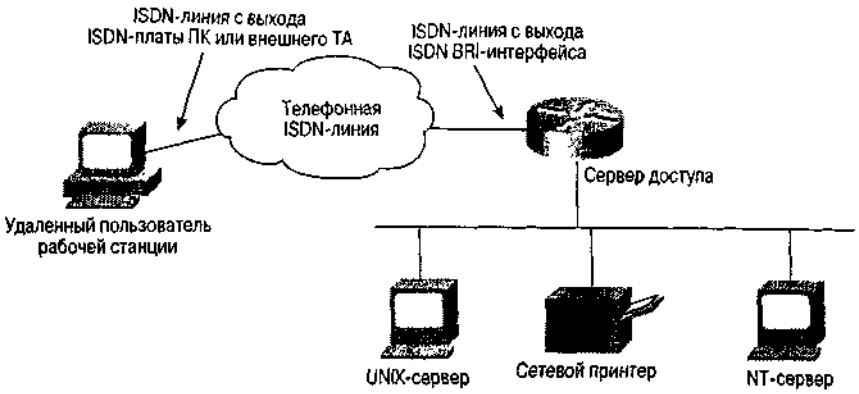


Рис. 4.10. Удаленный доступ к серверу доступа по коммутируемым каналам связи в ISDN-сети

Как и асинхронные интерфейсы, ISDN-интерфейсы могут конфигурироваться индивидуально или группой. При конфигурировании группой команды одновременного конфигурирования нескольких ISDN-интерфейсов адресуются специальному типу интерфейса, называемому интерфейсом вызова по номеру (dialer interface). Отдельные ISDN-интерфейсы по-прежнему конфигурируются с использованием своих специфических для ISDN команд, например, команд ввода SPID-информации. Однако команды, определяющие протоколы PPP, IP и их работу, конфигурируются на интерфейсе вызова по номеру. Каждый ISDN-интерфейс, который включается в структуру интерфейса вызова по номеру, конфигурируется командой dialer rotary-group. Эта команда воспринимает в качестве параметра целое число, представляющее тот интерфейс вызова по номеру, к которому принадлежит конфигурируемый интерфейс. Например, интерфейсы, конфигурируемые командой dialer rotary-group 1, принадлежат интерфейсу вызова по номеру 1. Ниже приводится пример конфигурирования четырех ISDN BRI-интерфейсов ISDN-сервера доступа компании ZIP в Сингапуре на принадлежность к интерфейсу вызова по номеру 1:

```
SingISDN#configure
Configuring  from terminal,   memory,   or network   [terminal]? Enter
configuration commands,   one per line.   End with CNTL/Z.
SinglSDN(config)#interface bri 4
SinglSDN(config-if)#dialer rotary-group 1
SingISDN(config-if)#interface bri 5
SinglSDN(config-if)#dialer rotary-group 1
SingISDN(config-if)#interface bri  6
SingISDN(config-if)#dialer rotary-group  1
SingISDN(config-if)#interface bri 7
SinglSDN(config-if)#dialer rotary-group  1
SinglSDN(config-if)#^Z
```

Рассмотрим конфигурирование средств защиты сервера доступа для служб работы с коммутируемыми каналами в IP-сети, которые обсуждались в предыдущем разделе. Как и при

асинхронном удаленном доступе, активация функций PPP-аутентификации и авторизации в сети выполняется командами глобального конфигурирования ОС 1OS aaa authentication ppp и aaa authorization network, соответственно. Для задания имен пользователей, обращающихся за доступом в сеть, используется команда глобального конфигурирования ОС IOS username. Ниже приведен пример конфигурирования ISDN-сервера доступа компании ZIP в Сингапуре на выполнение PPP-аутентификации и авторизации, а также задания пар из имени пользователя и пароля для удаленных пользователей Джима (Jim) и Джанет (Janet):

```
SinglSDN#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SinglSDN (conf ig)#aaa authentication default ppp local
SinglSDN (config)#aaa authorization network default if-authenticated
SinglSDN (conf ig)#username jim password dog
SinglSDN (conf ig)#username Janet password house
SinglSDN (config)#^Z
```

Информация IP-протокола, назначаемая ISDN-интерфейсам, подразделяется на три категории.

- Информация о том, как протоколы IP и PPP должны работать на ISDN- интерфейсе.
- Конфигурация IP-адреса для ISDN-интерфейса.
- Информация об IP-адресах, отсылаемых пользователю, работающему по удаленному доступу.

Начнем с краткого повторения рассмотренных ранее команд задания работы протоколов PPP и IP, а затем познакомимся с четырьмя новыми командами, используемыми для ISDN-интерфейсов.
Как мы видели на примере реализации работы протокола IP при асинхронном удаленном доступе, задание на ISDN-интерфейсе протокола PPP в качестве протокола канального уровня для протокола IP выполняется с помощью субкоманды конфигурирования интерфейса ОС IOS encapsulation. Активация PPP-аутентификации до начала предоставления доступа к службам IP-сети и задание протокола аутентификации осуществляется субкомандой конфигурирования интерфейса ОС IOS ppp authentication. С помощью субкоманды конфигурирования интерфейса ОС IOS compress mppc можно добавить сжатие данных по алгоритму компании Microsoft. Ниже приведен пример конфигурирования ISDN-сервера доступа компании ZIP в Сингапуре на использование протокола PPP в ISDN-интерфейсе вызова по номеру, а также выдачи серверу доступа команды на использование аутентификации и авторизации при получении доступа к сетевым службам. Здесь же показана активация на интерфейсе вызова по номеру режима сжатия данных по алгоритму компании Microsoft:

```
SinglSDN#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SinglSDN (config)#interface dialer 1
SinglSDN (config-if )#encapsulation ppp
SinglSDN (conf ig-if)#ppp authentication chap ms-chap pap callin
SinglSDN(config-if)#compress mppc
SinglSDN(config-if)#^Z
```

ISDN является канализируемой службой, т.е. она способна поддерживать несколько соединений через один физический интерфейс. Это позволяет удаленные ISDN-клиентам устанавливать одновременно несколько соединений с сервером доступа. Подобная способность дает удаленной ISDN-станции доступ к удвоенно? пропускной способности линии при использовании одного физического интерфейса. Эффективное использование многоканальности достигается за счет мультиплексирования данных между несколькими соединениями благодаря программно реализованному алгоритму для протокола PPP, называемому алгоритмом мультисвязи. Режим многоканального протокола PPP активируется командой конфигурирования интерфейса ОС IOS ppp multilink.
Для управления процессом открытия и закрытия ISDN-каналов с помощью команды глобального конфигурирования ОС IOS dialer-list задается список интересных пакетов. В

качестве параметров этой команды выступают названия конкретных протоколов, которые должны рассматриваться как интересные с точки зрения перевода (или поддержания) канала в активное состояние. Кроме того списки доступа можно использовать для получения дополнительной детализации вплоть до конкретных IP-адресов и типов служб транспортных протоколов. Правила, определяемые командой dialer-list, накладываются на интерфейс с помощью субкоманды конфигурирования интерфейса ОС IOS dialer-group, в которой параметром команды задается номер списка. Ниже показан пример конфигурирования в ISDN-сервере доступа компании ZIP в Сингапуре поддержки режима многоканального протокола PPP. Список интересных пакетов задается расширенным списком доступа 102.

```
SinglSDN#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SinglSDN(config)#interfaoe dialer 1
SinglSDN(config-if)#ppp multilink
SinglSDN(config-if)#dialer-group 1
SinglSDN(config-if)#dialer-list 1 protocol ip list 102
SinglSDN(config)#access-list 102 permit top any any eq telnet
SinglSDN(config)#access-list 102 permit tcp any any eq www
SinglSDN(config)#access-list 102 permit udp any any eq domain
SinglSDN(config)#access-list 102 permit tcp any any eq ftp
SinglSDN(config)#^Z
```

Примечание

Детальное управление распределением полосы пропускания при использовании свойства многоканальности описывается в Запросе на комментарий № 2125 "Bandwidth Allocation Control Protocol (BACP)" ("Протокол управления распределением полосы пропускания"). Протокол выделения полосы пропускания (Bandwidth Allocation Protocol — BAP), который является подмножеством протокола BACP, обеспечивает набор правил, управляющих динамическим распределением полосы пропускания в процессе управления звонком, и является стандартным методом добавления и удаления каналов из группы активных каналов. Серверы доступа и удаленные клиенты согласовывают между собой правила, по которым во время сеанса динамически увеличивается или уменьшается полоса пропускания. Функция поддержки протокола BACP ведена в ОС IOS версии 11.3.

Присвоение IP-адресов ISDN-интерфейсам серверов доступа и удаленным рабочим станциям с доступом по коммутируемым каналам осуществляется так же, как и для асинхронных интерфейсов. Если к ISDN-интерфейсам сервера доступа обращаются только удаленные ISDN-рабочие станции, то им не надо назначать конкретные IP-адреса. В этом случае интерфейс может конфигурироваться как ненумеруемый с помощью субкоманды конфигурирования интерфейса ОС IOS ip unnumbered. IP-адреса удаленным клиентам с доступом по коммутируемым каналам могут присваиваться с использованием любого из трех ранее рассмотренных методов с помощью субкоманды peer default ip address. Эти методы включают назначение отдельного IP-адреса, связанного с каждым ISDN-интерфейсом, использование пула IP-адресов, которые будут назначаться удаленным ISDN-клиентам, или назначение IP-адресов, получаемых для удаленных ISDN-клиентов от DHCP-сервера.
Ниже приведен пример конфигурирования ISDN-сервера доступа компании ZIP в Сингапуре на назначение подключающимся к ISDN-интерфейсам удаленным клиентам IP-адресов из пула адресов с именем isdn-users ("ISDN-пользователи"):

```
SinglSDN#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SinglSDN(config)#interface dialer 1
SinglSDN(config-if)#peer default ip address pool isdn-users
SinglSDN(config-if)#ip local pool isdn-users 131.108.1.91 131.108.1.106
SinglSDN(config-if)#^Z
```

IP-адреса серверов имен служб DNS и NetBIOS/WINS тоже могут поставляться удаленным ISDN-клиентам с использованием метода, изложенного в Запросе на комментарий № 1877. Как и при использовании асинхронных интерфейсов, ISDN-клиентам поставляются эти адреса с помощью конфигурирования команд глобального конфигурирования ОС IOS async-bootp dns-server и async-bootp nbns-server или субкоманд конфигурирования интерфейса ppp ipcp dns и ppp ipcp wins. При использовании любого метода IP-адреса поставляются как параметры команд. Ниже приведен пример конфигурирования ISDN-сервера доступа компании ZIP в Сингапуре на поставку IP-адресов служб DNS и NetBIOS/WINS удаленным ISDN-клиентам с использованием команд async-bootp:

```
SingISDN#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SinglSDN(config)#async-bootp dns-server 131.108.101.34 131.108.101.35
SinglSDN(config)#async-bootp nbns-server 131.108.21.70
SinglSDN(config)#^Z
```

Ниже представлен пример конфигурирования ISDN-сервера доступа компании ZIP в Сингапуре на поставку IP-адресов служб DNS и NetBIOS/WINS удаленным ISDN-клиентам с использованием команд ppp ipcp:

```
SinglSDN#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands,one per line.End with CNTL/Z.
SinglSDN(config)#interface dialer 1
SinglSDN(config-if)#ppp ipcp dns 131.108.101.34 131.108.101.35
SinglSDN(config-if)#ppp ipcp wins 131.108.21.70
SinglSDN(config-if)#^Z
```

Приведенное в этой главе описание конфигурирования ISDN и других служб удаленного доступа по коммутируемым каналам связи ни в коей мере не является исчерпывающим. Процесс развертывания таких служб подробно описан в комплекте руководств компании Cisco Systems, включая и аналитические обзоры, например, *Using ISDN Effectively in Multiprotocol Networks* ("Эффективное использование службы ISDN в многопротокольных сетях"), его можно найти на странице зарегистрированных сертифицированных пользователей устройств компании Cisco по адресу
www.cisco.com/univerad/cc/td/doc/cisintwk/ics/cs008.htro.

Верификация IP-взаимодействия и устранение неполадок

Рано или поздно каждый администратор выслушивает жалобы пользователей на то, что они не могут связаться с тем или иным адресатом в сети. Отсутствие связи может быть результатом сетевых неисправностей, вызванных отказами служб глобальной сети, неправильной конфигурации маршрутизаторов и других устройств в сети, внесением изменений в списки доступа (намеренными или нет), а также великого множества других причин. Хотя альтернативы оборудованию для тестирования сети, например, анализаторам протоколов, нет, маршрутизаторы тоже обладают несколькими весьма полезными инструментальными средствами для верификации IP-взаимодействия и исследования потенциальных проблем.
Как упоминалось ранее, маршрутизатор должен иметь конкретный маршрут, какой-нибудь маршрут по умолчанию или сводный маршрут до каждого пункта назначения, с которым IP-станция захочет связаться. Одним из лучших средств для поиска и устранения неисправностей является команда show ip route, которая рассматривалась ранее в этой главе. Если возникли проблемы со связью, будь то внутри или вне внутрикорпоративной сети, то первым делом следует проверить ближайший к пользователю маршрутизатор на предмет наличия в нем маршрута к IP-адресу пункта назначения. Если конкретный маршрут не найден, либо ожидаемый маршрут по умолчанию или сводный маршрут отсутствуют, необходимо исследовать протоколы динамической маршрутизации и определить, почему этого маршрута

нет. Причина может быть как вполне очевидной, скажем, отказ сегмента сети (например, из-за выхода из строя службы глобальной сети), так и скрытой, к примеру, незначительные ошибки конфигурирования другого маршрутизатора в сети.
Если установлено, что маршрут до пункта назначения существует, следует выполнить тест и определить, может ли маршрутизатор выйти на пункт назначения. Пользователи ОС UNIX знакомы с командой ping, название которой представляет собой акроним от словосочетания Packet Internet Groper ("межсетевой пакетоискатель"). Команда ping, реализованная в маршрутизаторе, использует межсетевой протокол управляющих сообщений (IP Control Message Protocol — ICMP) для отправки эхо-запросов по IP-адресу пункта назначения. Станция, получающая 1CMP-эхо-запрос, отсылает назад ICMP-эхо-ответ. Подобным образом станция-отправитель может определить, является ли достижимой станция-получатель, и приблизительно оценить, сколько времени уходит у эхо-запроса и ответа на то, чтобы добраться к станции-получателю и вернуться обратно. Ниже показан пример использования на маршрутизаторе компании ZIP SF-Core-1 команды ping для проверки достижимости маршрутизатора, находящегося в Сан-Хосе:

```
SF-Core-l#ping 131.108.100.1
Type escape sequence to abort.
Sending 5,100-byte ICMP Echos to 131.108.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 25/25/25 ms
SF-Core-ltt
```

Маршрутизатор отсылает пять ICMP-эхо-запросов и восклицательными знаками (!) сообщает, что все ответы получены. Он также показывает количество отправленных эхо-запросов и количество принятых эхо-ответов и вычисляет процент успешных пингований. Кроме того, вычисляется максимальное, среднее и минимальное время ответа.

Примечание

Когда маршрутизатор пингует IP-адрес впервые или после продолжительного перерыва, он, как правило, не принимает первый эхо-ответ, что приводит к получению только четырех из пяти ответов команды ping. Это происходит из-за того, что перед отправкой эхо-запроса маршрутизатор должен ждать разрешения IP-адреса по ARP-запросу. Обычно ARP-ответ не поступает к тому времени, когда еще можно послать первый эхо-запрос и получить ответ до истечения временного предела ожидания ответа на запрос.

В табл. 4.6 показаны различные ответные символы, которые могут быть получены в результате исполнения команды ping.

Таблица 4.6. Ответные символы команды ping

Символ	Описание	Пояснение
!	Каждый восклицательный знак указывает на получение ответа	Эхо-ответ был успешно принят
.	Каждая точка указывает, что сетевой сервер при ожидании ответа вышел за временной предел ожидания	Эхо-запрос, вероятно, дошел до пункта назначения, но пункт назначения не смог ответить или не имел обратного маршрута к отправителю запроса
U	Пункт назначения недостижим	IP-адрес пункта назначения не преобразуется в MAC-адрес или не допускает ICMP-эхо-запросов Посылающий маршрутизатор получил 1CMP-сообщение "destination unreachable" ("пункт назначения недостижим")
N	Сеть недостижима	Для заданного IP-адреса отсутствует маршрут до сети назначения Отсылающий маршрутизатор получил ICMP-сообщение "network unreachable"

		("сеть недостижима")
P	Протокол недостижим	IP-адрес пункта назначения не поддерживает ICMP-эхо-запросы Отсылающий маршрутизатор получил ICMP-сообщение "protocol unreachable" ("протокол недостижим")
Q	Запрашивается прекращение деятельности источника	IP-адрес пункта назначения принимает больше пакетов, чем он может занести в буфер Получатель отослал посылающему маршрутизатору ICMP-сообщение "source quench message" ("сообщение прекращения активности источника"), говорящее отправителю, чтобы тот "отвалил"
M	Фрагментирование не может быть произведено	Пакет вышел за пределы максимального передаваемого блока сегмента сети на пути к пункту назначения, и биту "Do Not Fragment" ("Фрагментация невозможна") присваивается значение 1 Посылающий маршрутизатор получил ICMP-сообщение "could not fragment" ("фрагментация не могла быть выполнена")
A	Пункт назначения административно недостижим	Пакет до адреса пункта назначения был отброшен, столкнувшись с фильтром пакетов или брандмауэром Посылающий маршрутизатор получил ICMP-сообщение "administratively unreachable" ("административно недостижим")
?	Пакет неизвестного типа	Посылающий маршрутизатор в ответ на запрос получил неизвестный отклик

Команда ping имеет как привилегированный, так и непривилегированный варианты. В пользовательском подрежиме режима EXEC непривилегированный вариант позволяет пользователю задавать только IP-адрес. Привилегированная версия, доступная в полнофункциональном режиме EXEC, позволяет модифицировать параметры эхо-запроса, включая количество запросов, размер отсылаемых пакетов, значение временного предела ожидания, IP-адрес источника запроса, структуру данных в эхо-запросе и многие другие величины. Ниже приводится пример привилегированного варианта команды ping, исполняемой на маршрутизаторе SF-Core-1. В этом примере в качестве адреса источника задается IP-адрес интерфейса Fast Ethernet, пунктом назначения является адрес маршрутизатора в Сан-Хосе 131.108.100.1, и размер пакета составляет 1500 байт.

```
SF-Core-l#ping
Protocol [ip]:
Target IP address: 131.108.100.1
Repeat count [5]:
Datagram size [100]:1500
Timeout  in seconds [2]:
Extended commands [n]:y
Source address or interface: 131.108.20.3
Type  of  service [0] :
Set  DF bit  in  IP header? [no]:
Validate  reply data? [no] :
Data pattern [OxABCD]:
Loose, Strict, Record, Timestamp,Verbose[none]:Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 1500-byte ICMP Echos to 131.108.100.1, timeout is 2 seconds:
i i i i i
Success   rate  is   100  percent   (5/5),   round-trip min/avg/max =  29/29/29 ms
SF-Core-l#
```

Если есть подозрение, что связь пропала из-за потери маршрута на нисходящем маршрутизаторе или неправильного пути, по которому следует пакет, то следует воспользоваться командой маршрутизатора, называемой trace, которая позволяет проверить путь следования пакета до IP-адреса пункта назначения. Функция trace аналогична функции утилиты отслеживания маршрута ОС UNIX. Как и команда ping, команда режима EXEC ОС IOS trace имеет как привилегированный, так и непривилегированный варианты. Непривилегированный вариант позволяет пользователю задавать только IP-адрес пункта назначения, а привилегированный — модифицировать параметры.
Для идентификации маршрутизаторов, стоящих на пути следования до IP-адреса пункта назначения, функция trace пользуется ICMP-сообщением "TTL-Expired" (TTL — Time To Live) ("Время жизни истекло") (TTL — "время жизни"). Маршрутизатор-отправитель посылает в пункт назначения UDP-пакет со значением параметра TTL равным 1. Первый стоящий на пути маршрутизатор принимает пакет и уменьшает значение параметра TTL на единицу. В результате время жизни истекает (принимает значение 0), и маршрутизатор не переадресовывает пакет. Вместо этого первый стоящий на пути маршрутизатор возвращает отправителю пакета ICMP-сообщение "TTL-Expired", так что теперь отправитель знает маршрутизатор первого перехода пути.
Маршрутизатор-отправитель посылает следующий UDP-пакет, но устанавливает параметр TTL в значение 2. Первый маршрутизатор по пути следования принимает пакет, уменьшает на 1 значение TTL и переадресовывает пакет второму маршрутизатору по пути следования. Второй маршрутизатор принимает пакет, уменьшает значение параметра TTL до 0 и не переадресовывает его, так как время жизни этого пакета истекло. Вместо этого он посылает станции-отправителю ICMP-сообщение "TTL-Expired", и теперь маршрутизатор-отправитель знает второй маршрутизатор, стоящий в пути следования. Этот процесс повторяется до те пор, пока пакет не дойдет до IP-адреса конечного пункта назначения. Пакет адресуется в UDP-порт с высоким номером, обычно выше 33434, который устройство пункта назначения не поддерживает. Поэтому IP-адрес пункта назначения отвечает ICMP-сообщением "Port Unreachable" ("Порт недостижим"), которое предупреждает маршрутизатор-отправитель о том, что конечный пункт назначения достигнут.
Ниже показан пример исполнения на маршрутизаторе компании ZIP SF-Core-1 команды trace с запросом пути до станции, находящейся за маршрутизатором Seoul-1:

```
SF-Core-l#trace 131.108.3.5
Type escape sequence to abort.
Tracing the route to testy.zipnet.com (131.108.3.5)
1 sO/0-SanJose-sj.zipnet.com (131.108.240.2) 25 msec 25 msec 25 msec
2 sl-Seoull-kr.zipnet.com (131.108.241.2) 176 msec *176 msec
3 testy.zipnet.com (131.108.3.5) 178 msec 178 msec 178 msec
SF-Core-l#
```

В приведенном выше примере после имени и IP-адреса маршрутизаторов, стоящих по пути в сети, выводятся значения времени. Эти значения представляют собой приблизительное время, затрачиваемое на передачу и получение подтверждения на переходе между адресом отправителя и маршрутизатором, стоящим в пути следования. Для каждого IP-адреса пункта назначения выводится до трех значений времени: по одному для каждого из трех зондирующих пакетов. Некоторые устройства имеют ограничения по скорости, с которой они способны отвечать ICMP-сообщениями. Для таких устройств может стоять менее трех значений. В этом случае для каждого зондирующего пакета, на который устройство не отвечает из-за ограничений по скорости ответов, вместо значения времени ставится звездочка. Пример этого можно видеть в показанном выше результате исполнения команды. Маршрутизатор второго перехода пути не смог ответить на второй зондирующий пакет, что и показано звездочкой. Устройства, работающие под управлением ОС IOS компании Cisco, имеют ограничение по скорости ICMP-ответов — один в секунду.
Есть и другие причины, кроме ограничения по скорости отсылки ICMP-сообщений, по которым некоторые маршрутизаторы в пути следования могут не отвечать ICMP-сообщением

"TTL-Expired". Некоторые могут повторно использовать параметр TTL входящих пакетов, что приводит к истечению срока жизни ICMP-сообщения до того, как оно сможет вернуться отправителю. А в некоторых случаях фильтрация пакетов может не дать пакетам ICMP-ответа дойти до маршрутизатора-отправителя. Во всех этих случаях в строке результата вместо адресной информации появляется строка звездочек. В показанном ниже примере результата исполнения команды trace второй маршрутизатор по пути следования не смог ответить на запросы команды trace:

```
SF-Core-l#trace 131.108.3.5
Type escape sequence to abort.
Tracing the route to testy.zipnet.com (131.108.3.5)
1 sO/0-SanJose-sj.zipnet.com (131.108.240.2) 25 msec 25 msec 25 msec
2 * * *
3 testy.zipnet.com (131.108.3.5) 178 msec 178 msec 178 msec
SF-Core-l#
4
```

В привилегированном варианте команда trace позволяет регулировать параметры команды, определяя, выполнять или нет обратное разрешение IP-адресов в имена хост-машин, количество зондирующих пакетов, посылаемых на каждом TTL-шаге, минимальное и максимальное значение параметра TTL и так далее. Ниже приведен предыдущий пример исполнения команды trace, но в привилегированном варианте. Здесь выводятся только числовые ответы.

```
SF-Core-l№trace
Protocol tip]:
Target IP address: 131.108.3.5
Source address:
Numeric display [n]: y
Timeout in seconds [3] :
Probe count [3]:
Minimum Time to Live [1] :
Maximum Time to Live [30]:
Pert Number [33434]:
Loose, Strict, Record, Timestamp, Verbose [none]:
Type escape sequence to abort.
Tracing the route to 131.108.3.5
1 131.108.240.2 25 msec 25 msec 25 msec
2 131.108.241.2 176 msec * 176 msec
3 131.108.3.5 178 msec 178 msec 178 msec
SF-Core-l#
```

Если станция, которая достижима через непосредственно подключенный интерфейс локальной сети, не отвечает, то причиной может быть то, что маршрутизатор не может преобразовать IP-адрес в MAC-адрес. Чтобы проверить MAC-адреса, которые маршрутизатор смог преобразовать, используется команда ОС IOS режима EXEC show ip arp Эта команда воспринимает в качестве параметра либо конкретный IP-адрес, либо конкретный интерфейс, либо конкретный 48-разрядный MAC-адрес и выводит на экран только ARP-записи для этого параметра. Если параметр не вводится, то на экран выдаются все IP ARP-записи. В выводимую командой информацию входит IP-ARP-отображение, возраст записи в таблице и интерфейс, с которым связана ARP-запись (По умолчанию маршрутизатор удаляет ARP-записи из ARP-таблицы по истечении четырех часов.) Ниже приведен пример исполнения команды show ip arp на маршрутизаторе компании ZIP SF-Core-1:

```
SF-Core-l#show ip arp
Protocol Address          Age(min)    Hardware Addr     Type I    Interface
Internet 131.108.20.      -           0000.Oc07.b627    ARPA      FastEthernetO/0
Internet  131.108.20.2     4           0000.Oc67.b62c    ARPA      FastEthernetO/0
Internet  131.108.20.4     2           0000.Ocf1.a9cl    ARPA      FastEthernetO/0
Internet  131.108.20.11    2           0000.Ocb8.02bc    ARPA      FastEthernetO/0
Internet  131.108.20.99    0           Incomplete        ARPA
SF-Core-l#
```

В примере выше запись ARP-таблицы для адреса 131.108.20.99 имеет вместо фактического аппаратного MAC-адреса слово *incomplete* (не заполнено), которое указывает на то, что маршрутизатор послал ARP-запрос, но ответ для занесения записи в ARP-таблицу не был получен. В подобной ситуации можно предположить, что либо не существует станции с таким адресом, либо станция не способна отвечать, может быть, по причине отключенного питания.
Полные статистические данные о работе протокола IP на маршрутизаторе могут быть получены с помощью команды show ip traffic. Эти данные включают результаты счета общего количества пакетов, приятных и посланных маршрутизатором, количество принятых и посланных широковещательных пакетов, статистику работы ICMP/UDP/TCP-протокола и многое другое. Такие статистические данные могут помочь определить, посылал или получал маршрутизатор ICMP-эхо-пакеты, были ли случаи неудач при преобразовании IP-адреса в MAC-адрес (известные как отказ инкапсуляции), и где пакеты, принадлежащие определенному протоколу маршрутизации, принимаются или посылаются. Результаты счета в команде show ip traffic носят коммулятивный характер и обнуляются, только когда маршрутизатор перезагружается или выключается-включается. Ниже показан пример информации, выводимой командой show ip traffic на маршрутизаторе компании ZIP SF-Core-1:

```
SF-Core-l#show ip traffic
IP statistics:
   Rcvd:  4686565 total, 2623438 local destination
          0 format errors, 0 checksum errors, 77 bad hop count 0 unknown protocol,
          1 not a gateway 0 security failures, 0 bad options, 0 with options
   Opts:  0 end, 0 пор,О basic security, 0 loose source route 0 timestarap, 0
   extended
          security, 0 record route 0 stream ID,0 strict source route,0 alert,0
          other
   Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
          0 fragmented,0 couldn 't fragment
   Bcast: 5981 received,0 sent
          Mcast: 2482184 received,3581861 sent
          Sent:  3893477 generated,2062048 forwarded
          954 encapsulation failed,208 no route
   ICMP statistics:
      Rcvd:  0 format errors, 0 checksum errors, 5  redirect, 5070 unreachable
          3 echo, 16 echo reply, 0 mask requests
          0 parameter,0 timestamp,0 info request,0 other
          0 irdp solicitations,0 irdp advertisements
      Sent:  0 redirects, 18050 unreachable, 66 echo, 3 echo reply
          0 mask requests, 0 mask replies, 0 quench, 0 timestamp
          0 info reply, 7 time exceeded, 0 parameter problem
          0 irdp solicitations,0 irdp advertisements
   UDP statistics:
      Rcvd: 52836 total, 4 checksum errors, 18085 no port
      Sent: 50699 total, 5949 forwarded broadcasts
   TCP statistics:
      Rcvd:  47895 total,0 checksum errors,1 no port
      Sent:  46883 total
   Probe statistics:
      Rcvd:  0 address requests,0 address replies
             0 proxy name requests,0 where-is requests,0 other
      Sent:  0 address requests, 0 address replies (0 proxy)
             0 proxy name replies, 0 where-is replies
   EGP statistics:
      Rcvd:  0 total, 0 format errors, 0 checksum errors,
      Sent:  0 total
   JGRP statistics:
      Rcvd:  0 total, 0 checksum errors
      Sent:  0 total
   OSPF statistics:
      Rcvd:  0 total,0 checksum errors
             0 hello, 0 database desc, 0 link state req
```

```
          0 link state updates, 0 link state acks
     Sent:  0 total
  IP-IGRP2 statistics:
     Rcvd:  2105381 total
     Sent:  3140121 total
  PIMv2 statistics: Sent/Received
     Total: 0/0, 0 checksum errors, 0 format errors
     Registers: 0/0, Register Stops:
  IGMP statistics: Sent/Received
     Total: 0/0, Format errors: 0/0,
     Host Queries: 0/0, Host Reports:
     DVMRP: 0/0, PIM: 0/0
  ARP statistics:
     Rcvd:  8540 requests, 4 replies,0 reverse, 0 other
     Sent: 89 requests, 9018 replies (0 proxy), 0 reverse
SF-Core-1#
```

Счетчики, стоящие в команде show ip traffic, выполняют подсчет как произошедших событий, так и типов пакетов, которые отсылались и принимались. Если показания счетчика отказов инкапсуляции увеличиваются, то это означает, что маршрутизатор не принимал ARP-ответы на свои ARP-запросы для пакетов, попытка коммутации которых на интерфейс пункта назначения выполнялась, и эти пакеты были отброшены. Результат счета ICMP-эхо-пакетоз показывает количество сгенерированных маршрутизатором пингов, а счетчик эхо-ответов говорит о количестве пингов, на которые он ответил.

Кроме команд поиска и устранения неисправностей и верификации, представленных в данном разделе, в ОС IOS в режиме EXEC существует множество отладочных команд debug, которые помогают оценить функционирование протокола IP в маршрутизаторе. Эти команды debug обеспечивают вывод как обшей, так и более подробной диагностической информации, которая может помочь в решении проблем при устранении неполадок и при проверке работы маршрутизатора, протоколов маршрутизации и других функций. Некоторые из команд debug, наиболее часто используемых для протокола TCP/IP, сведены в табл. 4.7.

Таблица 4.7. Отладочные команды для протокола IP

Команда	Описание
debug ip routing	Выводит изменения, которые возникают в таблице маршрутизации в результате добавлений или удалений маршрутов
debug ip packet	Показывает IP-адреса источника и пункта назначения пакетов, которые проходят через маршрутизатор. Эта команда debug может перегрузить маршрутизатор, поэтому при ее применении следует соблюдать осторожность. Для ограничения нагрузки на центральный процессор рекомендуется совместно с этой командой использовать список доступа
debug ip udp	Выводит информацию о посланных маршрутизатору UDP-пакетах
debug ip icmp	Выдает информацию об ICMP-сообщениях, посланных маршрутизатору и сгенерированных им
debug ip arp	Выводит информацию о ARP-запросах, сгенерированных маршрутизатором, и о посланных ему ответах

В состав отладочных команд для различных протоколов маршрутизации входят команды debug ip rip, debug ip eigrp, debug ip igrp, debug ip ospf и debug ip bgp. Каждая из этих команд имеет варианты параметров, которые управляют выводимой пользователю отладочной информацией о протоколе маршрутизации. Следует проявлять осторожность при использовании некоторых вариантов тех команд, которые интенсивно потребляют ресурсы центрального процессора. Полное описание всех отладочных команд и примеры выводимой информации можно найти на компакт-диске Cisco Connection Documentation (Документация по организации связи с помощью устройств компании Cisco) или в его интерактивной версии ПО адресу www. Cisco. com/univercd/home/home. htm.

Совет

Не исполняйте команды debug, увеличивающие нагрузку на центральный процессор, на консольном порту. Вместо этого деактивируйте режим протоколирования консоли командой глобального конфигурирования ОС IOS no logging console и активируйте командой глобального конфигурирования logging bufferd режим буферизации журнала. После этого исполняйте команду из сеанса виртуального терминала и из этого же сеанса просматривайте результат. Если сеанс перестает отвечать, то режим отладки может быть отменен через консоль, так как консольный сеанс имеет более высокий приоритет, чем сеанс виртуального терминала. Результаты отладки можно будет затем просматривать в буфере журнала с помощью команды ОС IOS режима EXEC show log. Если разрешено ведение системного журнала, то результат также можно просматривать и в журнальном файле на сервере системного журнала.

Конфигурирование других опций протокола IP

Операционная система IOS, под управлением которой работают маршрутизаторы компании Cisco и другие устройства, обладает десятками функций, призванных помочь в управлении сетью и самим маршрутизатором. В данном разделе рассмотрены четыре наиболее часто реализуемые на маршрутизаторе функции, которые улучшают работу сети и облегчают использование самого маршрутизатора.

Конфигурирование служб имен доменов

В современных TCP/IP-сетях большинство пользователей обращаются к серверам принтерам, рабочим станциям и другим IP-устройствам, используя их имена, а не IP-адреса. Поэтому где-нибудь в рамках внутрикорпоративной сети организации ставятся серверы, которые преобразовывают имена в IP-адреса. Такие серверы называются серверами службы доменных имен DNS. Маршрутизаторы могут пользоваться DNS-системой для преобразования имен в IP-адреса, помогая уменьшить количество IP-адресов, которые администратор сети должен держать в памяти.

Обычно функция DNS активирована в ОС IOS. Однако, если она была отключена, ее можно включить снова с помощью команды глобального конфигурирования ОС IOS ip domain-lookup. После активизации функции DNS в работающем под управлением ОС IOS устройстве должно быть сконфигурировано имя домена, в котором оно размещается, и IP-адрес сервера имен службы DNS, которым оно сможет пользоваться для преобразования имен. Имя домена может быть сконфигурировано командой глобального конфигурирования ОС IOS ip domain-name. DNS-серверы имен включаются в конфигурацию с помощью команды глобального конфигурирования ОС IOS ip name-server. Команда ip name-server имеет параметром один или несколько IP-адресов серверов имен. Если работающее под управлением ОС IOS устройство размещено сразу в нескольких DNS-доменах, то, чтобы задать перечень имен доменов, куда должны направляться неквалифицированные имена, воспользуйтесь командой глобального конфигурирования ОС IOS ip domain-list.

Ниже приведен пример конфигурирования службы DNS маршрутизатора компании ZIP SF-Core-1. В этом примере имя домена — zipnet.com, а IP-адреса сервера имен — 131.108.110.34 И 131.108.110.35.

```
SF-Core-1# configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-Core-1(config)#ip domain-lookup
SF-Core-1(config)#ip domain-name zipnet.com
SF-Core-1(config)#ip domain-list zipnet.com
SF-Core-1(config)#ip domain-list zipnet.net
SF-Core-1(config)#ip name-server 131.108.110.34 131.108.110.35
SF-Core-1(config)#^Z
```

Верификация установок службы DNS на маршрутизаторе может выполняться с помощью команды ОС IOS режима EXEC show host. Кроме того, команда show host выводит список хост-машин, имена которых имеют отображение на IP-адреса, а также возраст каждой записи. Ниже приводится результат исполнения команды show host на маршрутизаторе компании ZIP SF-Core-1:

```
SF-Core-l#show host
Default domain is zipnet.com
Domain list: zipnet.com, zipnet.net
Name/address lookup uses domain service
Name servers are 131.108.110.34, 131.108.110.35
Host                              Flags       Age         Type    Address(es)
testy.zipnet.com                  (temp,OK)     1         IP      131.108.3.5
sl-Seoull-kr.zipnet.com           (temp,OK)     1         IP      131.108.241.2
sO/0-SanJose-s].zipnet.com        (temp,OK)     1         IP      131.108.240.2
SF-Core-l#
```

Преобразования имен хост-машин в IP-адреса могут также конфигурироваться на маршрутизаторе статически. Это бывает необходимо, если DNS-серверы не доступны, нужно создать имена, которые отличаются от занесенных в службу DNS, или задать отображение на IP-адреса для отдельных терминальных портов сервера. Статическое отображение имени в IP-адрес конфигурируется с помощью команды глобального конфигурирования ОС IOS ip host. Параметрами этой команды являются имя хост-машины, альтернативный порт протокола Telnet и один или несколько IP-адресов, в которые может преобразовываться имя хост-машины. Ниже показан пример статического отображения нескольких различных имен хост-машин в IP-адреса на маршрутизаторе компании ZIP SF-Core-1:

```
SF-Core-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands,one per line.End with CNTL/Z.
SF-Core-1(config)lip host grouchy 131.108.3.5
SF-Core-1(config)tip host grouchy-console 2001 131.108.3.50
SF-Core-1(config)#ip host farout 131.108.3.88 131.108.3.150
SF-Core-1(config)#^Z
```

Статические отображения имен хост-машин в IP-адреса также могут верифицироваться с помощью команды show host. Ниже приводится результат ввода статических отображений имен хост-машин в IP-адреса для маршрутизатора компании ZIP SF-Core-1:

```
SF-Core-l#show host
Default domain is zipnet.com
Domain list: zipnet.com, zipnet.net
Name/address lookup uses domain service
Name servers are 131.108.110.34, 131.108.110.35
Host                          Flags       Age         Type        Address(es)
testy.zipnet.com              (temp,OK)   1           IP          131.108.3.5
sl-Seoull-kr.zipnet.com       (temp,OK)   1           IP          131.108.241.2
sO/0-SanJose-sj.zipnet.com    (temp,OK)   1           IP          131.108.240.2
grouchy                       (perm,OK)   2           IP          131.108.3.5
grouchy-console               (perm,OK)   2           IP          131.108.3.50
farout                        (perm,OK)   2           IP          131.108.3.88
                                                                  131.108.3.150
SF-Core-1#
```

Статические записи в таблице имен хост-машин можно отличить от записей, полученных из службы DNS, по полю Flags (флаги) записи для конкретного имени хост-машины. Тип флага temp (временная запись) говорит о том, что имя было получено динамически из службы DNS и через определенный период времени будет удалено из таблицы по возрасту. Тип флага perm (постоянная запись) указывает на то, что имя было сконфигурировано статически и никогда не будет удалено из таблицы по возрасту.

Временные записи в таблице IP-хосr-машин удаляются с помощью команды ОС IOS

режима EXEC clear host. Отображения отдельных имен хост-машин могу удаляться путем указания в качестве параметра команды имени хост-машины. Все временные записи хост-машин удаляются при вводе в качестве параметра звездочки. Ниже приводится пример удаления на маршрутизаторе компании ZIP SF-Core-записи отображения имени хост-машины в IP-адрес для хост-машины с HMCHBIV testy.zipnet.com:

```
SF-Core-l#clear host testy.zipnet.com
SF-Core-l#
```

Переадресация широковещательных IP-пакетов

Одним из преимуществ маршрутизаторов является ограничение распространена широковещательных IP и MAC-пакетов в пределах локального сегмента сети. Большинство широковещательных рассылок используются для запроса информации о неизвестном MAC-адресе для IP-адреса (ARP-запросы) в локальном сегменте сети, и поэтому изоляция таких широковещательных пакетов в пределах локального сегмент сети не приводит к неизбежным (в противном случае) проблемам и в высшей степени выгодна для производительности сети.

В некоторых ситуациях IP-станции используют широковещательные UDP-посылки, чтобы найти службы, которые могут находиться не в локальном сегменте сети. Например, прикладные программы, которые работают с NetBIOS через протокол IP, используют широковещательные UDP-рассылки для поиска необходимой пользователю службы конкретного типа. Если эта служба размещена не в том сегменте сети в котором находится станция пользователя, то маршрутизатор заблокирует эту рассылку, делая службу недоступной. Другие службы, например, DHCP или протокол начальной загрузки (Bootstrap Protocol — BOOTP), посылают широковещательные UDP-пакеты, чтобы помочь IP-станциям определить их IP-адреса во время процесса начальной загрузки; эти широковещательные пакеты принимаются серверами, назначающими адреса. Если серверы находятся вне локального сегмента сети, то IP-станция не сможет получить назначенный сервером IP-адрес.

Чтобы компенсировать свойство маршрутизатора по изолированию широковещательных пакетов, ОС IOS имеет возможность переадресовывать широковещательные UDP-пакеты на конкретную хост-машину или в конкретную подсеть. Эта функция, известная под названием *переадресация широковещательных IP-пакетов,* активируется путем использования субкоманды конфигурирования интерфейса ОС IOS ip helper-address и команды глобального конфигурирования ОС IOS ip forward-protocol. Общее назначение этих команд состоит в переадресации DHCP-запросов адреса из локального сегмента сети в тот сегмент сети, в котором размешается DHCP-сервер, что показано на рис. 4.11. Исследуем применение функции переадресации широковещательных пакетов на примере маршрутизатора SF-2, находящегося на площадке компании ZIP в Сан-Франциско.

В сети компании ZIP в Сан-Франциско за маршрутизатором SF-2 станции, работающие под ОС Microsoft Windows 95/98, NT и Windows 2000, используют протокол DHCP для динамического получения своих IP-адресов. Эти рабочие станции находятся в сегментах локальной сети Ethernet 0 и Ethernet 1 маршрутизатора SF-2. DHCP-сервер находится в сегменте локальной сети Fast Ethernet 0. Широковещательные пакеты из сегментов Ethernet не проходят через маршрутизатор, и, таким образом, широковещательные DHCP-пакеты не попадают в сегмент Fast Ethernet и на DHCP-сервер. Чтобы разрешить переадресацию широковещательных пакетов в отношении сегментов Ethernet, из которых маршрутизатор принимает эти широковещательные пакеты, используется команда ip helper-address. В качестве параметра команды ip helper-address выступает IP-адрес хост-машины или широковещательный IP-адрес. Указываемый адрес представляет собой либо адрес хост-машины конкретного DHCP-сервера, либо широковещательный адрес сегмента локальной сети, в котором находится DHCP-сервер.

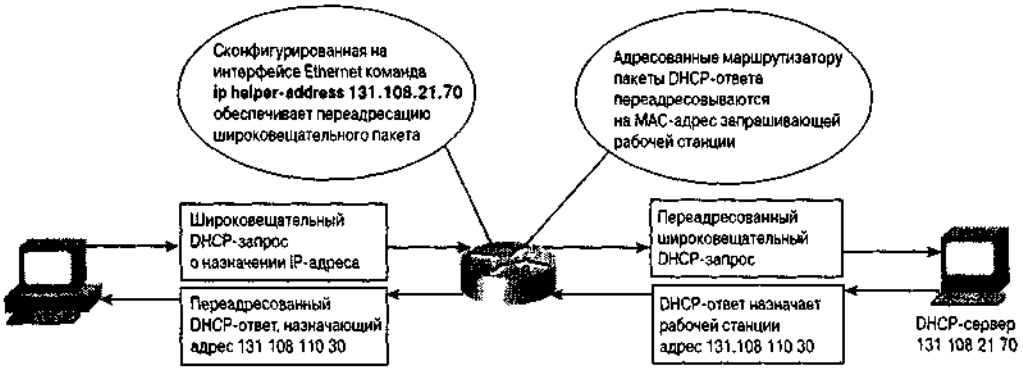


Рис. 4.11. Широковещательные пакеты DHCP-запроса переадресовываются с использованием адреса помощника

Ниже приведен пример использования на маршрутизаторе компании ZIP SF-2 команды ip helper-address, в результате чего широковещательные пакеты переадресовываются непосредственно DHCP-серверу по адресу 131.108.21.70.

```
SF-2#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-2(config)#interface ethernet 0
SF-2(config-if)#ip helper-address 131.108.21.70
SF-2(config)#^Z
```

Вместо прямой переадресации DHCP-серверу широковещательные пакеты могут переадресовываться в сегмент локальной сети, в котором DHCP-сервер размещается. Такая альтернатива полезна в тех случаях, когда на запрос может отвечать не один DHCP-сервер. Ниже дан пример использования в качестве пункта назначения переадресации широковещательного IP-адреса сегмента сети, в котором находится DHCP-сервер:

```
SF-2 # configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
SF-2(config)#interface ethernet 1
SF-2(config-if)#ip helper-address 131.108.23.255
SF-2(config-if)#^Z
```

Команда ip helper-address определяет, куда должны переадресовываться широковещательные пакеты. Команда ip forward-protocol указывает, какие широковещательные UDP-пакеты переадресовываются. Как только в отношении интерфейс используется команда ip helper-address, по умолчанию переадресовывается НІ сколько типов широковещательных UDP-пакетов.

- Пакеты простейшего протокола передачи файлов (TFTP) (порт 69).
- Пакеты системы именования доменов (порт 53).
- Пакеты службы времени (порт 37).
- Пакеты сервера имен NetBIOS (порт 137).
- Пакеты сервера дейтаграмм NetBIOS (порт 138).
- Пакеты клиента протокола начальной загрузки (BOOTP) и серверных дейтаграмм (порты 67 и 68).
- Пакеты службы TACACS (порт 49).

Если есть приложение, которое выполняет широковещание на порт, отличающихся от перечисленных, и эти широковещательные пакеты должны переадресовываться то используется команда ip forward-protocol, которой задается конкретный тип широковещательных пакетов, подлежащий включению в состав переадресовываемы При добавлении ключевого слова по эта команда может быть использована для запрещения переадресации пакетов любого из протоколов по умолчанию. Команда ip forward-protocol имеет параметрами тип переадресации, который должен выполняться (например UDP), и

конкретный номер порта протокола, подлежащего переадресации. Ниже показан пример использования команды для разрешения переадресации широковещательных пакетов, поступающих на UDP-порт 1965, и запрещения переадресации пакетов сервера имен NetBIOS и сервера дейтаграмм в маршрутизаторе компании ZIP SF-2:

```
SF-2#configure
Configuring from terminal, memory, or network [terminal]? Enter
configuration commands, one per line. End with CNTL/Z. SF-2(config)#ip
forward-protocol udp 1965
SF-2(config)#no ip forward-protocol udp 137
SF-2(config)#no ip forward-protocol udp 138
SF-2(config)#^Z
```

Конфигурации, созданные командой ip helper-address, можно проверить с помощью команды show ip interface.

Дополнительная справка: другие приложения с широковещательной рассылкой

Рассматриваемая в этом разделе техника переадресации широковещательных пакетов проектировалась для удовлетворения потребностей ограниченных сред с переадресацией широковещательных пакетов. Она хорошо подходит для решения таких задач, как переадресация запросов IP-адреса серверу или группе серверов в рамка> протоколов DHCP или BOOT, при этом последние размещаются в центральном узле сети. Но существуют другие приложения, для которых может потребоваться более солидная технология переадресации. Такие приложения, как периодические биржевые сводки ("тиккер"), обычно используют широковещательные рассылки для поставки информации большой группе пользовательских рабочих станций, размещенных в большой части сети. Подобные приложения не очень хорошо подходят под модель с адресом помощника. Для предотвращения затопления центрального процессора маршрутизатора трафиком широковещательных пакетов и репликацией им требуются более совершенные методики, например, лавинная UDP-рассылка и репликация широковещательной рассылки в многоадресную. В настоящее время компания Cisco Systems располагает аналитическим материалом, в котором обсуждаются практические реализации моделей адреса помощника и лавинной адресации. Его можно найти на странице зарегистрированных сертифицированных пользователей устройств компании Cisco по адресу www.Cisco.com/univercd/cc/td/doc/cisintwk/ics/cs006.htm.

Динамическое назначение адресов с помощью DHCP-сервера ОС IOS

В предыдущем разделе обсуждалось приложение для переадресации широковещательных IP-пакетов — переадресация DHCP-запросов назначения адресов. Когда маршрутизатор переадресует такие запросы назначения адреса, говорят, что он действует в качестве DHCP-агента-ретранслятора. Роль агента-ретранслятора DHCP заключается в том, чтобы принимать широковещательные пакеты локальной сети с запросом о назначении адреса и переадресовывать их ранее идентифицированному DHCP-серверу. DHCP-сервер обычно представляет собой рабочую станцию или сервер, например, работающую под UNIX или Windows NT систему, на которой исполняется программное обеспечение сервера DHCP или соответствующей службы. Кроме того, источником динамически назначаемых адресов может быть работающий под управлением ОС 1OS маршрутизатор или сервер доступа.

DHCP-сервер ОС IOS работает аналогично DHCP-серверу на рабочей станции, принимая запросы/обновления назначения адресов и назначая адреса из предварительно заданных групп адресов, называемых *пулами*. Пулы адресов могут предоставлять запрашивающему клиенту дополнительную информацию, например, 1P-адрес(а) DNS-сервера(ов), маршрутизатор по умолчанию и другую полезную информацию. DHCP-сервер ОС IOS может принимать широковещательные пакеты из локальных сегментов сети или DHCP-запросы, которые были переадресованы другими агентами-ретрансляторами DHCP, находящимися в пределах сети.

Примечание
Кроме DHCP-сервера, в составе ОС IOS компанией Cisco Systems также выпускается DNS- и DHCP-сервер, называющийся *Cisco Network Registrar* и ориентированный на рабочие станции, работающие под ОС Solaris, HP-UX и Microsoft Windows. Решение об использовании DHCP-сервера в составе ОС IOS или ориентированного на рабочие станции зависит от многих факторов: размера сети, количества узлов, требующих динамически назначаемых адресов, частоты запросов адресов и их обновлений, необходимости в резервном дублировании и стоимости. В общем случае DHCP-сервер в составе ОС IOS наиболее практичен в малых и средних по размеру сетях или в децентрализованной модели со множеством удаленных офисов. Ориентированные на рабочую станцую DHCP-серверы больше подходят для больших организаций с потребностью в резервном дублировании и с высокоцентрализованной схемой управления.

DHCP-сервер из состава ОС IOS обычно участвует в двух этапах процесса назначения адреса: DHCPOFFER и DHCPACK. На рис. 4.12 изображены основные этапы процесса запроса адреса DHCP-клиентом у DHCP-сервера. DHCP-клиент посылает широковещательное сообщение DHCPDISCOVER, пытаясь найти DHCP-сервер. DHCP-сервер предлагает параметры присвоения адреса клиенту в своем одноадресном ответ DHCPOFFER. После этого DHCP-клиент возвращает DHCP-серверу широковещательный формальный запрос на предложенное назначение адреса DHCPREQUEST. DHCP сервер в ответ посылает одноадресный ответ DHCPACK, говорящий, что запрошенные адреса назначены клиенту. Четыре этапа, показанные на рис. 4.12, отображают нормальный процесс переговоров о назначении адреса, когда нет ошибок или конфликте] Полный процесс назначения адреса, включая обработку сообщений отклонения адрес DHCPDECLINE, описывается в Запросе на комментарий № 2131 "Dynamic Ho; Configuration Protocol" ("Протокол динамического конфигурирования хост-машины").

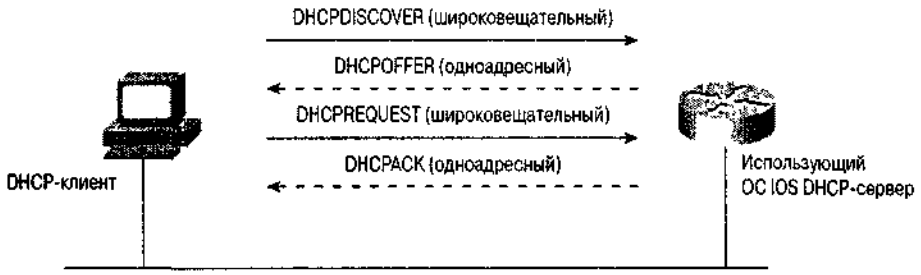


Рис. 4.12. DHCP-процесс назначения адреса DHCP-сервером

В работающем под управлением ОС IOS маршрутизаторе или сервере доступа активация функции DHCP-сервера выполняется в четыре этапа.

- Идентификация места, где будет выполняться протоколирование информации процесса назначения адресов.
- Создание списка IP-адресов, которые исключаются из процесса динамического назначения.
- Создание пула адресов, использующихся для динамического назначения.
- Введение в пулы адресов дополнительных атрибутов, которые будут предоставляться запрашивающим станциям.

Рассмотрим конфигурирование DHCP-сервера в составе ОС IOS на примере маррутизатора компании ZIP в Куала-Лумпур.

Первым шагом в активации DHCP-сервера ОС IOS является конфигурирован* места в сети, где будет производиться протоколирование и хранение DHCF назначений адресов (так называемая *привязка).* Как правило, это рабочая станция ил сервер, которые поддерживают один из следующих протоколов передачи файле TFTP, FTP или RCP. Задание этого места позволяет перезапускать маршрутизатор или сервер доступа без потери информации о том, какие адреса какому DHCF клиенту были выделены. Кроме того, здесь же будет выполняться протоколирован* конфликтов назначения адресов, которые могут возникать в процессе DHCP-переговоров. Для

задания местоположения используется команда глобального конфигурирования ОС IOS ip dhcp database. В качестве параметра этой команды выступает унифицированный указатель ресурсов (URL), задающий адрес сервера и имя файла, используемые для протоколирования. Данная команда конфигурирования может повторяться несколько раз, чтобы обеспечить возможность хранения привязок на нескольких серверах. Ниже показан пример конфигурирования в качестве местоположения DHCP-базы данных журнала маршрутизатора компании ZIP в Куала-Лумпур сервера с IP-адресом 131.108.2.77 и файла с именем kl-dhcp-info. В качестве протокола используется протокол TFTP.

```
Kuala-Lumpur#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Kuala-Lumpur(config)#ip dhcp database tftp://131.108.2.77/kl-dhcp-info
Kuala-Lumpur(config)#^Z
```

Во время процесса назначения адреса DHCP-сервер ОС IOS пытается гарантировать, что предлагаемый адрес не находится в пользовании. С этой целью он до отправки ответа DHCP-клиенту посылает по предлагаемому адресу серию ping-пакетов. Если адрес уже используется, то он протоколируется как конфликтный и не предлагается до тех пор, пока администратор сети не разрешит конфликт.

Если сервер для протоколирования привязок DHCP-адресов не доступен, и команда ip dhcp database не конфигурируется, то необходимо отключить и функцию протоколирования DHCP-конфликтов. Отключение функции протоколирования конфликтов осуществляется с помощью команды глобального конфигурирования ОС IOS no ip dhcp conflict logging. Узел сети компании ZIP в Куала-Лумпур имеет TFTP-сервер, но ниже показан пример отключения функции протоколирования DHCP-конфликтов, если бы его не было:

```
Kuala-Lumpur#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Kuala-Lumpur(config)#no ip dhcp conflict logging
Kuala-Lumpur(config)#^Z
```

После задания места для протоколирования привязок создается список адресов, которые должны быть исключены из состава динамически предлагаемых назначений. Такой список включает адрес маршрутизатора или маршрутизаторов в данном диапазоне адресов, любые статически назначенные адреса или адрес, который может быть резервным и не должен предлагаться DHCP-клиенту. Для создания такого списка используется команда глобального конфигурирования ОС IOS ip dhcp excluded-address. В качестве параметра команда воспринимает либо отдельный подлежащий исключению IP-адрес, либо пару адресов, которые представляют собой начальный и конечный адреса диапазона IP-адресов. При конфигурировании команда может повторяться несколько раз, что необходимо в случаях, когда надо исключить несколько адресов, не следующих непрерывно один за другим или входящих в несколько пулов IP-адресов для назначения. Ниже показан пример исключения на маршрутизаторе компании ZIP в Куала-Лумпуре диапазона IP-адресов с 131.108.2.1 по 131.108.2.10 и отдельного IP-адреса 131.108.2.57:

```
Kuala-Lumpur#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Kuala-Lumpur(config)#ip dhcp excluded-address 131.108.2.1 131.108.2.10
Kuala-Lumpur(config)#ip dhcp excluded-address 131.108.2.57
Kuala-Lumpur(config)#^Z
```

Последним шагом в активации DHCP-сервера ОС IOS является задание пулов IP-адресов, которые будут использоваться для динамического предоставления адресов. Как минимум, пул DHCP-адресов задает диапазон адресов (без исключенных адресов), которые будут предлагаться

запрашивающим адрес DHCP-клиентам. Если к выступающему в роли DHCP-сервера маршрутизатору или серверу доступа подключено несколько сегментов локальной сети или если он обслуживает адресами несколько сетевых сегментов, находящихся где-нибудь в другом месте корпоративной сети, то на DHCP-сервере ОС IOS могут быть сконфигурированы несколько пулов. Задает пул адресов для назначения команда глобального конфигурирования ОС IOS ip dhcp pool. В качестве параметра команда воспринимает либо произвольную цепочку символов, описывающую пул, либо целое число. После задания названия пула из режима субкоманд DHCP-конфигурирования, о чем свидетельствует появление в командной строке запроса на ввод (config-dhcp) #, вводятся дополнительные команды конфигурирования пулов адресов. Ниже приведен пример конфигурирования DHCP-пула адресов с названием kl-users на маршрутизаторе в Куала-Лумпуре с переводом сетевого администратора в режим субкоманд DHCP-конфигурирования для продолжения конфигурирования пула адресов.

```
Kuala-Lumpur#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Kuala-Lumpur(config)#ip dhcp pool kl-users
Kuala-Lumpur(config-dhcp)#^Z
```

Для задания диапазона адресов, которые данный пул адресов будет предлагать DHCP-клиентам, используется субкоманда DHCP-конфигурирования ОС IOS network. Субкоманда network требует двух параметров: сетевого IP-адреса и сетевой маски или маски в формате с контрольной суммой. Задаваемые для пула сетевой адрес и маска должны соответствовать сетевому адресу и маске сегмента локальной сети, для которого этот пул будет предлагать адреса. Если DHCP-сервер должен поставлять адреса для нескольких сегментов локальной сети, задаются отдельные пулы, каждый со своей субкомандой network, содержащей адрес и маску, соответствующие этому сегменту локальной сети. Ниже приводится пример конфигурирования на маршрутизаторе в Куала-Лумпуре пула DHCP-адресов kl-users с использованием субкоманды network, задающей диапазон адресов, которые будут назначаться DHCP-клиентам (отметим использование вместо сетевой маски 255.255.255.128 маски в формате с контрольной суммой /25):

```
Kuala-Lumpur #configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Kuala-Lumpur(config)#ip dhcp pool kl-users
Kuala-Lumpur(config-dhcp)#network 131.108.2.0 /25
Kuala-Lumpur(config-dhcp)#^Z
```

В данном примере задание сети 131.108.2.0 с маской в формате с контрольной суммой /25 означает, что DHCP-клиентам будут предлагаться адреса в диапазоне от 131.308.2.1 до 131.Ю8.2.128 (без ранее исключенных адресов). При просмотре исполняемого или запускающего конфигурационного файла маска в формате с контрольной суммой /25 будет преобразована в сетевую маску 255.255.255.128.

Существуют дополнительные субкоманды DHCP-конфигурирования, которые позволяют сетевому администратору конфигурировать DHCP-сервер ОС IOS на поставку дополнительной информации DHCP-клиенту, используя процесс переговоров об адресах. Обычно дополнительная информация представляет собой адрес или адреса маршрутизатора по умолчанию для клиента в данном сегменте локальной сети, адреса DNS-серверов, адреса NetBIOS/WINS-серверов. Дополнительной может быть любая информация, которая в противном случае должна была бы конфигурироваться пользователем или сетевым администратором вручную. Ниже приведен перечень самых распространенных субкоманд DHCP-конфигурирования:

- domain-name задает имя DNS-домена, к которому будет принадлежать клиент;
- dns-server определяет один или несколько IP-адресов DNS-серверов, которым клиент может посылать запросы о преобразовании имен в IP-адреса;
- netbios-name-server задает IP-адреса NetBIOS/WINS-серверов, которым NetBIOS-клиенты (обычно это рабочие станции, работающие под управлением ОС Windows) могут

посылать запросы относительно местонахождения ресурсов в сети;

- netbios-node-type задает рабочий режим NetBIOS-клиента в сети;
- default-router определяет IP-адреса маршрутизатора по умолчанию, которому клиенты могут переадресовывать пакеты с неизвестным пунктом назначения;
- lease задает срок, в течение которого динамически назначенные (lease — арендованные) адреса действительны без возобновления.

Каждая из субкоманд dns-server, netbios-name-sarver и default-router воспринимает в качестве параметров от одного до восьми IP-адресов, с которыми клиент может контактировать по каждой из этих функций. Параметром субкоманды domain-name является произвольная цепочка символов, представляющая имя DNS-домена для клиента. Субкоманда lease воспринимает в качестве параметров до трех целых чисел, которые задают количество дней, часов и минут, в течение которых назначенный адрес действителен. Может также быть использовано и ключевое слово infinit, свидетельствующее, что аренда адреса действительна неограниченный период времени. Субкоманда netbios-node-type имеет параметром символьные значения b, p, m или h, которые обозначают, соответственно, широковещательный NetBIOS-узел, равноправный узел, смешанный узел и гибридный узел, чем определяется рабочий режим клиента. Если вы не знакомы с этими рабочими режимами, выбирайте гибридный режим.

Ниже показано, как следует конфигурировать маршрутизатор в Куала-Лумпуре с применением DHCP-субкоманд конфигурирования, чтобы обеспечить предоставление DHCP-клиентам информации о серверах в сети компании ZIP:

```
Kuala-Lumpur#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Kuala-Lumpur(config)#ip dhcp pool kl-users
Kuala-Lumpur(config-dhcp)#dns-server 131.108.101.34 131.108.101.35
Kuala-Lumpur(config-dhcp)#domain-name zipnet.com
Kuala-Lumpur(config-dhcp)#netbios-name-server 131.108.21.70
Kuala-Lumpur(config-dhcp)#netbios-node-type h
Kuala-Lumpur(config-dhcp)#default-router 131.108.2.1
Kuala-Lumpur(config-dhcp)#lease 0 1
Kuala-Lumpur(config-dhcp)#^Z
```

Как упоминалось ранее, на одном DHCP-сервере ОС IOS может быть сконфигурировано несколько DHCP-пулов адресов. Группу DHCP-пулов адресов на таком сервере называют DHCP-базой данных. DHCP-база данных организована по иерархической или древовидной структуре, так что один адресный пул может быть подсетью сетевого адреса другого DHCP-пула адресов. Иерархическая структура позволяет пулу адресов, являющемуся подсетью другого пула, наследовать свойства. Свойства, которые являются общими для нескольких пулов, должны задаваться на самом высоком сетевом или подсетевом уровне конфигурируемого DHCP-сервера или сети. Вместе с тем, свойства, заданные на более высоком уровне, могут быть отменены на более низком подсетевом уровне. Рассмотрим этот аспект на примере маршрутизатора компании ZIP в Куала-Лумпуре.

В предыдущем примере был задан адресный пул kl-users для сети с адресом 131.108.2.0/25. Отдельно были заданы такие дополнительные свойства, как DNS-серверы, маршрутизатор по умолчанию и так далее. Для создания второго пула адресов с сетевым адресом 131.108.2.128/25 в будущем пришлось бы снова задавать для него эти свойства, так как он не является подсетью ранее заданного пула. Таким образом, конфигурация маршрутизатора в Куала-Лумпуре приобрела бы следующий вид:

```
Kuala-Lumpur#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Kuala-Lumpur(config)#ip dhcp excluded-address 131.108.2.129 131.108.2.135
Kuala-Lumpur(config)#ip dhcp pool kl-users-2
Kuala-Lumpur(config-dhcp)#network 131.108.2.128/25
Kuala-Lumpur(config-dhcp)#dns-server 131.108.101.34 131.108.101.35
```

```
Kuala-Lumpur(config-dhcp)#domain-naine zipnet.com
Kuala-Lumpur(config-dhcp)#netbios-name-server 131.108.21.70
Kuala-Lumpur(config-dhcp)#netbios-node-type h
Kuala-Lumpur(config-dhcp)#default-router 131.108.2.129
Kuala-Lumpur(config-dhcp)#lease 0 1
Kuala-Lumpur(config-dhcp)#^Z
```

При конфигурировании пула адресов kl-users-2 используется ряд команд, параметры которых совпадают с параметрами для пула kl-users, что делает назначаемые такими субкомандами свойства идентичными.

Чтобы исключить повторение субкоманд при конфигурировании пулов kl-users и kl-users-2, эти пулы адресов можно переконфигурировать так, чтобы они стали подсетями другого пула сетевых адресов. Для этого понадобятся только те субкоманды, которые задают свойства, являющиеся уникальными для каждого из этих пулов. Для маршрутизатора в Куала-Лумпуре будет задаваться пул адресов для сетевого адреса 131.108.2.0/24, при этом будут задаваться только те свойства, которые будут наследоваться подсетевыми адресными пулами. Таким образом, задание пулов адресов kl-users и kl-users-2 сведется всего к двум субкомандам вместо семи.

Ниже показан пример переписанных DHCP-пулов адресов на маршрутизаторе в Куала-Лумпуре, где адресные пулы kl-users и kl-users-2 будут наследовать свойства пула адресов, названного kl-common:

```
Kuala-Lumpur#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Kuala-Lumpur(config)#ip dhcp pool lei-common
Kuala-Lumpur(config-dhcp)#network 131.108.2.0/24
Kuala-Lumpur(config-dhcp)#dns-server 131.108.101.34 131.108.101.35
Kuala-Lumpur(config-dhcp)#domain-name zipnet.com
Kuala-Lumpur(config-dhcp)#netbios-name-server 131.108.21.70
Kuala-Lumpur(config-dhcp)#netbios-node-type h
Kuala-Lumpur(config-dhcp)#lease 0 1
Kuala-Lumpur(config-dhcp)#ip dhcp pool kl-users
Kuala-Lumpur(config-dhcp)#network 131.108.2.0/25
Kuala-Lumpur(config-dhcp)#default-router 131.108.2.1
Kuala-Lumpur(config-dhcp)#ip dhcp pool kl-users-2
Kuala-Lumpur(config-dhcp)#network 131.108.2.128/25
Kuala-Lumpur(config-dhcp)#default-router 131.108.2.129
Kuala-Lumpur(config-dhcp)#^Z
```

В этом примере благодаря тому, что сети 131.108.2.0/25 и 131.108.2.128/25 являются подсетями сети 131.108.2.0/24, соответствующие пулы адресов будут наследовать общие свойства пула более высокого уровня иерархии. И только субкоманда default-router используется для задания своего IP-адреса, соответствующего каждому подсетевому адресному пулу.

После того как пулы адресов и их свойства заданы, и DHCP-сервер ОС IOS начал присваивать IP-адреса, работу DHCP-сервера можно верифицировать с помощью нескольких команд режима EXEC ОС IOS. Верификация того, что DHCP-сервер ОС IOS производит протоколирование информации о привязках и конфликтах на занесенной в конфигурацию рабочей станции или на сервере, выполняется с помощью команды режима EXEC ОС IOS show ip dhcp database. Для вывода на экран информации о конкретной базе данных протоколирования этой команде требуется параметр URL. Если параметр не указывается, то выводится информация обо всех существующих узлах протоколирования. Ниже показан пример исполнения команды show ip dhcp database на маршрутизаторе компании ZIP в Куала-Лумпуре:

```
Kuala-Lumpur>show ip dhcp database
URL          :     tftp://131.108.2.77/kl-dhcp-info
Read         :     Never
Written      :     Jun 30 2000 12:01 AM
Status       :     Last Write Successful.
Delay        :     300 seconds
```

```
Timeout      :      300 seconds
Failures     :      0
Successes    :      72
Kuala-Lumpur>
```

В выводимой командой show ip dhcp database информации указывается местонахождение узла с данными о привязке, дате и времени последнего чтения или записи в базе данных привязок, статусе последнего чтения или записи и количестве успешных завершений и неудач при попытке сделать запись в базе данных привязок. Информацию о конкретном назначении адреса можно просмотреть, используя команду режима EXEC ОС IOS show ip dhcp binding. Если для этой команды указать в качестве необязательного параметра IP-адрес, то будет показана информация о привязке для этого конкретного адреса. В противном случае выводится вся информация о привязках. Ниже приводится пример исполнения команды show ip dhcp binding на маршрутизаторе компании ZIP в Куала-Лумпуре, где в выводимой информации показываются текущие данные о выделенных и назначенных адресах, соответствующий MAC-адрес DHCP-клиента и время истечения срока аренды адреса:

```
Kuala-Lumpur>show ip dhcp binding
IP  address        Hardware  address  Lease  expiration         Type
131.108.2.89       OOaO.9802.32de     Jul 01 2000 12:00 AM     Automatic
131.108.2.156      OOaO.9478.43ae     Jul 01 2000 1:00 AM      Automatic
Kuala-Lumpur>
```

Информацию о конфликтах адресов, которые имели место, когда DHCP-сервер ОС IOS пытался назначить адрес DHCP-клиенту, можно просмотреть с помощью команды show ip dhcp conflict. Если этой команде в качестве необязательного параметра указывается IP-адрес, информация о конфликтах показывается только по этому адресу (если таковая имеется); в противном случае выводится вся информация о конфликтах. Ниже приведен пример исполнения команды show ip dhcp conflict на маршрутизаторе компании ZIP в Куала-Лумпуре, где указывается конфликтный IP-адрес, время обнаружения конфликта и метод, с помощью которого конфликт был обнаружен.

```
Kuala-Lumpur>show ip dhcp conflict
IP address     Detection Method       Detection time
131.108.2.126  PingJul 02 2000        12:28 AM
131.108.2.254  Gratuitous ARP         Jul 02 2000 01:12 AM
Kuala-Lumpur>
```

В столбце Detection Method (Метод обнаружения) указывается метод, использованный DHCP-сервером ОС IOS для определения конфликтности адреса. Если указан метод обнаружения ping, то это говорит о том, что перед назначением адреса DHCP-сервер сделал попытку пропинговать этот адрес и получил успешный ответ. Метод обнаружения Gratuitious ARP (метод беспричинного разрешения адреса) означает, что до выделения адреса DHCP-сервер ОС IOS обнаружил в своей ARP-таблице текущую и достоверную ARP-запись для этого адреса. Наличие ссылки на любой из этих методов говорит о том, что адрес, вероятно, используется (может быть, это результат несанкционированного использования, а возможно, кто-то забыл внести адрес в список исключаемых адресов).

Верификация того, что DHCP-сервер ОС IOS принимает и отвечает на DHCP-запросы, может быть выполнена с помощью команды ОС IOS режима EXEC show ip dhcp server statistics. Эта команда поставляет такую полезную информацию, как количество сконфигурированных пулов адресов, объем памяти, занимаемый базой данных DHCP-привязок, а также данные счета количества DHCP-сообщений разных типов, которые были получены или отосланы. Ниже приведен пример результата исполнения команды show ip dhcp server statistics на маршрутизаторе компании ZIP в Куала-Лумпуре:

```
Kuala-Lumpur>show ip dhcp server statistics
Memory usage         40392
Address pools        3
Database agents      1
Automatic bindings   48
Manual bindings      0
Expired bindings     7
Malformed messages   0
Message              Received
BOOTREQUEST          22
DHCPDISCOVER         175
DHCPREQUEST          168
DHCPDECLINE          0
DHCPRELEASE          0
DHCPINFORM           0
Message              Sent
BOOTREPLY            17
DHCPOFFER            166
DHCPACK              155
DHCPNAK              3
Kuala-Lumpur>
```

Резервное дублирование в IP-сетях с помощью протокола маршрутизатора горячего резерва

Многих сетевых администраторов беспокоит наличие в сети точек критических отказов. Они всеми силами пытаются обеспечить в ключевых узлах наличие как резервных путей, так и резервного оборудования, стараясь не допустить, чтобы одно устройство послужило причиной недоступности жизненно важных ресурсов сети. Маршрутизаторы (и некоторые серверы) весьма неплохо справляются с многообразием IP-путей, обмениваясь информацией динамической маршрутизации о различных путях по сети, выбирая наилучший на данный момент времени путь или пути и выполняя перемаршрутизацию, если путь изменился в результате отказа оборудования или канала.

Однако многие реализации рабочих станций, серверов и принтеров не способны обмениваться информацией динамической маршрутизации. Такие устройства обычно конфигурируются на один IP-адрес шлюза по умолчанию, который и служит им той трубой, которая связывает их с остальной сетью. Если маршрутизатор, выполняющий роль шлюза по умолчанию, отказывает, то устройство оказывается в ситуации, когда оно может общаться только в пределах локального сегмента IP-сети и эффективно отрезается от всей остальной сети. Даже при наличии резервного маршрутизатора, который мог бы служить в качестве шлюза по умолчанию, все равно нет динамического метода, который бы могли использовать рабочие станции для переключения на новый IP-адрес шлюза по умолчанию, а переконфигурирование вручную часто выходит за технические возможности пользователя.

Чтобы помочь сетевым администраторам справляться с подобной неприятной ситуацией, компанией Cisco Systems был разработан протокол маршрутизатора горячего резерва (Hot Standby Router Protocol — HSRP). Протокол HSRP был спроектирован для сегментов локальных сетей, в которых присутствует несколько маршрутизаторов и стоят устройства, использующие только статически задаваемый IP-адрес шлюза по умолчанию.

Концепция протокола HSRP довольно проста. Администратор создает виртуальный адрес шлюза по умолчанию и назначает его дублирующим маршрутизаторам, которые участвуют в обмене по протоколу HSRP в данном сегменте локальной сети. IP-устройства конфигурируются на использование виртуального адреса шлюза в качестве шлюза по умолчанию. Маршрутизаторы управляют этим виртуальным адресом шлюза, общаясь между собой и определяя ответственного за отправку трафика на виртуальный IP-адрес. Через регулярные интервалы времени они обмениваются информацией и выясняют, кто из них еще присутствует и способен пересылать трафик. Если первичный или лидирующий маршрутизатор в группе HSRP-маршрутизаторов выходит из строя, резервный маршрутизатор из этой группы начинает пересылать трафик HSRP-группы Поскольку маршрутизаторы сами решают, кто должен обрабатывать трафик, поступающий на виртуальный адрес, а стоящие в сегменте рабочие станции знают только виртуальный IP-

адрес своего шлюза по умолчанию, отказ первичного маршрутизатора едва ли обнаружим пользователями рабочих станций и не требует вмешательства со стороны пользователя или администратора сети.

Протокол HSRP весьма гибок. Администратор сети имеет возможность управлять всем поведением маршрутизаторов, входящих в HSRP-группу, включая такие аспекты, как назначение первичного и резервного маршрутизатора (или маршрутизаторов), задание поведения резервного маршрутизатора (должен ли он обрабатывать трафик после того, как первичный маршрутизатор снова становится доступным) и возможность перевода трафика на резервный маршрутизатор через другой интерфейс ведущего маршрутизатора.

Рассмотрим конфигурирование протокола HSRP на маршрутизаторах компании ZIP, находящихся в Сеуле. В Сеуле два маршрутизатора (Seoul-1 и Seoul-2) подключены к одной и той же логической IP-сети 131.108.3.0. Наличие двух или нескольких маршрутизаторов, которые могут выполнять роль шлюзов по умолчанию, является первой частью критериев для конфигурирования протокола HSRP. Другой частью критериев является присутствие в сети IP-устройств, которые могут поддерживать только один IP-адрес шлюза по умолчанию. В данном случае принтеры, серверы и ПК-рабочие станции удовлетворяют этому критерию.

Для базового конфигурирования протокола HSRP требуется только одна субкоманда конфигурирования интерфейсов ОС IOS standby ip. Эта команда имеет параметром IP-адрес, который будет использоваться в качестве виртуального IP-адреса шлюза по умолчанию. Команда применяется в отношении всех маршрутизаторов в одной логической IP-сети, которые будут членами одной HSRP-группы. Ниже приведен пример конфигурирования протокола HSRP на маршрутизаторах компании ZIP Seoul-1 и Seoul-2 путем задания командой standby ip виртуального IP-адреса 131.108.3.3:

```
Seoul-l#configrure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul-1(config)#interface ethernet 0
Seoul-1(config-if)#standby ip 131.108.3.3
Seoul-1(config-if)#^Z
Seoul-2#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul-2(config)#interface ethernet 0
Seoul-2(config-if)#standby ip 131.108.3.3
Seoul-2(config-if)#^Z
```

После конфигурирования резервного HSRP-адреса определяется, какой маршрутизатор будет первичным переадресующим, а какой — резервным. Оба маршрутизатора вводят в свои ARP-таблицы IP-адрес и MAC-адрес для виртуального IP-адреса. Первичный переадресующий маршрутизатор начинает переадресацию трафика, посылаемого на виртуальный IP-адрес, а также отвечает на пинги и принимает сеансы виртуального терминала по этому адресу. Заметим, что MAC-адрес для виртуального IP-адреса на интерфейсах Ethernet, Fast Ethernet, Gigabit Ethernet и FDDI имеет форму 0000. Oc07 .acXX, где **xx** представляет собой идентификатор HSRP-группы. MAC-адрес для виртуального IP-адреса на интерфейсе Token Ring представляет собой функциональный адрес вида 1000.xxxx.xxxx. Ниже приведен пример исполнения команды show ip arp 131.108.3.3 на маршрутизаторе компании ZIP Seoul-1 со сконфигурированным протоколом HSRP:

```
Seoul-l#show ip arp 131.108.3.3
Protocol    Address          Age (min)   Hardware Addr     Type        Interface
Internet    131.108.3.3      -           0000.Oc07.acOO    ARPA        EthernetO
Seoul-l#
```

Совет

Некоторые устройства в сетях Token Ring не воспринимают MAC-адрес IP-устройства в качестве группового функционального адреса. В этом случае следует использовать субкоманду конфигурирования интерфейса ОС IOS standby use-bia, чтобы заставить виртуальный IP-адрес протокола HSRP использовать адрес, занесенный в ППЗУ интерфейса, что ограничивает количество HSRP-групп на интерфейсе до одной.

Как упоминалось ранее, администратор сети имеет несколько опций конфигурирования, которые управляют поведением в рамках протокола HSRP. Для выбора первичного переадресующего маршрутизатора используется субкоманда конфигурирования интерфейса ОС IOS standby priority. Эта команда имеет параметром значение приоритета, лежащее в пределах от 0 до 255. Маршрутизатор из HSRP-группы с наивысшим приоритетом становится переадресующим маршрутизатором. В нашем примере маршрутизатор компании ZIP Seoul-1 конфигурируется значением HSRP-приоритета 100, а маршрутизатору Seoul-2 присваивается приоритет 95, в результате чего активным переадресующим маршрутизатором становится маршрутизатор Seoul-1:

```
Seoul-1#configure
Configuring from terminal, memory, or network [terminal]? Enter
configuration commands, one per line. End with CNTL/Z.
Seoul-1(config)#interface ethernet 0
Seoul-1(config-if)#standby priority 100
Seoul-1(config-if)#^2
Seoul-2#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul-2(config)#interface ethernet 0
Seoul-2(config-if)#standby priority 95
Seoul-2(config-if)#^Z
```

Если требуется, чтобы резервный маршрутизатор стал активным, то он автоматически перебирает на себя эту роль. Можно управлять и тем, становится ли бывший ранее первичным маршрутизатор активным переадресующим после того, как вновь стал доступным. Для того чтобы маршрутизатор становился активным переадресующим вместо маршрутизатора с меньшим значением приоритета, используется субкоманда конфигурирования интерфейсов ОС IOS standby preempt. В нашем примере маршрутизатор Seoul-2 имеет меньший приоритет, чем маршрутизатор Seoul-1. Если маршрутизатор Seoul-1 отказывает, то роль активного переадресующего берет на себя маршрутизатор Seoul-2. Без команды standby preempt маршрутизатору Seoul-1 маршрутизатор Seoul-2 сохранит роль активного переадресующего. В примере ниже команда standby preempt приведет к тому, что маршрутизатор компании ZIP Seoul-1 станет активным переадресующим после восстановления, так как у него более высокое значение HSRP-приоритета:

```
Seoul-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul-1 (config)#interface ethemet 0
Seoul-1(config-if)#standby preempt
Seoul-1(config-if)#^Z
```

В некоторых ситуациях рабочий статус интерфейса непосредственно влияет на то, какой маршрутизатор должен играть роль активного переадресующего. Это происходит, когда каждый из маршрутизаторов в HSRP-группе имеет свой отличающийся путь к другим частям сети. В сети компании ZIP маршрутизатор Seoul-1 имеет соединение с маршрутизатором в Сан-Хосе, который, в свою очередь, связан с Сан-Франциско. Маршрутизатор Seoul-2 имеет прямую связь с Сан-Франциско и Сан-Хосе. Если в маршрутизаторе Seoul-1 соединение через интерфейс глобальной сети деградирует или выходит из строя, то пакеты, посылаемые на маршрутизатор Seoul-1, активный переадресовывающий маршрутизатор, не могут достичь Сан-Франциско или Сан-Хосе. В конце

концов, работа протоколов динамической маршрутизации приведет к тому, что маршрутизатор Seoul-1 начнет слать пакеты маршрутизатору Seoul-2 для пересылки по его работоспособному интерфейсу глобальной сети. Но такая реконвергенция может занять несколько минут и нарушить нормальный поток сетевого трафика. Однако, если бы маршрутизатор Seoul-2 смог перебрать на себя роль активного переадресующего, то он смог бы немедленно переадресовывать пакеты в Сан-Франциско и Сан-Хосе через свое функционирующее соединение по интерфейсу глобальной сети.

ОС IOS имеет HSRP-функцию, которая может заставить маршрутизатор Seoul-1 отрегулировать свой HSRP-приоритет в HSRP-группе на интерфейсе Ethernet 0 так, что активным переадресующим маршрутизатором станет маршрутизатор Seoul-2. Эта функция — отслеживание интерфейса — активизируется субкомандой конфигурирования интерфейса ОС IOS standby track. В качестве параметра этой команды выступает название интерфейса, подлежащего отслеживанию, и, как вариант, величина, на которую будет уменьшаться значение HSRP-приоритета конфигурируемого интерфейса. Если значение уменьшения не задается, то маршрутизатор будет вычитать стандартную величину — десять.

Ниже приведен пример конфигурирования команды standby track на маршрутизаторе компании ZIP Seoul-1 таким образом, чтобы в случае выхода из строя интерфейса глобальной сети Serial 1 активным переадресующим маршрутизатором становился маршрутизатор Seoul-2:

```
Seoul-l#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Seoul-1 (config) #interface ethemet 0
Seoul-1(config-if)#standby track serial 1
Seoul-1(config-if) #^Z
```

Верифицировать работу протокола HSRP можно с помощью команды режима EXEC ОС IOS show standby. Эта команда имеет необязательный параметр, в качестве которого указывается название конкретного интерфейса, для которого требуется вывести информацию о деятельности протокола HSRP. Если название интерфейса не указывается, то информация о работе протокола HSRP выводится для всех интерфейсов. Ниже показан пример информации, выводимой командой show standby на маршрутизаторах Seoul-1 и Seoul-2:

```
Seoul-l#show standby
EthernetO - Group 0
  Local state is Active, priority 100, may preempt
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:01.880
  Hot standby IP address is 131.108.3.3 configured
  Active router is local
  Standby router is 131.108.3.2 expires in 00:00:07
  Tracking interface states for 1 interface, 1 up:
   Up SerialO Seoul-l#
Seoul-2#show standby
EthernetO - Group 0
  Local state is Standby,priority 95, may preempt
  Hellotime 3 holdtime 10
  Next hello sent in 00:00:01.380
  Hot standby IP address is 131.108.3.3 configured
  Active router is 131.108.3.1 expires in 00:00:06
  Standby router is local
Seoul-2#
```

Команда show standby выводит разнообразную информацию протокола HSRP, включая состояние переадресации, HSRP-приоритет и названия отслеживаемых интерфейсов на запрашиваемом маршрутизаторе. Также выводится сконфигурированный виртуальный IP-адрес и IP-адрес (адреса) возможных резервных маршрутизаторов внутри каждой HSRP-группы.

Одним из недостатков первоначального варианта протокола HSRP было то, что он не позволял администратору сети разделять трафик нагрузки между обоими маршрутизаторами группы с

резервированием. По сути, резервный маршрутизатор просто находился на холостом ходу, если активный переадресующий маршрутизатор не выходил из строя.

Чтобы решить эту проблему, в ОС IOS была добавлена возможность поддержки на одном интерфейсе нескольких HSRP-групп. На одном интерфейсе можно создавать несколько HSRP-групп, причем каждую со своим отличающимся виртуальным IP-адресом, чтобы резервировать друг друга. Например, если взять маршрутизаторы компании ZIP Seoul-1 и Seoul-2, первая HSRP-группа может иметь виртуальный IP-адрес 131.108.3.3 с маршрутизатором Seoul-1, назначенным в качестве первичного переадресуюшего из-за его более высокого HSRP-приоритета. Можно сконфигурировать вторую HSRP-группу с виртуальным IP-адресом 131.108.3.4, но с маршрутизатором Seoul-2, выделенным в качестве первичного переадресующего маршрутизатора с помощью присвоения ему во второй HSRP-группе более высокого HSRP-приоритета, чем у маршрутизатора Seoul-1. Тогда в первой HSRP- группе Seoul-1 будет активным переадресующим маршрутизатором, а Seoul-2 — резервным, и одновременно во второй HSRP- группе Seoul-2 будет активным переадресующим маршрутизатором, а Seoul-1 — резервным.

Имея заданными две HSRP- группе с двумя виртуальными IP-адресами, администратор сети может сконфигурировать шлюз по умолчанию на части хост-машин с одним виртуальным адресом, а на остальных хост-машинах — с другим. Хотя этим и не достигается точная балансировка, такая конфигурация позволяет разделить нагрузку между двумя маршрутизаторами, а не иметь существенно перегруженным один и полностью неиспользуемым второй.

Несколько HSRP-групп создаются опционным заданием номера группы во всех командах standby. Например, команды standby I ip address 131.108.3.3 И standby 1 priority 100 показывают, что эти HSRP-команды используются для групп с резервированием ПОД номером 1. Команды standby 2 ap address 131.108.3.4 и standby 2 priority 100 показывают, что эти HSRP-команды применяются в отношении группы с резервированием под номером 2.

Резюме

В данной главе на примере сети компании ZIP были рассмотрены базовые конфигурации некоторых наиболее общих элементов сетевого протокола TCP/IP. Что касается всех технических особенностей ОС IOS, существуют сотни дополнительных субфункций и кнопок, которые администратор сети может конфигурировать, чтобы улучшить работу сети и маршрутизатора. Изучая различные документальные источники и экспериментируя в лаборатории, администратор сети может начать более четко понимать и оценивать мощь ОС IOS при создании устойчивой к сбоям высокопроизводительной и многофункциональной сетевой среды. В данной главе ключевыми являются следующие концепции.

- IP-адреса являются десятичными представлениями 32-разрядных двоичных чисел. Они группируются в блоки сетевых адресов и делятся на категории в соответствии с конкретными классами сетей. Сетевые администраторы могут распределять адресное пространство сети между несколькими сегментами локальной или глобальной сети, используя для этого технику подсетей.
- Конфигурирование IP-адресов заключается в присвоении IP-адресов интерфейсам маршрутизатора. IP-адреса назначаются либо из адресного пространства сетей общего пользования, либо из адресного пространства частных сетей. Адреса общего пользования предоставляются провайдерами Internet или региональными реестрами адресов. Конфигурирование IP-адресов на интерфейсах глобальных сетей требует дополнительных команд для отображения адресов канального уровня на IP-адреса вручную.
- Конфигурирование IP-маршрутизации позволяет маршрутизатору реализовывать функцию коммутации IP-пакетов. Для построения таблицы сетевых адресов пунктов назначения, называемой таблицей маршрутизации, могут использоваться статические маршруты. Сводные маршруты и маршруты по умолчанию обеспечивают информацию о достижимости, одновременно минимизируя объем той информации, которую необходимо хранить в таблице маршрутизации. Бесклассовая маршрутизация позволяет маршрутизаторам посылать пакеты, имеющие пунктом назначения сетевые адреса, которые не попадают в

пределы границ традиционных отвечающих классам сетей.

- Протоколы динамической IP-маршрутизации позволяют маршрутизаторам обмениваться информацией о достижимости для тех сетей, которые локально подключены к ним. Протоколы динамической маршрутизации подразделяются на две основные категории: протоколы внутренних шлюзов и протоколы внешних шлюзов. Существует два основных типа протоколов внутренних шлюзов: протоколы на основе метода вектора расстояния и протоколы на основе метода учета состояния каналов связи. ОС IOS обеспечивает инструментальные средства для управления распространением информации сетевой маршрутизации и взаимодействием маршрутизаторов, обменивающихся информацией динамической маршрутизации.
- Списки IP-доступа позволяют осуществлять фильтрацию потока пакетов в IP-сети для обеспечения зашиты и конфиденциальности информации. Активация списков доступа производится в два этапа: задание критериев фильтрации и наложение их Списки доступа служат в качестве инструмента для активации других типов фильтрации, например, в процессе обмена информацией динамической маршрутизации.
- Основные IP-службы удаленного доступа по коммутируемым каналам позволяют удаленным пользователям иметь доступ в сеть через модемы и ISDN-каналы, если те подключены к среде локальной сети.
- Верификация IP-взаимодействия может осуществляться с помощью таких команд, как show ip route и ping. Диагностические возможности команд trace и debug позволяют сетевому администратору обнаруживать ошибки конфигурирования и идентифицировать проблемы, возникающие как в маршрутизаторах, так и в сети.
- Такие IP-функции, как служба имен доменов, облегчают решение задачи поддержки, возложенной на плечи администратора сети. Функция переадресации широко вещательных пакетов позволяет службам с широковещательной рассылкой, например DHCP, работать в маршрутизируемых сетях. Входящий в состав ОС IOS DHCP-сервер обеспечивает для сетей малого и среднего размера реализацию функции динамического назначения адресов самим маршрутизатором или сервером доступа На рабочих станциях, которые не поддерживают протоколы динамической маршрутизации, протокол маршрутизатора горячего резерва Hot Standby Router Protocol обеспечивает устойчивость к отказам и резервное дублирование.
  - В табл. 4 8 приводятся основные команды режима EXEC для протокола IP

Таблица 4.8. Сводная таблица команд режима EXEC для протокола IP

Команда	**Описание**
clear host	Удаляет временные записи из таблицы IP-хост-машин
Clear ip access list counters	строки списка IP-доступа Обнуляет счетчики удовлетворения критериев каждой
clear *ip* route	Очищает всю таблицу маршрутизации или удаляет конкретный маршрут
ping *ip-адрес*	Проверяет указанный IP-адрес на предмет его достижимости и способности отвечать
show {frame-relay | atm | x25 | dialer) map	Показывает отображения IP-адресов на адреса канального уровня для заданного типа среды глобальной сети
show access-list	Показывает все списки доступа, заданные на маршрутизаторе
show host	Верифицирует конфигурацию службы DNS на маршрутизаторе и выводит на экран список хост-машин, именам которых были поставлены в соответствие IP-адреса
show interface *интерфейс*	Обеспечивает общую информацию об интерфейсе, включая IP-адрес и сетевую маску
show ip access-list	Показывает все списки IP-доступа, заданные на маршрутизаторе
show ip arp	Выводит на экран все IP-адреса, которые маршрутизатор смог преобразовать в MAC-адреса
show ip dhcp binding	Выводит на экран информацию обо всех назначениях адресов, сделанных DHCP-сервером ОС IOS

show ip dhcp conflict	Выводит на экран информацию о конфликтах IP-адресов, обнаруженных DHCP-сервером ОС IOS во время процесса выделения адресов
show ip dhcp database	Выводит на экран информацию о местонахождении и статусе базы данных, используемой DHCP-сервером ОС IOS для протоколирования DHCP-привязок и конфликтов
show ip dhcp server statistics	Выводит на экран информацию о статусе и результаты счета, связанные с работой DHCP-сервера ОС IOS
show ip interface brief	Показывает краткое резюме информации об IP-адресах и статусах всех интерфейсов, имеющихся на устройстве
show ip interface *интерфейс*	Показывает все параметры, связанные с IP-конфигурацией интерфейса
show ip masks *сетевой адрес*	Дает список сетевых масок, которые были применены в отношении названной сети и количество маршрутов, использовавших каждую маску
show ip protocols	Показывает исполняемые протоколы маршрутизации и различные их атрибуты При использовании с ключевым словом summary показывает только названия протоколов и числа, соответствующие идентификаторам процессов
show ip route	Выводит данные таблицы IP-маршрутизации маршрутизатора
show ip route connected	Показывает маршруты, связанные с находящимися в рабочем состоянии и непосредственно подключенными интерфейсами маршрутизатора
show ip route *ip-адрес*	Выдает информацию о маршрутизации для заданного маршрута
show ip route static	Показывает данные о маршрутах, полученных на основании команд конфигурирования сетевых маршрутов вручную
show ip traffic	Показывает обобщенные статистические данные работы маршрутизатора в рамках протокола IP
show standby	Выводит на экран информацию о работе протокола HSRP
terminal ip netmask-format {decimal I bit-count | hexidecimal}	Задает формат отображения сетевых масок, который будет использоваться во время текущего сеанса виртуального терминала или сеанса консоли
trace ip-адрес	Выводит на экран каждый этап сетевого пути, который проходит пакет, добираясь до указанного IP-адреса

В табл. 4. 9 приведены команды конфигурирования протокола IP

Таблица 4.9. Сводная таблица команд конфигурирования протокола IP

Команда	Описание
aaa authentication ppp *метод из списка*	Определяет, что аутентификация в рамках протокола ppp должна выполняться с использованием приведенного метода
aaa authorization network *метод*	Задет что сетевые службы должны аутентифицироваться с использованием приведенного AAA-метода
access-list	Создает нумерованный список доступа и связанные с ним критерии фильтрации
arp-server	Идентифицирует ATM ARP-сервер, который может преобразовывать IP-адреса в ATM NSAP-адреса
async-bootp dns-server *ip адрес*	Задает на глобальной основе IP-адрес (адреса) DNS-сервера, который будет предоставляться удаленным клиентам при установлении связи по звонку
async-bootp nbns-server *ip адрес*	Задает на глобальной основе IP-адрес (адреса) NetBIOS/WINS-сервера имен, который будет предоставляться удаленным клиентам при установлении связи по звонку
async mode {interactive | dedicated}	Устанавливает метод взаимодействия с пользователями на асинхронном интерфейсе для удаленных пользователей, работающих с коммутируемыми каналами связи

autoselect during-login	Задает выполнение процесса автовыбора во время процесса аутентификации
autoselect ppp	Задает автовыбор протокола PPP для асинхронной линии, сконфигурированной на работу в интерактивном режиме
compress	Определяет попытку переговоров относительно алгоритма сжатия данных во время установления удаленного соединения по коммутируемой линии при использовании протокола PPP
default-metric	Определяет значения метрики маршрутизации по умолчанию, которые должны использоваться при редистрибуции маршрутов между протоколами динамической маршрутизации
default-router *адрес*	Задает один или несколько IP-адресов маршрутизаторов по умолчанию, которые поставляются DHCP-клиентам DHCP-сервером ОС IOS
dialer-group *целочисленное значение*	Задает номер группы интерфейсов вызова по номеру, к которой принадлежит интерфейс, и номер списка сетевых протоколов, трафик в рамках которых следует считать интересным
dialer-list *номер списка* protocol *типовой метод*	Задает список, который определяет сетевые протоколы и методы, используемые для определения того, что трафик представляет интерес для сеансов удаленного доступа по коммутируемой линии
dialer map ip	Отображает IP-адрес в имя системы и номер телефона для ISDN-соединений
dialer rotary-group *целочисленное значение*	Приписывает интерфейс ISDN к групповой структуре интерфейсов вызова по номеру
distribute-list	Накладывает список доступа на задачу приема и объявления сетевых маршрутов
dns-server *адрес*	Задает один или несколько IP-адресов DNS-серверов, которые поставляются DHCP-клиентам DHCP-сервером OCIOS
domain-name *имя домена*	Задает DNS-имя домена, которое поставляется DHCP-клиентам DHCP-сервером ОС IOS
flowcontrol *{hardware | software)*	Задает метод управления потоком в асинхронной линии
frame-relay map ip	Отображает IP-адрес на DLCI-идентификаторы интерфейса Frame Relay
group-range *начало конец*	Задает асинхронные интерфейсы, включаемые в групповой асинхронный интерфейс
ip access-group list *(in I out}*	Накладывает указанный список доступа на задачу фильтрации входящих или исходящих пакетов интерфейса
ip access-list {extended | standard} *имя*	Создает именованный список IP-доступа и связанные с ним критерии фильтрации
ip address *ip-адрес сетевая маска*	Назначает IP-адрес и сетевую маску интерфейсам локальных и глобальных сетей
ip classless	Позволяет маршрутизатору работать в бесклассовом режиме, когда IP-адреса пунктов назначения отвечают суперсетевым маршрутам или маршрутам, определяемым

	CIDR-блоками
ip default-information originate	Заставляет протокол OSPF генерировать маршрут по умолчанию из пограничного маршрутизатора автономной системы в остальную часть OSPF-домена
ip default-network *сетевой адрес*	Конфигурирует заданный сетевой адрес в качестве сводной сети или сети по умолчанию
{no} ip dhcp conflxct logging	Активирует или деактивирует в DHCP-сервере ОС IDS функцию протоколирования информации о конфликтах адресов
ip dhcp database *url*	Задает местонахождение базы данных и метод протоколирования информации о привязках сделанных DHCP-сервером и конфликтах
ip dhcp excluded-address	Задает один или несколько IP-адресов которые должны быть исключены из предложений DHCP-клиентам от DHCP-сервера ОС IOS
ip dhcp pool *имя*	Создает DHCP-пул адресов, который может быть сконфигурирован с помощью дополнительных субкоманд DHCP-конфигурирования
ip dhcp-server *ip-adpec*	Задает IP-адрес DHCP-сервера который может динамически назначать IP-адреса удаленным клиентам работающим по коммутируемым каналам связи
ip domain-lxst *имя*	Устанавливает список имен доменов для присоединения к именам неопознанных хост-машин
ip domain-lookup	Активирует работу DNS-службы
ip domain-name *имя*	Конфигурирует имя первичного домена которое будет присоединяться к именам неопознанных хост-машин
ip forward-protocol udp type	Управляет типом широковещательных UDP-пакетов которые будут переадресовываться
ip helper-address *ip адрес*	Переадресует широковещательные UDP-пакеты по за данному IP-адресу
ip host	Конфигурирует статическое отображение имени хост-машины в IP-адрес (адреса)
ip local pool (default | pool-name} *начальный ip-адрес конечный ip-adpec*	Создает пул IP-адресов для динамического назначения IP-адресов удаленным клиентам работающим по коммутируемым каналам связи
ip name-server *ip-adpec*	Конфигурирует DNS-сервер (серверы) имен
ip netmask-format {decxmal bxt-count | hexxdecxmal}	Конфигурирует формат отображения сетевых масок | во время сеансов виртуального терминала или консольных сеансов
ip ospf network {broadcast | non-broadcast |poxnt-to-multipoint}	Конфигурирует тип сети с широковещанием без широковещания и из точки многим которая для протокола OSPF подключена к интерфейсу
ip rip {send I recexve) versxon	Задает номер версии протокола RIP для отправки и приема через конкретный интерфейс
ip route 0.0.0.0 0.0.0.0	Конфигурирует маршрут по умолчанию 0.0.0.0

ip-адрес пункта назначения

ip route *сетевой адрес сетевая маска ip-адрес пункта назначения*	Конфигурирует статический маршрут
ip route *сетевой адрес сетевая маска подсетевои ip-адрес*	Конфигурирует сводный маршрут используя в качестве параметра сводный маршрут сетевую маску и неподключенную подсеть
ip routing	Активирует в маршрутизаторе функцию IP- маршрутизации
ip subnet-zero	Позволяет назначить интерфейсу первую подсеть из диапазона сетевых адресов (нулевую подсеть)
ip unnumbered *интерфейс*	Конфигурирует ненумерованный двухточечный IP-интерфейс глобальной сети
map-group	Назначает интерфейсу именованную группу отображений для использования им в процессе отображения IP-адресов на ATM-адреса канального уровня
map-list	Создает именованный список отображений для конфигурирования отображения IP-адресов на постоянные или коммутируемые виртуальные каналы в процессе ATM-адресации
modem autooonf igure {discovery | *тип модема]*	Задает тип автоматического конфигурирования модема подключенного к асинхронной линии по распознаванию или с помощью установок для названного типа модема
modem {dailin | inout)	Задает разрешенное направление асинхронных вызовов по звонку
neighbor *ip-адрес*	Задает IP-адрес соседнего маршрутизатора для обмена информацией динамической маршрутизации
neighbor *ip-адрес description*	Позволяет вводить комментарии в BGP-команде neighbor
neighbor *ip-адрес distribute-list*	Позволяет осуществлять фильтрацию для каждого одноранговогo BGP-соседа отдельно
neighbor *ip-адрес* remote-as asn	Конфигурирует соседний маршрутизатор с указанным адресом в указанной автономной системе в качестве одноранговогo BGP-маршрутизатора
neighbor *ip-адрес* update-source *интерфейс*	Указывает что IP-адрес источника для установления сеанса BGP-обмена между одноранговыми маршрутизаторами следует брать из названного интерфейса
netbios-name-server *адрес*	Задает один или несколько IP-адресов NetBIOS/WINS-сервера для предоставления DHCP-клиентам DHCP-сервером ОС IOS
netbios-node-type *тип*	Задает режим поведения NetBIOS, сообщаемый DHCP-клиентам DHCP-сервером ОС IOS
network *сетевой адрес*	Указывает, что подключенные интерфейсы, адреса которых согласуются с заданным сетевым адресом, должны быть включены в маршрутные объявления
network *сетевой адрес* area *номер области*	Указывает, что подключенные интерфейсы, адреса которых согласуются с заданным сетевым адресом,

	должны включаться в маршрутные объявления по протоколу OSPF, и интерфейсы должны быть приписаны к заданной области
network *номер сети [маска | длины}*	Задает диапазон IP-адресов, которые будут *префикс* предлагаться DHCP-клиентам DHCP-сервером из заданного DHCP-пула адресов
no auto-summary	Запрещает автоматическое сведение адресов на границах класса сети и разрешает передачу информации о подсетях
no inverse-arp	Отключает функцию динамического отображения IP-адресов в DLCI-идентификаторы на интерфейсе Frame Relay
passive-interface *интерфейс*	Конфигурирует маршрутизатор таким образом, чтобы он слушал, но не объявлял маршрутную информацию на заданном интерфейсе
peer default ip address {pool | dhcp | *ip-adpec}*	Задает метод, используемый для назначения IP-адреса удаленной рабочей станции клиента, работающей по коммутируемым каналам связи
ppp autentication *метод*	Указывает на то, что PPP-аутентификация должна выполняться до начала работы сетевых служб Между сервером доступа и удаленным клиентом используется названный протокол аутентификации
ppp ipcp {dns I wins}	Задает поставку IP-адреса (адресов) DNS или NetBIOS/WINS-серверов удаленным клиентам во время установления PPP-сеанса на поинтерфейсной основе
ppp multilink	Свидетельствует о необходимости активации на интерфейсе программного мультиплексирования каналов
redistribute protocol	Активирует редистрибуцию маршрутов из указанного протокола
router {rip | igrp | ospf | eigrp | bgp)	Разрешает маршрутизатору исполнение заданного протокола динамической маршрутизации
speed *бит в секунду*	Задает скорость передачи по асинхронной линии
standby ip *ip-adpec*	Конфигурирует указанный IP-адрес в качестве виртуального IP-адреса для HSRP-группы
standby preempt	Заставляет маршрутизатор с более высоким значением HSRP-приоритета перебирать на себя роль активного переадресующего после того, как он вновь становится доступным
standby priority *приоритет*	Назначает значение приоритета HSRP-маршрутизатору для управления процессом выбора первичного переадресующего маршрутизатора
standby track *интерфейс*	Активирует динамическую регулировку HSRP-приоритета на основе рабочего статуса заданного интерфейса
stanby use-bia	Осуществляет принудительное восприятие аппаратно запрограммированного MAC-адреса интерфейса в качестве виртуального IP-адреса
{no} synchronization	Активирует или отменяет требование предварительного обучения маршрутизаторов через IGP-процесс маршрутизации до объявления маршрутов EBGP-соседям
username *имя пользователя*	Задает локальное значение пары имя пользовате-

`password` *слово*	ля/пароль для использования в процессе аутентификации удаленного пользователя
`version` *версия протокола RIP*	Задает номер версии протокола `RIP`, использующегося на маршрутизаторе, для которого разрешено исполнение этого протокола
`x25 map ip`	Отображает `IP`-адрес на адрес протокола `X 121`

Дополнительная литература

Поставленные в данной главе вопросы более подробно рассматриваются в следующих изданиях.

1. Bellovin, S.M., and W.R. Cheswick, *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, Massachusetts: Addison-Wesley, 1994.
2. Comer, D.E. *Internetworking with TCP/IP*. Volume I, Fourth Edition. Englewood Cliffs, New Jersey: Prentice Hall, 2000. (Готовится к изданию перевод на русский язык этой книги, который выйдет в свет в ИД "Вильяме" в начале 2002 года. — *Прим ред.)*
3. Сэм Хелеби, Денни Мак-Ферсон. *Принципы маршрутизации в Internet*, 2-е издание. ИД "Вильяме", 2001 г.
4. Halabi B., and D. McPherson. *Internet Routing Architectures*, Second Edition Indianapolis, Indiana: Cisco Press, 2000.
5. Zwicky, E.D., et al. *Building Internet Firewalls*, Second Edition. Sebastopol, California: O'Reilly & Associates, 2000.

Глава 5

Ключевые темы этой главы

- **Система адресации и структура адресов в протоколе AppleTalk** Основы структуры адресов протокола AppleTalk и структура сети
- **Конфигурирование адресов для протокола AppleTalk** Обзор схемы адресации протокола AppleTalk, а также примеры конфигурирования адресов для интерфейсов глобальных и локальных сетей различных типов
- **Конфигурирование маршрутизации по протоколу AppleTalk** Основы конфигурирования маршрутизации по протоколу AppleTalk с использованием статических маршрутов и верификация AppleTalk-маршрутизадии
- **Конфигурирование протоколов динамической маршрутизации, работающих с протоколом AppleTalk** Характеристики протоколов динамической маршрутизации RTMP и EIGRP протокола AppleTalk И примеры базовых конфигураций
- **Конфигурирование фильтрации в протоколе AppleTalk с применением списков доступа** Управление доступом в сети и защита информации с помощью команд access-list И appletalk access-group
- **Конфигурирование основных служб удаленного доступа по коммутируемым каналам связи протокола AppleTalk** Опции конфигурирования ОС IOS для обеспечения удаленного доступа пользователям, работающим по протоколу AppleTalk на коммутируемых каналах связи
- **Верификация взаимодействия в сети с протоколом AppleTalk и устранение неполадок** Идентификация проблем связи с помощью применения команд show, ping и debug

Основы AppleTalk

Протокол AppleTalk является одной из самых ранних реализаций вычислений в архитектуре клиент-сервер. Он был создан в середине 1980-х годов компанией Apple Computer для конечных пользователей семейства продуктов Macintosh, чтобы обеспечить возможность коллективного пользования ресурсами и, прежде всего, принтерами 1 и размещенными на серверах файлами.

Благодаря легкости в использовании протокол AppleTalk приобрел сторонников среди конечных пользователей, однако у инженеров-сетевиков и разработчиков он вызвал негативное отношение как немасштабируемый протокол, который трудно обслуживать в среде крупных корпораций. Усовершенствования сняли ряд критических замечаний со стороны специалистов в области сетевых систем, однако основными адвокатами протокола AppleTalk остаются конечные пользователи По иронии судьбы, некоторые функциональные особенности протокола AppleTalk, например, динамические переговоры о назначении адреса, которые и послужили причиной его критики со стороны проектировщиков за чрезмерную нагрузку на сеть, позже были реализованы в других широко применяемых протоколах, в частности, в протоколе IP в форме протокола динамического конфигурирования хост-машины Dynamic Host Configuration Protocol (DHCP)

На рис. 5.1 показаны различные протоколы, входящие в набор сетевых протоколов AppleTalk. В данной главе не будет рассматриваться весь комплект Основное внимание

будет уделено тем протоколам, которые используются на сетевом и транспортном уровнях, — протоколу разрешения адреса (AppleTalk Address Resolution Protocol — AARP), протоколу доставки дейтаграмм (Datagram Delivery Protocol — DDP), протоколу управления таблицами маршрутизации (Routing Table Maintenace Protocol — RTMP), протоколу привязки имен (Name Binding Protocol — NBP) и протоколу эхо-ответов (AppleTalk Echo Protocol — AEP). Дополнительно в разделе "Конфигурирование фильтрации в протоколе AppleTalk с применением списков доступа" будет рассмотрен протокол обмена зонной информацией (Zone Information Protocol — ZIP) Другие сетевые протоколы, показанные на рис. 5.1, с которыми читатель, вероятно, знаком, приведены просто для полноты картины.

Примечание

Не все версии ОС IOS компании Cisco поддерживают протокол AppleTalk Следует удостовериться, что версия ОС IOS, исполняемая на вашем маршрутизаторе, поддерживает набор протоколов для настольных систем *Desktop Protocol Suite*.

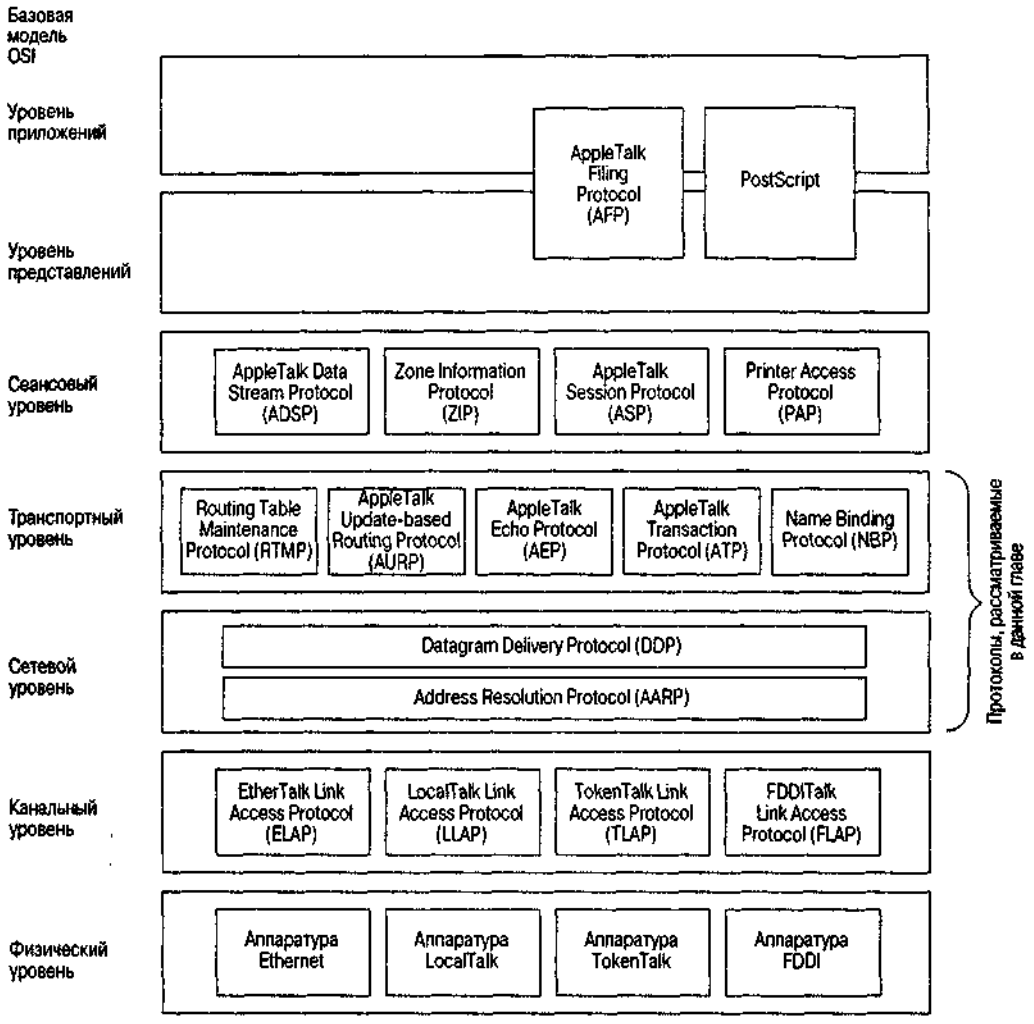


Рис. 5.1. Набор протоколов из состава протокола AppleTalk

Система адресации и структура адресов в протоколе AppleTalk

В отличие от протокола TCP/IP, который рассматривался в главе 4, "Основы TCP/IP", протокол AppleTalk является закрытым протоколом и контролируется компанией Apple Computers. Он имеет уникальную структуру сетевых адресов и единственную в своем роде методологию именования сетевых служб.

В данном разделе рассматривается структура сетевых адресов протокола AppleTalk, которой должны следовать все клиенты (также называемые рабочими станциями) и

серверы для того, чтобы иметь возможность обмениваться данными в рамках межсетевого взаимодействия по протоколу AppleTalk.

Сетевой AppleTalk-адрес представляет собой 24-разрядный адрес, состоящий из двух различных компонентов — 16-разрядной сетевой части и 8-разрядного адреса узла. Сетевая часть идентифицирует сегмент локальной или глобальной сети, а адрес узла — рабочую станцию или сервер. Эти компоненты обычно пишутся слитно в виде *сеть.узел с* использованием десятичной формы представления. Например, адрес 52.6 идентифицирует рабочую станцию или сервер 6 в сети 52. В отличие от протокола TCP/IP, который обладает многоуровневой иерархией адресов и предусматривает возможность суммирования, протокол AppleTalk ограничен этими двумя уровнями. Администрирование адресов в сети AppleTalk координирует протокол DDP, который к тому же обеспечивает доставку AppleTalk-пакетов, не устанавливая соединения.

Сетевые адреса для сегментов локальной или глобальной сети сетевой администратор задает точно так же, как для идентификации сегмента сети он задает TCP/IP-подсети. В протоколе AppleTalk существует два различных метода сетевой адресации сегментов локальной и глобальной сети: методы AppleTalk фазы 1 и 2. При методе AppleTalk фазы 1 сегменты сети идентифицируются одним номером сети.

При использовании метода AppleTalk фазы 2 сегменты сети идентифицируются величиной, называемой *кабельным диапазоном* (cable-range), который соответствует одному или нескольким логическим номерам сети. Кабельный диапазон может быть одним сетевым номером или непрерывной последовательностью сетевых номеров, задаваемой начальным и конечным сетевыми номерами в формате *начало-конец*. Например, кабельный диапазон 100-100 идентифицирует логическую сеть, имеющую один сетевой номер 100, тогда как кабельный диапазон 50-64 идентифицирует логическую сеть, охватывающую 15 сетевых номеров с 50 по 64.

Для общения с другими устройствами каждое устройство, стоящее в сети AppleTalk, должно иметь свой номер узла. В отличие от сетевых протоколов, которые требуют, чтобы адреса узлов или хост-машин назначал администратор сети, устройства в сети AppleTalk определяют свой адрес узла динамически. Как и для сетевой части адреса, методы адресации AppleTalk фазы 1 и фазы 2 выдвигают различные требования к выбору адреса узла во время переговоров.

Сегменты сети AppleTalk фазы 1 могут иметь до 254 адресов узлов: 127 резервируются для рабочих станций и 127 — для серверов. Каждая рабочая станция или сервер в сетевом сегменте с адресацией по методу фазы 1 должны иметь уникальный номер узла. Поэтому метод адресации AppleTalk фазы 1 мог поддерживать только 127 AppleTalk-хост-машин. Как оказалось, это вызывает проблему с масштабируемостью, решаемую в методе адресации AppleTalk фазы 2.

Сетевые сегменты с адресацией по методу AppleTalk фазы 2 классифицируются по адресам узлов на две категории, называемые *расширенными сегментами* и *нерасширенными сегментами.* В нерасширенных сетевых сегментах с адресацией по методу фазы 2 с одним сетевым адресом сегмента могут быть связаны 253 номера узла. Каждому серверу или рабочей станции присваивается уникальный номер узла, значение которого лежит в диапазоне от 1 до 253. Расширенные сетевые сегменты с адресацией по методу фазы 2 также позволяют назначать адреса узлов от 1 до 253. Однако, поскольку в сегменте могут существовать несколько сетевых номеров (благодаря кабельному диапазону), каждой рабочей станции или серверу присваивается уникальная комбинация сетевого адреса и адреса узла. Возможно, разница между расширенными и нерасширенными адресами выглядит трудноуловимой. Если говорить коротко, то расширенная сеть может поддерживать несколько сетевых номеров, а нерасширенная — только один-единственный сетевой адрес.

Примечание

Нерасширенные сетевые сегменты с адресацией по методу фазы 2 обычно представляют собой либо сети LocalTalk, либо сегменты глобальной сети. LocalTalk является первой реализацией сетевого взаимодействия на канальном и физическом уровнях с использованием телефонного кабеля в качестве физической транспортной среды и метода множественного доступа с контролем несущей и обнаружением конфликтов на канальном уровне (CSMA/CD). LocalTalk и AppleTalk фазы 1 были разработаны для приложений масштаба рабочей группы. Метод AppleTalk фазы 2 позволил улучшить

масштабируемость протокола AppleTalk для поддержки работы на уровне всего предприятия. Поскольку многие одинаковые характеристики свойственны сетевым сегментам AppleTalk фазы 1 и нерасширенным сетевым сегментам фазы 2, можно считать, что сетевой сегмент с адресацией по методу AppleTalk фазы 1 представляет собой нерасширенный сетевой сегмент с адресацией по методу фазы 2.
Маршрутизаторы компании Cisco никогда не поддерживали протокол LocalTalk, хотя сегменты глобальной сети могут адресоваться в стиле AppleTalk фазы 1 Однако ради достижения совместимости, ясности и гибкости рекомендуется при конфигурировании устройств компании Cisco использовать исключительно адресацию по методу фазы 2.

Как упоминалось ранее, переговоры по согласованию адреса узла проводятся динамически во время загрузки или перезапуска устройства, исполняющего протокол AppleTalk. Ответственным за переговоры по согласованию адресов узлов для устройств в сегменте сети является протокол AARP Динамическое назначение адреса выполняется с применением очень простого алгоритма. Каждый раз, когда устройство, на котором исполняется протокол AppleTalk, перезагружается и пытается подсоединиться к сети, оно проверяет, остался ли еще тот сетевой адрес, который был ему назначен ранее. Если это так, то устройство посылает AARP-пакет, чтобы удостовериться, что адрес еще корректен и не был заявлен другим узлом сегмента сети. Если адрес доступен, он используется, и узел начинает нормальную работу в сети Если адрес был заявлен, то узел отсылает ряд дополнительных AARP-пакетов, предлагая новый адрес узла до тех пор, пока не будет найден корректный адрес. На рис. 5.2 иллюстрируется процесс переговоров по согласованию адреса.

Для того чтобы улучшить взаимодействие между пользователем и сетью AppleTalk, компания Apple избавила пользователей от изучения специфики адресации сетей и узлов. Вместо информации о том, что рабочая станция 5 в сети 10 хочет связаться с сервером 8 в сети 20, пользователю достаточно узнать имена устройств. Компания Apple создала схему именования, которая позволяет логически группировать рабочие станции и назначать индивидуальные имена отдельным рабочим станциям и серверам. Логическая группа рабочих станций или серверов называется *зоной*

Определения зон могут задаваться исходя из любой логической характеристики организации, например, это могут быть ее производственные операции, подразделения или территориальное место расположения. Например, компания может создать маркетинговую, торговую и техническую зону, каждая из которых территориально будет находиться в нескольких местах. Или компания может иметь нью-йоркскую зону, которая обладает всеми функциями организации на заданной территории. Выбор и именование зон находятся полностью на усмотрении администратора сети. Чтобы учесть возможность логического группирования устройств, находящихся в нескольких физических сегментах локальной или глобальной сети, администратор может отнести к одной зоне несколько сетей. Кроме того, сегменту сети может быть назначено несколько имен зон, учитывая то, что логические группы могут иметь сетевые ресурсы, находящиеся в этом сегменте.

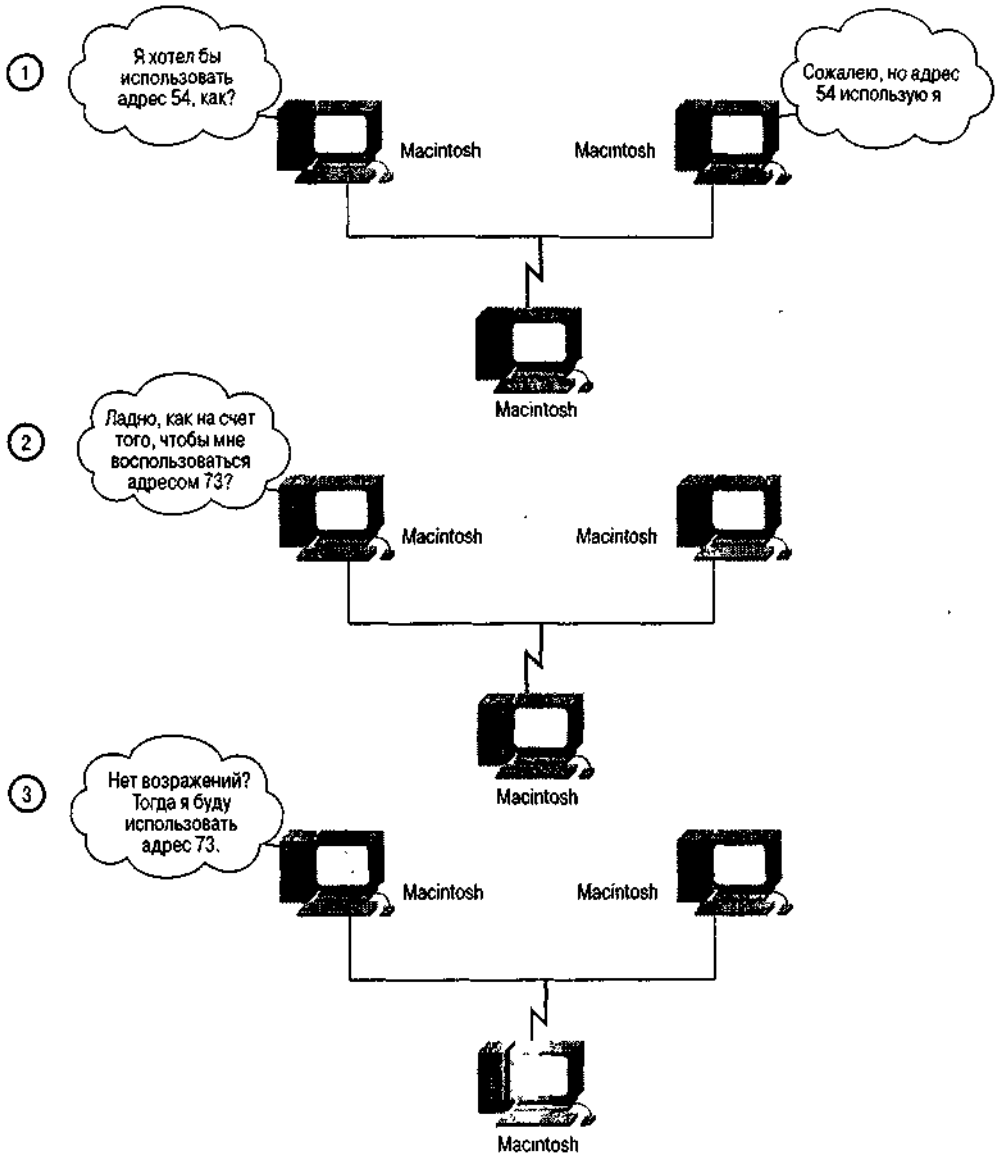


Рис. 5.2. Процесс выбора адреса узла в протоколе AppleTalk

В отличие от имен зон, которые задаются администратором сети, имена отдельных рабочих станций и серверов задаются пользователем или администратором этого устройства. Человек может дать рабочей станции имя, например, "компьютер Макинтош Джона" или "Годзилла", тогда как администратор может назвать сервер по его функции, например, "Финансы" или "Публикации". Эти имена вместе с зоной, к которой они относятся, регистрируются в сети с помощью протокола NBP сразу после запуска устройства.

Протокол NBP связывает имена и атрибуты устройств с адресами. Фактически он дирижирует процессом привязки имен, включающим регистрацию имени, его подтверждение, удаление и поиск. После того как имена будут зарегистрированы в протоколе NBP, приведенный ранее пример с рабочей станцией 10.5, желающей связаться с сервером 20. 8, может быть описан следующим образом: компьютер "Макинтош Джона" из нью-йоркской зоны хочет связаться с сервером "Финансы" из зоны "Бухгалтерия". Протокол NBP позволяет пользователям обращаться к сетевым ресурсам по именам, что во многом похоже на то, что делает служба имен домена (DNS) протокола TCP/IP.

Примечание

Чтобы зарегистрироваться в протоколе NBP, работающие под управлением ОС IOS устройства используют имена, назначаемые глобальной командой hostname. Маршрутизатор компании Cisco регистрируется в рамках протокола NBP как устройство типа

ciscoRouter. Привязки, сделанные протоколом NBP, можно увидеть, воспользовавшись командой ОС IOS режима EXEC show appietalk nbp. Эта команда будет рассмотрена в разделе "Верификация взаимодействия в сети с протоколом AppleTalk и устранение неполадок".

Хотя назначение имени зоны не является частью сетевого адреса, это — неотъемлемая часть правильной работы сети AppleTalk. Конфигурирование протокола AppleTalk на маршрутизаторах требует назначения зон в дополнение к присвоению номеров сети и кабельных диапазонов.
В табл. 5.1 приведены различия между сетями и требованиями к нумерации узлов.

Таблица 5.1. Технические характеристики протоколов AppleTalk фазы 1 и фазы *2*

Характеристика	**AppleTalk фазы 1**	**AppleTalk фазы 2**
Сети, узлы и зоны		
Количество логических сетей (кабельных сегментов)	1	65279
Максимальное количество устройств	254*	253**
Максимальное количество конечных узлов	127	Нет ограничений на количество конечных узлов; общее количество узлов не более 253
Максимальное количество серверов	127	Нет ограничений на количество конечных узлов; общее количество узлов не более 253
Количество зон, в которых может существовать сеть	1	1 (нерасширенная); 255 (расширенная)
Инкапсуляция на уровне физической среды		
Нерасширенная сеть	Понятие отсутствует	Да
Расширенная сеть	Понятие отсутствует	Да
Адресация кабельной системы	Понятие отсутствует; используются номера сетей	Один сетевой номер (нерасширенная сеть); кабельный диапазон от 1 и более (расширенная сеть)

- Номера узлов 1 и 255 — резервные.

** Номера узлов 0, 254 и 255 — резервные.

Конфигурирование адресов для протокола AppleTalk

В этом разделе исследуется конфигурирование адресов для протокола AppleTalk на интерфейсах локальных и глобальных сетей. Перед тем как приступать к назначению адресов, необходимо разработать разумную общую схему адресации сети. Адресация подчиняется нескольким правилам.

- Каждый сегмент локальной или глобальной сети должен иметь один единственный номер сети.
- Каждый сегмент локальной или глобальной сети должен иметь свое значение кабельного диапазона; не допускается повторение кабельного диапазона или его части для других сегментов сети.
- Кабельный диапазон с одним сетевым номером должен назначаться интерфейсам глобальных сетей.
- Рекомендуется следовать правилу добавления в кабельном диапазоне одного сетевого номера на каждые 50 узлов сегмента сети.
- Использование схемы адресации логических сетей может упростить устранение неполадок в

будущем.

Последняя рекомендация может показаться очевидной, но давайте рассмотрим этот момент немного подробнее на примере схемы назначения адресов для сети компании ZIP, находящейся в Сан-Франциско. В табл. 5.2 показано присвоение адресов в сети AppleTalk компании ZIP, расположенной в Сан-Франциско.

В сети компании ZIP для определенных рабочих площадок были зарезервированы диапазоны сетевых адресов. В данном случае вся адресация сети в Сан-Франциско лежит в диапазоне номеров сети 1—1000. В процессе устранения неисправностей сетевые администраторы компании ZIP, основываясь на сетевом адресе устройства, могут быстро определить, что оно находится в Сан-Франциско. Аналогично, диапазон 1~ 1000 был разбит на более мелкие части. Диапазон 1-10 был закреплен за серверным магистральным каналом данных, а диапазон 11—900 резервировался за этажами здания, на которых находятся пользовательские рабочие станции и принтеры. Наконец, диапазон 901—1000 был назначен для адресации каналов глобальных сетей.

Таблица 5.2. Назначение адресов в сети AppleTalk компании ZIP, расположенной в Сан-Франциско

Кабельный диапазон	Географическое расположение	Этаж	Ресурс
1-10	Сан-Франциско	Независимо	Кольцо FDDI и серверный магистральный канал
11-100	Сан-Франциско	1-й этаж	Пользовательские рабочие станции
101-200	Сан-Франциско	2-й этаж	Пользовательские рабочие
201-900	Сан-Франциско	Резерв для потенциального роста	Резерв для потенциального роста
901-901	Сан-Франциско -Сан-Хосе	-	Канал глобальной сети
902-902	Сан-Франциско - Сеул	-	Канал глобальной сети
903-1000	Сан-Франциско – внешний мир	Резерв для роста в будущем	Неназначенные глобальные сети

Как видно, подобный логический подход к назначению сетевых адресов позволяет быстро распознавать функции и местонахождение устройств в сети AppleTalk.

Конфигурирование интерфейсов локальных сетей

Все маршрутизаторы компании Cisco, которые осуществляют маршрутизацию с использованием протокола AppleTalk, имеют уникальный адрес *сеть.узел* для каждого подключенного к ним сегмента локальной сети. Адрес *сеть.узел* определяется динамически на основе либо номера сети системы адресации фазы 1, либо присвоенного интерфейсу кабельного диапазона системы адресации фазы 2. Назначение уникальных адресов каждому интерфейсу позволяет маршрутизатору определять, какие сети подключены к какому интерфейсу и куда следует посылать пакеты, предназначенные для этих сетей.

Аналогично протоколу TCP/IP каждый из пяти типов локальных сетей (Ethernet/IEEE 802.3, Fast Ethernet, Gigabit Ethernet, Token Ring/IEEE 802.5 и FDDI), описанных в главе 3, "Основы интерфейсов устройств Cisco", поддерживает концепцию динамического отображения MAC-адреса адаптера локальной сети на AppleTalk-адрес, назначенный интерфейсу. Этот процесс преобразования адресов поддерживается протоколом AARP, который участвует и в процедуре динамического назначения адреса узлу. Когда станция, работающая по протоколу AppleTalk, хочет вступить в контакт с другой AppleTalk-станцией, находящейся в той же логической сети, но не знает адреса канального уровня этой станции, она посылает широковещательный запрос на предоставление
канального адреса для нужного AppleTalk-адреса. Каждая станция, находящаяся в этой логической сети, проверяет запрос и, если MAC-адрес станции соответствует запрашиваемому AppleTalk-адресу, отвечает своим MAC-адресом.

Аналогично протоколу разрешения адреса (ARP — Address Resolution Protocol) в протоколе

TCP/IP протокол AARP исключает необходимость знать MAC-адреса, принадлежащие логической сети станции, чтобы связываться с другими рабочими станциями или серверами. Однако многие протоколы глобальных сетей не поддерживают динамическое отображение адресов канального уровня на AppleTalk-адреса. Поэтому, чтобы обеспечить обмен данными с другими станциями через интерфейс глобальной сети, конфигурирование глобальной сети требует дополнительного конфигурирования AppleTaik-адреса.

Примечание

Протокол AppleTalk работает хорошо с каждым из описанных в главе 3 интерфейсов локальных сетей. Хотя протокол канального уровня в протоколе AppleTalk одинаков для всех типов локальной сети (IEEE 802.2 SNAP LLC), компания Apple называет свою реализацию протокола AppleTalk для каждой среды различными именами. Протокол AppleTalk для локальной сети Ethernet она называет EtherTalk, для локальной сети Token Ring — TokenTalk и для локальной сети FDDI — FDDITalk.

Компанией Apple также были присвоены имена каждому из протоколов канального уровня, поддерживающих протокол AppleTalk в этих средах. Эти протоколы включают протокол доступа к каналу протокола EtherTalk (EtherTalk Link Access Protocol — ELAP), протокол доступа к каналу протокола TokenTalk (TokenTalk Link Access Protocol — TLAP) и протокол доступа к каналу протокола FDDITalk (FDDITalk Link Access Protocol — FLAP). Основное различие между этими типами протоколов канального уровня заключается в том, как в них реализована SNAP-инкапсуляция. На интерфейсах Ethernet, FDDI и Token Ring SNAP-инкапсуляция включает заголовки, соответственно, в стандарте IEEE 802.3, в стандарте IEEE 802.5 или FDDI и заголовок протокола IEEE 802.2 SNAP LLC. Присваивание имен протоколам канального уровня упрощает обсуждение реализации протокола AppleTalk в таких средах и ссылку на драйвер, который необходим операционной системе компании Apple, чтобы поддерживать протокол AppleTalk.

Назначение номеров сетей AppleTalk фазы 1 на интерфейсах как локальных, так и глобальных сетей выполняется с помощью интерфейсной субкоманды ОС IOS appletalk address. Эта команда имеет параметром номер сети и узла в формате *сеть.узел*. Указываемый номер сети должен согласовываться с номером сети других активных маршрутизаторов, уже присутствующих в конфигурируемом сегменте локальной или глобальной сети. Этот номер является предлагаемым. Он может меняться в процессе динамических переговоров, который описывался ранее. Хотя компания ZIP предпочла исключительно адресацию по методу фазы 2, ниже приведен пример конфигурирования маршрутизатора SF-1 на работу неиспользуемого интерфейса ethtrnet 1 с адресом по методу протокола AppleTalk фазы 1:

```
SF-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
SF-1(config)#interface ethernet 1
SF-1(config-if)#appletalk address 201.1
SF-1(config-if)#^Z
```

В соответствии с методом адресации по протоколу AppleTalk фазы 2 кабельные диапазоны для интерфейсов локальных и глобальных сетей назначаются с помощью интерфейсной субкоманды ОС IOS appletalk cable-range. В качестве параметра эта команда воспринимает диапазон номеров в формате *начало-конец*, который указывает начальный и конечный номера адресов сети, подлежащих включению в кабельный диапазон. Указываемое значение кабельного диапазона должно согласовываться с аналогичными значениями других активных маршрутизаторов, входящих в конфигурируемый сегмент локальной или глобальной сети. В качестве необязательного параметра команды можно указать начальный адрес в формате *сеть.узел*, который будет использоваться во время динамических переговоров по согласованию адреса. В примере ниже показано конфигурирование для всех трех интерфейсов локальной сети маршрутизатора SF-2 кабельного диапазона AppleTalk фазы 2:

```
SF-2#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
SF-2 (config)#interface ethernet 0
SF-2 (config-if)#appletalk cable-range 151-200
SF-2 (config-if)#interface ethernet 1
SF-2 (config-if)#appletalk cable-range 101-150
SF-2 {config-if)#interface fddi 0
SF-2 (config-if)#appletalk cable-range 1-10
SF-2 (config-if)#^Z
```

После конфигурирования адресов для успешной установки адресации протокола AppleTalk следует сконфигурировать на интерфейсах имена зон с помощью интерфейсной субкоманды ОС IOS appletalk zone. Параметром этой команды является цепочка символов, которая и представляет собой имя зоны. Имя зоны может содержать буквенные, цифровые и специальные символы, а также символы из специального набора символов компьютеров семейства Macintosh. Имена зон зависят от регистра символов.

Путем многократного повторения команды appletalk zone на заданном интерфейсе можно определить несколько имен зон. Имя первой заданной зоны считается первичным для этого интерфейса. Конфигурация зоны интерфейса должна точно совпадать по имени и по количеству зон с зонами, которые уже были сконфигурированы на активных и работающих с протоколом AppleTalk маршрутизаторах, расположенных в этом же сегменте сети. В примере, приведенном ниже, каждый из интерфейсов маршрутизатора SF-2 конфигурируется на одно имя зоны:

```
SF-2#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
SF-2 (config)#interface ethernet 0
SF-2 (config-if)#appletalk zone Marketing
SF-2 (config-if)#interface ethernet 1
SF-2 (config-if)#appletalk zone Sales
SF-2 (config-if)#interface fddi 0
SF-2 (config-if)#appletalk zone SF Zone
SF-2 (config-if)#^Z
```

Протоколы AppleTalk и ОС IOS компании Cisco поддерживают концепцию динамического конфигурирования сетевого адреса и имени (имен) зоны на интерфейсах локальной сети. Основой для этого служит информация, доступная из активных маршрутизаторов, которые уже присутствуют в сегменте сети. Динамическое конфигурирование выполняется путем перевода интерфейса в режим поиска. Режим поиска -используется, если в сегменте сети уже есть маршрутизатор, в котором его сетевой адрес и имя (имена) зоны были заданы вручную с помощью команд конфигурирования (это могут быть команды, которые воспринимаются маршрутизаторами, работающими под управлением ОС IOS, либо команды, воспринимаемые маршрутизаторами других типов). Новые маршрутизаторы, добавляемые в этот сегмент сети, просто получают конфигурацию из такого сконфигурированного маршрутизатора.

Режим обнаружения также позволяет легко переконфигурировать все маршрутизаторы, находящиеся в сегменте сети, так как только один конфигурируемый вручную маршрутизатор, так называемый *маршрутизатор посева*, требует ручного конфигурирования. Остальные работающие в режиме поиска маршрутизаторы этого сегмента сети просто перестраиваются на новую конфигурацию от реконфигурированного маршрутизатора посева.

Для нормальной работы режима поиска необходимо, чтобы в сегменте сети присутствовал, по крайней мере один, маршрутизатор посева. Если в сегменте сети все маршрутизаторы переводятся в режим поиска, то ни один из них не сможет установить конфигурацию протокола AppleTalk и начать передавать трафик в рамках этого протокола.

Конфигурирование режима поиска протокола AppleTalk осуществляется с помощью

интерфейсной субкоманды ОС IOS appletalk discovery. Обычно эта команда используется вместо команд appletalk address и appletalk cable-range. Как вариант, режим поиска можно включить, задав адрес *сеть.узел* 0.0 в качестве параметра команды appletalk address или appletalk cable-range. Вне зависимости от команды, включающей режим поиска, команда appletalk zone не используется. Хотя в сети компании ZIP режим поиска не применяется, ниже показан пример конфигурирования протокола AppleTalk на интерфейсе tokenring 0/0 маршрутизатора, расположенного в Сан-Хосе:

```
San-Jose#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
San-Jose(config)#interface tokenring 0/0
San-Jose(config-if)#appletalk discovery
San-Jose(config-if)^Z
```

Конфигурирование интерфейсов глобальных сетей

Адресация глобальных сетей в протоколе AppleTalk аналогична адресации локальных сетей, т.е. адреса конфигурируются с помощью субкоманд конфигурирования интерфейса appletalk address и appletalk cable-range и команды aplletalk zone. Однако режим поиска протокола AppleTalk не поддерживается ни одним из интерфейсов глобальной сети. В данном разделе рассматриваются назначения AppleTalk-номеров сети двухточечным и многоточечным интерфейсам глобальных сетей. Заметим, что для работы интерфейса глобальной сети требуется специальный метод инкапсуляции (например, X.25 или Frame Relay), используемый протоколом AppleTalk. Все интерфейсы глобальной сети требуют наличия уникальных AppleTalk-адресов узла, но конкретные интерфейсы в одной глобальной сети коллективно используют значение кабельного диапазона и имя зоны.

Адресация двухточечного интерфейса глобальной сети

Как отмечалось в главе 4 при обсуждении протокола IP, двухточечный интерфейс глобальной сети соединяет два устройства. Чтобы два маршрутизатора могли направлять AppleTalk-пакеты на двухточечный интерфейс, они оба должны иметь на подключенных к сети интерфейсах одинаковые значения номера сети или кабельного диапазона. Как и в локальной сети, каждое устройство, подключенное к интерфейсу глобальной сети, обладает динамически заданным и уникальным AppleTalk-номером узла.

Конфигурирование AppleTalk-номеров сети на двухточечных интерфейсах глобальной сети выполняется с помощью интерфейсной субкоманды ОС IOS aplletalk address для адресов по методу фазы 1 или интерфейсной субкоманды ОС IOS appletalk cable-range для адресов по методу фазы 2. Каждому отдельному двухточечному соединению глобальной сети (или двухточечному подынтерфейсу) должен назначаться отдельный AppleTalk-номер сети или кабельный диапазон. Имена AppleTalk-зон также могут присваиваться двухточечным интерфейсам глобальной сети с помощью интерфейсной субкоманды ОС IOS appletalk zone. В примере, приведенном ниже, маршрутизатор Seoul-1 конфигурируется значениями кабельных диапазонов и именами зон для каждого из его двухточечных интерфейсов (два подинтерфейса Frame Relay и один интерфейс высокоуровневого протокола управления каналом передачи данных (High-Level Data Link Control — HDLC)):

```
Seoul-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Seoul-1 (config)#interface serial 0.16 point-to-point
Seoul-1 (config-if)#appletalk cable-range 2901-2901
Seoul-1 (config-if)#appletalk zone WAN Zone
Seoul-1 (config-if)#interface serial 0.17 point-to-point
Seoul-1 (config-if)#appletalk cable-range 2902-2902
Seoul-1 (config-if)#appletalk zone WAN Zone
Seoul-1 (config-if)#interface serial 1
```

```
Seoul-1 (config-if)#appletalk cable-range 1901-1901
Seoul-1 (config-if}#appletalk zone WAN Zone
Seoul-1 (config-if)#^Z
```

Адресация многоточечного интерфейса глобальной сети

Общие моменты, связанные с конфигурированием адресов сетевого протокола на многоточечных интерфейсах глобальной сети обсуждались в главе 4 на примере протокола IP. Как и протокол IP, протокол AppIeTalk может использоваться с различными интерфейсами глобальной сети, включая Frame Relay, X.25, ISDN и ATM. Каждый из этих интерфейсов конфигурируется на маршрутизацию AppleTalk-пакетов с помощью интерфейсных субкоманд ОС IOS appletalk address или appletalk cable-range. Как и для ранее рассмотренных интерфейсов, правильная работа протокола AppIeTalk на многоточечных интерфейсах требует субкоманды ЭС IOS appletalk zone.

Протоколу AppIeTalk также необходимо отображение конкретного адреса канального уровня на конкретный адрес формата *сеть.узел* протокола AppIeTalk. Такое отображение конфигурируется по разному для каждого протокола глобальной сети. Команды, используемые для выполнения таких отображений, требуют наличия конкретного адреса *сеть.узел*. Чтобы обеспечить администратору сети точное представление о том, каким маршрутизаторам многоточечной глобальной сети какие адреса узлов назначены, рекомендуется указывать адреса сеть.узел в качестве параметров команд appletalk address и appletalk cable-range.

При работе с многоточечными интерфейсами Frame Relay маршрутизатор нуждается в отображении идентификационных номеров соединений канального уровня (DLCI-идентификаторов) на многоточечном интерфейсе Frame Relay на номер *сеть.узел* протокола AppleTalk. Динамическое отображение DLCI-идентификатора на номер сети и узла протокола AppleTalk может выполнять функция обратного разрешения адресов Frame Relay Inverse ARP. Для статического отображения DLCI-адреса, связанного с сетью и номером узла протокола AppleTalk, которые достижимы через многоточечный интерфейс глобальной сети, можно также воспользоваться субкомандой конфигурирования интерфейса frame-relay map appletalk.

Адресация многоточечных интерфейсов глобальной сети X.25 похожа на адресацию интерфейсов Frame Relay тем, что в обоих случаях статическое отображение реализуется через субкоманды конфигурирования интерфейса. Интерфейсы X.25 должны иметь отображения своих адресов *сеть.узел* протокола AppleTalk на адреса протокола X. 121, используемые для настройки виртуальных каналов между системами. Каждый виртуальный канал идентифицируется адресом протокола X.121, используемым для соединения. Чтобы выполнить на многоточечном интерфейсе глобальной сети статическое отображение между адресом протокола AppleTalk и адресом протокола X.121, используется субкоманда конфигурирования интерфейса x25 map appletalk.

Адресация многоточечных интерфейсов ISDN также требует команд статического отображения. Однако для протокола AppleTalk, в отличие от протокола IP, ISDN-команды отображения требуются для каждого устройства, которое хочет обмениваться данными с другим устройством через ISDN-соединение. Для отображения адресов *сеть.узел* протокола AppleTalk на имена систем и телефонные номера, используемые для соединения в ISDN-сети, применяется субкоманда конфигурирования интерфейса ОС IOS dialer map appletalk.

Отображение адресов канального уровня протокола ATM в виде идентификаторов виртуального пути/идентификаторов виртуального канала на номер *сеть.узел* протокола AppleTalk на многоточечном интерфейсе ATM зависит от используемых типов ATM-протоколов и виртуальных каналов. При работе с протоколом AppleTalk как для постоянных виртуальных каналов, так и для коммутируемых виртуальных каналов ATM-сети можно использовать инкапсуляцию протокола логического управления связью/протокола управления доступом к сети (LLC/SNAP). В режиме работы с постоянными виртуальными каналами в ATM-сети организуется постоянный виртуальный канал, и пакеты идентифицируются как направленные на AppleTalk-адрес на другом конце конкретного виртуального канала. В режиме работы с коммутируемыми виртуальными каналами AppleTalk-пакеты идентифицируются как направленные на конкретный статически заданный ATM-адрес канального уровня. ATM-

коммутатор устанавливает виртуальный канал по требованию, когда маршрутизатор запрашивает соединение с ATM-адресом для конкретного AppleTalk-адреса *сеть.узел*.

Процесс LLC/SNAP-инкапсуляции в постоянных виртуальных каналах использует субкоманду конфигурирования интерфейса ОС IOS map-group и команду глобального конфигурирования ОС IOS map-list для отображения AppleTalk-адресов *сеть.узел* на конкретные постоянные виртуальные каналы. Процесс LLC/SNAP-инкапсуляции в коммутируемых виртуальных каналах использует субкоманду конфигурирования интерфейса ОС IOS map-group и команду глобального конфигурирования ОС IOS map-list для отображения AppleTalk-адресов на адреса точек доступа к сетевой службе (Network Service Access Point — NSAP), применяемых для идентификации удаленных устройств в ATM-сети.

Проверка конфигурации AppleTalk-адресов

Верификация AppleTalk-адресов и других атрибутов протокола AppleTalk, которые были присвоены интерфейсам, может выполняться с помощью команды режима :XEC show appletalk interface. Эта команда дает полное представление о пара-1етрах, связанных с конфигурацией протокола AppleTalk на всех интерфейсах. Если в качестве параметра команды указывается конкретный интерфейс, на экран выводится только та информация, которая относится к этому интерфейсу. Ниже приведен результат исполнения команды show appletalk interface ethernet 0 на маршрутизаторе компании ZIP SF-2:

```
F-2#show appletalk interface ethernet 0
EthernetO is up, line protocol is up AppleTalk cable range is 151-200
   AppleTalk address is 198.72, Valid
   AppleTalk zone is "Marketing "
   AppleTalk address gleaning is disabled
   AppleTalk route cache is enabled
```

В первой строке выводимых данных показан административный и рабочий статус интерфейса. При подтверждении конфигурации протокола AppleTalk на этом интерфейсе другими активными маршрутизаторами, принадлежащими сегменту сети, информация о статусе выводится выше или в этой строке. Во второй строке приводится значение кабельного диапазона сетевого адреса AppleTalk-сети. третьей строке стоит адрес *сеть.узел* и указывается, конфликтует ли этот адрес с каким-либо другим адресом на этом интерфейсе. В четвертой строке показывает имя зоны, которой этот интерфейс принадлежит. Результат может содержать и дополнительные строки, если активны какие-либо другие функции протокола AppleTalk, например, фильтры пакетов. Подобная ситуация будет рассмотрена в этой главе в разделе "Конфигурирование фильтрации в протоколе AppleTalk с применением списков доступа".

Команда ОС IOS режима EXEC show appletalk interface имеет дополнительную форму, которая позволяет увидеть краткую сводную информацию об AppleTalk-адресах и статусах всех имеющихся в устройстве интерфейсов. Такая суммирующая версия данных может быть получена с помощью команды show appletalk interface brief.

Ниже показан результат исполнения команды show appletalk interface brief на маршрутизаторе компании ZIP SF-2:

```
SF-2#show appletalk interface brief
Interface          Address      Config        Status/LineProtocol   Atalk    Protocol
EthernetO          198.72       Extended            up              up
Ethernetl          120.45       Extended            up              up
FddiO               7.12        Extended            up              up
Loopbackl          unassigned   not config'd        up              n/a
```

Кроме проверки конфигурации протокола AppleTalk на самом маршрутизаторе, можно просматривать как статические, так и динамические отображения AppleTalk-адресов *сеть.узел* на адреса канального уровня для различных сред многоточечных глобальных сетей. Для этого следует воспользоваться командами ОС IOS режима EXEC show frame-relay map, show atm map и show

dialer map, что уже демонстрировалось в предыдущих главах.

Конфигурирование маршрутизации по протоколу AppleTalk

Назначение AppleTalk-номеров сетей и имен зон работающим под управлением ОС IOS устройствам и интерфейсам — необходимое, но не достаточное условие для того, чтобы устройства могли обмениваться друг с другом информацией по протоколу AppleTalk. Для двусторонней передачи данных рабочие станции и серверы в сети AppleTalk должны также знать пути, по которым они могут связаться друг с другом. AppleTalk-маршрутизаторы создают и постоянно обращаются к таблицам номеров сетей, известным под названием *таблиц маршрутизации.* Таблицы маршрутизации протокола AppleTalk работают во многом так же, как таблицы маршрутизации протокола IP, обеспечивая информацию о сетевом пути, что позволяет маршрутизатору доставлять данные непосредственно в пункт конечного назначения или следующему маршрутизатору, стоящему по пути к пункту назначения. Для определения места нахождения AppleTalk-сетей и совместного использования этой информации маршрутизаторы применяют алгоритмы маршрутизации, также известные под названием протоколов маршрутизации.

В рамках протокола AppleTalk протоколы маршрутизации могут иметь либо статическую, либо динамическую природу. В статических протоколах таблица AppleTalk-маршрутизации конфигурируется информацией о сетевых путях вручную. Динамические протоколы маршрутизации полагаются на сами маршрутизаторы, которые объявляют друг другу информацию о различных AppleTalk-сетях, к которым они подсоединены. Протокол AppleTalk использует два различных протокола динамической маршрутизации, которые будут рассматриваться в этой главе в разделе "Конфигурирование протоколов динамической маршрутизации, работающих с протоколом AppleTalk".

Команды конфигурирования маршрутизации по протоколу AppleTalk

До того как маршрутизатор сможет конфигурироваться информацией протокола AppleTalk и начнет пропускать AppleTalk-трафик, функция маршрутизации по протоколу AppleTalk должна быть активирована. Работающие под управлением ОС IOS устройства не разрешают автоматически выполнять маршрутизацию по протоколу AppleTalk.

Чтобы разрешить маршрутизацию по протоколу AppleTalk, используется команда глобального конфигурирования ОС IOS appletalk routing. В примере ниже выполняется разрешение маршрутизации по протоколу AppleTalk на маршрутизаторе компании ZIP SF-2:

```
SF-2#configure
Configuring  from terminal,   memory,   or network   [terminal]? Enter
configuration commands,   one per  line.   End with CTRL+Z.
SF-2   (config)#appletalk routing
SF-2(config)#^Z
```

После того как маршрутизация по протоколу AppleTalk разрешена, маршрутизатор строит таблицу маршрутизации, которая используется для коммутации пакетов. По умолчанию, когда интерфейс локальной или глобальной сети конфигурируется AppleTalk-адресом или значением кабельного диапазона, и этот интерфейс переводится в рабочее состояние, информация о AppleTalk-сети этого интерфейса помещается в таблицу маршрутизации. В таблицу маршрутизации заносится информация всех интерфейсов, подключенных к маршрутизатору. Если в сети находится только один маршрутизатор, то он обладает информацией обо всех подключенных к нему AppleTalk-сетях. Записи таблицы динамической маршрутизации создаются только тогда, когда в сети присутствует несколько маршрутизаторов. Для записей таблицы динамической маршрутизации используется протокол управления таблицей маршрутизации (Routing Table Maintenance Protocol — RTMP).

Для просмотра таблицы маршрутизации протокола AppleTalk можно использовать команду ОС IOS режима EXEC show appletalk route. Если эта команда вводится без указания параметров, то на экран выводится вся таблица маршрутизации протокола AppleTalk. В примере ниже показана информация таблицы маршрутизации маршрутизатора SF-2 сети компании ZIP, в которой есть только данные о подключенных активных интерфейсах и нет никаких дополнительных записей:

```
SF-2#show appletalk route
Codes:   R - RTMP derived,   E - EIGRP derived, C -connected, A - AURP S - static
         P — proxy 3  routes  in internet
The first  zone listed for each entry is its default   (primary)   zone.
C Net  1-10 directly connected,   FddiO,   zone SF Zone
C Net  101-150 directly connected,   Ethernetl,   zone Sales
C Net  151-200 directly connected,   EthernetO,   zone Marketing
```

Команда show appletalk route обеспечивает администратору сети много полезных данных. Она является основным инструментом, используемым для выяснения путей, которыми AppleTalk-пакеты проходят через сеть. Выводимая с помощью этой команды информация аналогична информации, выводимой командой show ip route, которая показывает содержимое таблицы IP-маршрутизации, что рассматривалось в главе 4.

Первая часть выводимой информации представляет собой легенду первого столбца таблицы. Она рассказывает о том, откуда был получен маршрут. Каждая из трех последних строк этой таблицы маршрутизации протокола AppleTalk показывает один маршрут к множеству AppleTalk-сетей, заданному кабельным диапазоном, как этот маршрут был получен, зоны, которым принадлежат сети, и интерфейс, связанный с маршрутом. Буква "C" в первом столбце указывает на то, что все маршруты определены из находящихся в рабочем состоянии подключенных AppleTalk-сетей. Более подробно команда show appletalk route будет рассмотрена в этой главе в разделе "Проверка конфигурации маршрутизации по протоколу AppleTalk".

Конфигурирование статической маршрутизации

При обсуждении в главе 4 IP-маршрутизации были названы различные причины для использования статических IP-маршрутов, включая нестабильность сетевых каналов и соединений по коммутируемым линиям связи. То же применимо и к статическим AppleTalk-маршрутам. Для конфигурирования в AppleTalk-таблице маршрутизации статических AppleTalk-маршрутов можно использовать команду глобального конфигурирования appletalk static. В примере ниже маршрутизатор SF-2 компании ZIP конфигурируется статическим маршрутом, направляющим AppleTalk-пакеты, пункт назначения которых — сеть 40000-40000, в узел 5.10, находящийся на интерфейсе FDDI. В этом примере имя зоны SF Zone связывается с кабельным диапазоном 40000-40000 командой appletalk static.

```
SF-2#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
SF-2 (config)#appletalk static cable-range 40000-40000 to 5.10 zone SF Zone
SF-2 (config)t^Z
```

Используя команду show appletalk route, можно проверить наличие записи о статическом маршруте в таблице маршрутизации маршрутизатора SF-2:

```
SF-2#show appletalk route
Codes: R — RTMP derived, E — EIGRP derived, C -connected,   A - AURP S —
        static P — proxy
4 routes in internet
The first zone listed for each entry is its default (primary) zone.
```

```
C Net 1-10 directly connected, FddiO, zone SF Zone
C Net 101-150 directly connected, Ethernetl, zone Sales
C Net 151-200 directly connected, EthernetO, zone Marketing
S Net 40000-40000 [1/G ] via 5.10, 315 sec,FddiO, zone SF Zone
```

Информацию о статических AppleTalk-маршрутах также можно просматривать с помощью команды ОС IOS режима EXEC show apple static:

```
SF-2#show apple static
                AppleTalk       Static          Entries:
                ------------------------------------------------------------
Network         NextIR          Zone            Status
40000-40000     5.10            SF Zone         A
```

Проверка конфигурации маршрутизации по протоколу AppleTalk

Как уже говорилось, конфигурация AppleTalk-маршрутизации может верифицироваться с помощью команды ОС IOS режима EXEC show appletalk route. В данном разделе рассмотрены дополнительные команды, которые помогают в верификации и управлении конфигурацией таблицы маршрутизации протокола AppleTalk.

Команда show appletalk route используется для просмотра состояния таблицы маршрутизации протокола AppleTalk. Конфигурируются ли статические маршруты или исполняются протоколы динамической маршрутизации, эта команда показывает, присутствуют ли реально на маршрутизаторе сконфигурированные ранее маршруты, или ожидается, что они будут получены в процессе обучения. Протоколы динамической AppleTalk-маршрутизации будут рассматриваться в следующем разделе этой главы. А ниже показана выдержка из информации, выведенной командой show appletalk route на маршрутизаторе компании ZIP SF-2:

```
SF-2#show appletalk route
Codes: R — RTMP derived, E — EIGRP derived, C -connected, A - AUR, S -
          static, P — proxy 5 routes in internet
The first zone listed for each entry is its default (primary) zone.
C Net 1-10 directly connected, FddiO, zone SF Zone
C Net 101-150 directly connected, Ethernetl, zone Sales
C Net 151-200 directly connected, EthernetO, zone Marketing
R Net 11-100 [1/G] via 2.12, 10 sec, FddiO, zone Operations
S Net 40000-40000 [1/G] via 5.10, 315 sec, FddiO, zone SF Zone
```

В приведенной выше информации представлены данные о маршрутах к непосредственно соединенным с маршрутизатором SF-2 AppleTalk-сетям и о маршруте к AppleTalk-сети 11-100, полученном в процессе динамического обучения от маршрутизатора SF-1 с использованием протокола динамической AppleTalk-маршрутизации RTMP. Выводимые данные также предоставляют следующую информацию.

- Адрес *сеть.узел* AppleTalk-маршрутизатора следующего перехода и тип выходного интерфейса для показанных маршрутов (или просто выходной интерфейс для маршрутов через непосредственное подключение).
- Если маршрут был получен в процессе динамического обучения, продолжительность времени (в секундах), в течение которого маршрут находился в таблице, или продолжительность времени с момента последнего обновления, в зависимости от конкретного протокола маршрутизации.
- Метрика протокола маршрутизации (число слева от косой черты в квадратных скобках) и состояние маршрута, которое указывается для всех маршрутов, кроме маршрутов через непосредственное подключение. Состояние маршрута обозначается буквой справа от косой черты в квадратных скобках. Для обозначения хорошего маршрута (т.е. активного и доступного), подозрительного или плохого, используются буквы G, S и B, соответственно.
  Состояния маршрута контролируются отдельным процессом, проверяющим каждые 20 секунд конкретные маршруты, которые не обновлялись. После каждого 20-секундного перерыва в обновлении маршрута его статус изменяется с хорошего (G) на

подозрительный (S) и с подозрительного (S) на плохой (B). После истечения 1 минуты без обновлений маршрут удаляется. Каждый раз, когда принимается полезное обновление, статус маршрута восстанавливается до хорошего. Обновления называются *полезными,* если они объявляют маршрут хорошим или лучше, чем тот, что на данный момент находится в таблице.

Подобно команде show ip route, команда show appletalk route, если задать качестве параметра номер сети, позволяет просмотреть конкретный маршрут, также можно удалять AppleTalk-маршруты из таблицы маршрутизации, воспользовавшись привилегированной командой режима EXEC clear appletalk route, при устранении проблем, связанных с AppleTalk-маршрутизацией, можно сначала с помощью этой командой, удалить запись о маршруте, а затем использовать команду how appletalk route, чтобы проверить, откуда маршрутизатор первоначально шал об этом маршруте.

Конфигурацию имен зон протокола AppleTalk можно проверить, используя команду ОС 1OS режима EXEC show appletalk zone. Если в этой команде не указывается имя зоны в качестве параметра, то выводятся все имена зон. Ниже показана выдержка из результата, получаемого после выполнения команды show appletalk zone на маршрутизаторе компании ZIP SF-2:

```
SF-2#show appletalk zone
Name                    Network(s)
SF Zone                 1-10 40000-40000
Sales                   101-150
Marketing               151-200
Operations              11-100
Total of 4 zones
```

Если в сети находится несколько AppleTalk-маршрутизаторов, они обмениваются информацией динамической маршрутизации. Для того чтобы проверить, активирована ли AppleTalk-маршрутизация и есть ли в сети другие AppleTalk-маршрутизаторы, используется команда ОС IOS режима EXEC show appletalk neighbors. Ниже показан фрагмент результата, получаемого после выполнения команды show appletalk neighbors на маршрутизаторе компании ZIP SF-2. Из него видно, что маршрутизатор SF-2 знает о соседнем AppleTalk-маршрутизаторе SF-1 и что на маршрутизаторе SF-1 исполняется протокол динамической маршрутизации RTMP.

```
SF-2#show appletalk neighbors
AppleTalk neighbors:
2.12   SF-l.FddiO    FddiO,  uptime 33:27,  2 sees
Neighbor is reachable as a RTMP peer
```

Конфигурирование протоколов динамической маршрутизации, работающих с протоколом AppleTalk

Как указывалось в главе 4, на решение о том, какой протокол маршрутизации следует использовать в сети, оказывают влияние множество факторов. И эти факторы — топология сети, масштабируемость, легкость внедрения и скорость сходимости — также важны при выборе протокола динамической маршрутизации для протокола AppleTalk.

ОС IOS компании Cisco предлагает два протокола динамической маршрутизации для протокола AppleTalk: протокол управления таблицей маршрутизации (Routing Table Maintenance Protocol — RTMP) и AppleTalk EIGRP. В отличие от протокола TCP/IP, протокол AppleTalk имеет свой протокол динамической маршрутизации по умолчанию: RTMP, который работает без ручного конфигурирования администратором сети. Протокол AppleTalk EIGRP может устанавливаться на сетевых устройствах с ОС IOS на по сегментной основе. Однако, поскольку протокол EIGRP поддерживают только те устройства, которые работают под управлением ОС IOS, сегменты сети с AppleTalk-маршрутизаторами, которые не работают с ОС IOS, или сегменты со смешанными маршрутизаторами будут требовать для нормальной работы применения

протокола RTMP.

Исследуемая в качестве примера сеть компании ZIP использует как протокол RTMP, так и протокол E1GRP. Протокол EIGRP реализован для уменьшения потребления полосы пропускания каналов глобальной сети, которое имеет место при применении протокола RTMP. Если AppleTalk-сеть не слишком велика и имеет приемлемую полосу пропускания глобальной сети, то одного протокола RTMP, вероятно, будет достаточно. При этом исключается необходимость в конфигурировании дополнительных команд установки протокола EIGRP.

В последующих разделах рассматривается конфигурирование маршрутизации на основе протоколов RTMP и EIGRP.

Конфигурирование протокола AppleTalk RTMP

RTMP является протоколом динамической маршрутизации протокола AppleTalk по умолчанию. По своим функциям он аналогичен протоколу маршрутной информации Routing Information Protocol протокола IP (IP RIP). AppleTalk RTMP — это протокол маршрутизации, использующий метод вектора расстояния. Он формирует содержимое таблиц AppleTalk-маршрутизации, управляет им в маршрутизаторах, работающих с протоколом AppleTalk. Свойства протоколов маршрутизации на основе метода вектора расстояния и протокола IP RIP были рассмотрены в главе 4. AppleTalk RTMP относится к классу протоколов внутренних шлюзов (IGP). Протокол AppleTalk не использует протоколы маршрутизации класса протоколов внешних шлюзов (EGP), поскольку всегда применяется только во внутренних корпоративных сетях и никогда — в сетях общего пользования типа Internet. Активация протокола AppleTalk RTMP на AppleTalk-интерфейсах происходит по умолчанию в тот момент, когда вводится команда глобального конфигурирования appletalk routing.

Будучи первым протоколом динамической маршрутизации для сетей AppleTalk, протокол RTMP не обладает некоторыми усовершенствованными функциями новых протоколов динамической маршрутизации и, прежде всего, в области обеспечения масштабируемости и снижения потребления полосы пропускания. Одним из основных недостатков протокола RTMP является его чрезвычайно "болтливая" природа: он посылает пакеты актуализации маршрутизации каждые 10 секунд. Как будет видно из материала в следующем разделе, разработанные позднее протоколы динамической маршрутизации решают некоторые из этих проблем.

Подобно протоколу IP RIP, протокол AppleTalk RTMP использует метрику количества переходов. Количество переходов является мерой количества межмаршрутизаторных переходов, которые пакет должен пройти, чтобы проделать путь от своего источника до пункта назначения. Максимальное количество переходов, поддерживаемое протоколом AppleTalk RTMP, равно 30. Любой маршрут, который имеет более 30 переходов, отмечается как недоступный. В выводимой командой show appletalk route информации из маршрутизатора SF-2 видно, что маршрут до AppleTalk-сети 11-100 имеет метрику в 1 переход, что указывается в таблице AppleTalk-маршрутизации через обозначение [1/G]:

```
SF-2#show appletalk route
Codes:   R - RTMP derived, E - EIGRP derived, C - connected, A - AURP,
         S - static, P - proxy 5 routes in internet
The first zone listed for each entry is its default (primary) zone.
C Net 1-10 directly connected, FddiO, zone SF Zone
C Net 101-150 directly connected, Ethernetl, zone Sales
C Net 151-200 directly connected, EthernetO, zone Marketing
R Net 11-100 [1/G] via 2.12, 10 sec, FddiO, zone Operations
S Net 40000-40000 [1/G] via 5.10, 315 sec, FddiO, zone SF Zone
```

По умолчанию в любой конкретный момент времени в таблице маршрутизации содержится только один маршрут до конкретной AppleTalk-сети. Такое поведение отличается от IP-маршрутизации, при которой маршрутизатор автоматически сохраняет несколько путей с

равной стоимостью. Чтобы разрешить маршрутизатору помещать в свою таблицу AppleTalk-маршрутизации пути с равной стоимостью, используется команда глобального конфигурирования appletalk maximum-paths. Например, команда appletalk maximum-paths 2 позволяет маршрутизатору обучиться двум равностоимостным путям до пункта назначения в сети AppleTalk. Количество путей с равной стоимостью, на сохранение которых дается разрешение маршрутизатору, зависит от топологии сети AppleTalk. Если сохраняется несколько путей с равной стоимостью, нагрузка маршрутизатора на попакетной основе разделяется между всеми параллельными равностоимостными путями до пункта назначения в сети AppleTalk.

Конфигурирование протокола AppleTalk EIGRP

AppleTalk EIGRP представляет собой усовершенствованную версию исходного протокола внутренней маршрутизации между шлюзами (Interior Gateway Routing Protocol — IGRP) компании Cisco, адаптированную к использованию в AppleTalk-сетях. Протокол AppleTalk EIGRP применяет тот же транспортный механизм, алгоритм обновлений DUAL и процесс обнаружения соседей, что и протокол EIGRP для IP-маршрутизации, который обсуждался в главе 4. Протокол AppleTalk EIGRP обладает характеристиками, которые свойственны протоколам, основанным на методе учета состояния канала, например, частично инкрементальные пакеты актуализации и уменьшенное время сходимости. Протокол EIGRP посылает пакеты актуализации маршрутизации только в тех случаях, когда в топологии сети происходят изменения, и поэтому он занимает меньшую полосу пропускания, чем протокол RTMP, который посылает частые пакеты актуализации с полной таблицей маршрутизации. Применение протокола EIGRP в каналах глобальной сети, в частности, в тех, которые имеют ограниченную полосу пропускания, может улучшить производительность сети по трафику, проходящему через такие каналы.

Конфигурирование процесса маршрутизации по протоколу AppleTalk EIGRP состоит из двух этапов: разрешение маршрутизатору выполнить протокол EIGRP и идентификация интерфейсов, которые включаются в процесс EIGRP-маршрутизации.

Чтобы разрешить работу с протоколом AppleTalk EIGRP, используется команда глобального конфигурирования ОС IOS appletalk routing eigrp. В качестве параметра этой команды выступает идентификатор процесса, которым часто является номер автономной системы, используемый при конфигурировании протоколов IP EIGRP или IP BGP. В примере ниже маршрутизатору в Сингапуре разрешается работа с протоколом AppleTalk EIGRP, для чего используется номер автономной системы 25000:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore (config) #appletalk routing eigrp 25000
Singapore (config)#^Z
```

После того как работа протокола AppleTalk EIGRP будет разрешена, необходимо идентифицировать интерфейсы маршрутизатора, которые включаются в EIGRP-процесс обмена актуальной маршрутной информацией. Инструкция маршрутизатору о том, какой AppleTalk-протокол динамической маршрутизации использовать на конкретном интерфейсе, выдается с помощью субкоманды конфигурирования интерфейса ОС IOS appletalk protocol. Параметром у этой команды могут быть ключевые слова eigrp или rtmp. В сети компании ZIP работа протокола EIGRP была разрешена на всех интерфейсах глобальной сети. Ниже показан пример конфигурирования протокола EIGRP в качестве протокола маршрутизации на интерфейсе глобальной сети маршрутизатора компании ZIP в Сингапуре:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interface serial 0.100
Singapore(config-if)#appletalk protocol eigrp
Singapore(config-if)#^Z
```

Поскольку протокол RTMP сконфигурирован на всех AppleTalk-интерфейсах по умолчанию, то на те интерфейсы, на которых активирован протокол EIGRP, посылаются пакеты актуализации маршрутной информации как протокола EIGRP, так и протокола RTMP. Это можно проверить, воспользовавшись командой show appletalk interface, что и видно на примере маршрутизатора компании ZIP в Сингапуре:

```
Singapore#show appletalk interface serial 0.100
SerialO.100 is up, line protocol is up
     AppleTalk cable range is 2902-2902
     AppleTalk address is 2902.2,Valid
     AppleTalk zone is "WAN Zone "
     Routing protocols enabled: RTMP EIGRP
     AppleTalk address gleaning is not supported by hardware
     AppleTalk route cache is not initialized
```

Следует отметить, что после того, как на маршрутизаторе будет разрешена работа протокола AppleTalk EIGRP, начнется автоматическая редистрибуция маршрутной информации между протоколами AppleTalk EIGRP и RTMP. Этот процесс гарантирует взаимный обмен маршрутами, выявленными каждым протоколом динамической маршрутизации, так что и EIGRP- и RTMP-маршрутизаторам известны все доступные сетевые адреса. В конфигурацию маршрутизатора, настраиваемого на работу с протоколом AppleTalk EIGRP, автоматически вводится команда глобального конфигурирования ОС IOS appletalk route-redistribution, которая инициирует процесс редистрибуции. Преднамеренное отключение автоматической редистрибуции может привести к тому, что EIGRP-маршрутизаторы не будут знать о маршрутах, выявленных протоколом RTMP, и наоборот. Потенциальным следствием этого может быть недоступность некоторых сетевых ресурсов для определенных пользователей.

На интерфейсах, которые обслуживаются только маршрутизаторами, работающими под управлением ОС 1OS компании Cisco, RTMP-маршрутизация может быть отключена, чтобы исключить дублирующую рассылку пакетов актуализации маршрутной информации. Однако важно не отключать протоколы RTMP на тех интерфейсах, которые имеют AppleTalk-рабочие станции, серверы, принтеры или AppleTalk-маршрутизаторы, не использующие ОС IOS. Запрещение работы протокола RTMP на интерфейсах с такими устройствами лишит их доступа к сетевым службам протокола AppleTalk. В сети компании ZIP работа протокола RTMP была запрещена на всех интерфейсах глобальной сети, имеющих только AppleTalk-маршрутизаторы, которые работают под управлением ОС IOS. Чтобы запретить работу протокола RTMP, используется субкоманда конфигурирования интерфейса ОС IOS no appletalk protocol rtmp. Ниже приведен пример запрещения работы протокола RTMP на интерфейсе глобальной сети маршрутизатора компании ZIP в Сингапуре:

```
Singapore#configure
Configuring  from terminal,   memory,   or network   [terminal]?
Enter  configuration commands,   one  per  line.   End with CTRL+Z.
Singapore(config)#interface  serial  0.100
Singapore(config-if)#no appletalk protocol rtmp Singapore(config-if}#^z
```

Как только что отмечалось, работа протокола RTMP не может быть полностью запрещена на AppleTalk-интерфейсах, имеющих другие AppleTalk-рабочие станции конечных пользователей и AppleTalk-маршрутизаторы, работающие не под управлением ОС 1OS. Однако, если в сегменте сети находятся только AppleTalk-станции конечных пользователей (такой сегмент известен под названием глухой сети), рассылка полноформатных RTMP-пакетов актуализации маршрутной информации может быть заменена рассылкой модифицированных укороченных пакетов обновления маршрутной информации. Укороченная форма позволяет рабочим станциям, серверам и принтерам продолжать работу и не увеличивает накладных расходов в сети, обусловленных отправкой RTMP-пакетов актуализации, содержащих полные таблицы маршрутизации.

Конфигурирование маршрутизатора так, чтобы он посылал на конфигурируемый интерфейс только пакеты актуализации для глухих сетей, осуществляется с помощью команды appletalk rtmp-stub. Хотя в сети компании ZIP было решено эту функцию не использовать, ниже показан пример конфигурирования с помощью этой команды, если бы было решено воспользоваться этой функцией на интерфейсе Ethernet маршрутизатора в Сингапуре:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interface ethernet
Singapore(config-if)#appletalk rtmp-stub
Singapore(config-if)#^Z
```

Работу как протокола AppleTalk EIGRP, так и протокола RTMP можно проверить с помощью ранее рассмотренной команды show appletalk route. Для более детальной проверки конфигурации и работы протокола AppleTalk EIGRP могут быть использованы дополнительные команды ОС IOS режима EXEC, которые показаны в табл. 5.3.

Таблица 5.3. Команды ОС IOS режима EXEC для протокола AppleTalk EIGRP

Команда ОС IOS режима EXEC для протокола EIGRP	Функция
show appletalk eigrp interfaces	Выводит информацию об интерфейсах, сконфигурированных на работу с протоколом AppleTalk EIGRP
show appletalk eigrp neighbors	Выводит данные о соседях, выявленных протоколом AppleTalk EIGRP
show appletalk eigrp topology	Выводит таблицу топологии протокола AppleTalk EIGRP
show appletalk eigrp traffic	Выводит данные о количестве пакетов, отправленных и принятых процессом (процессами) протокола AppleTalk EIGRP

Конфигурирование фильтрации в протоколе AppleTalk с применением списков доступа

Средства фильтрации пакетов протокола AppleTalk в ОС IOS компании Cisco позволяют администратору сети, основываясь на различных критериях, ограничивать доступ к определенным ресурсам в AppleTalk-сети, включая отдельные серверы, принтеры, сегменты сети, диапазоны адресов и зоны целиком. Как и конфигурирование списков доступа для протокола TCP/IP, процесс конфигурирования фильтрации пакетов состоит из задания критериев фильтрации и наложения этих критериев на конкретные AppleTalk-интерфейсы.

Задание списков доступа

Списки доступа для протокола AppleTalk несколько более сложны, чем списки доступа для протокола TCP/IP. Частично это обусловлено наличием логических зон, которые могут охватывать несколько интерфейсов и номеров AppleTalk-сетей. Кроме того, протокол AppleTalk пользуется зарегистрированными протоколом NBP именами устройств для доступа рабочих станций и серверов к сетевым ресурсам. Как уже рассматривалось ранее, адреса *сеть*. *узел*, связанные с этими сетевыми ресурсами, могут изменяться со временем в результате

динамических переговоров об адресе узла устройства.

Из-за наличия таких условий не рекомендуется использовать возможности фильтрации протокола AppleTalk для построения фильтров, основанных на сетевых адресах. Попытка ограничить доступ к ресурсам в конкретной зоне путем ограничения доступа к конкретной сети или кабельному диапазону может оказаться чрезвычайно трудоемкой, если эта зона включает в себя несколько интерфейсов и несколько географических мест расположения. Кроме того, основанный на адресе сети или узла список доступа не сможет работать, если адрес этого ресурса динамически изменится. Неправильная конфигурация может позволить доступ туда, где он нежелателен, и ограничить доступ тем, кто в нем нуждается.

Вместо фильтрации на основе сетевых адресов протокола AppleTalk, рекомендуется использовать фильтры на основе имен служб протокола AppleTalk, которые зарегистрированы в протоколе NBP, и на базе запросов и распространения имен зон. Поскольку эта концепция имеет глубокую связь с работой протокола AppleTalk, то вполне логично управлять доступом на основе этих критериев. В последующей части данного раздела будет рассмотрена фильтрация на основе имен службы NBP и имен зон.

Все критерии фильтрации в протоколе AppleTalk реализуются через команду глобального конфигурирования ОС IOS access-list с использованием нумерованных списков доступа, номера которых лежат в диапазоне значений 600—699. В отличие от списков доступа протокола IP и списков доступа протокола межсетевого обмена пакетами (Internetwork Packet Exchange — IPX), порядок команд формирования списка доступа для протокола AppleTalk значения не имеет. Однако при проектировании списков доступа протокола AppleTalk следует не упускать из вида два важных критерия.

Во-первых, элементы списка доступа не должны перекрывать друг друга. Примером перекрытия может быть ситуация, при которой командой permit network разрешается доступ к некоторой сети, а затем командой deny network доступ к этой же сети запрещается. Если при конфигурировании возникает ситуация с перекрытием элементов, то последний введенный элемент списка записывается вместо предыдущего и удаляет его из списка доступа. В нашем случае оператор permit network удаляется из списка доступа, как только вводится оператор deny network.

Во-вторых, в протоколе AppleTalk критерии логической и сетевой фильтрации вводятся одним и тем же списком доступа, и оценка обоих типов критериев производится одновременно. Поэтому каждый список доступа должен всегда содержать метод для обработки пакетов с данными или пакетов с обновлениями маршрутной информации, которые не соответствуют ни одному из операторов управления доступом из списка доступа. Для того чтобы явно определить, какие пакеты данных или пакеты с обновлениями маршрутной информации как следует обрабатывать, в зависимости от обстоятельств используйте одну из следующих команд:

- команду глобального конфигурирования access-list other-access (при задании условий доступа к сетям и кабельным диапазонам);
- команду глобального конфигурирования access-list additional-zones (при задании условий доступа к зонам);
- команду глобального конфигурирования access-list other-nbps (при задании условий доступа к именованным сетевым ресурсам с использованием NBP-пакетов).

Эти команды могут ставиться в любом месте списка доступа. ОС IOS автоматически размещает команду access-list deny other-access в конце списка доступа. Она также размещает в конце списка команды access-list deny additional-zones и access-list deny other-nbps при задании условий доступа к зонам и NBP-именам, соответственно. Если явно не задать способ обработки пакетов с данными или пакетов обновления маршрутной информации, которые не удовлетворяют критериям операторов управления доступом, они автоматически пропущены не будут, а пакеты с данными будут уничтожены.

Для реализации фильтрации на основе имен сетевых ресурсов, зарегистрированных службой NBP, используйте в качестве параметра для нумерованного списка доступа протокола AppleTalk ключевое слово npb. Дополнительные ключевые слова позволяют осуществлять фильтрацию по типам объектов, именам объектов, по зонам, в которых размещаются объекты или по типам NBP-функции. В приведенном ниже примере задается

NBP-фильтр на маршрутизаторе в Сан-Хосе, который запрещает доступ ко всем серверам, находящимся в Сан-Хосе, кроме выделенного сервера общего пользования в техническом департаменте. Заданный список доступа разрешает доступ к именованному ресурсу (в данном случае это сервер с именем "Открытый сервер технического департамента" (Engineering Public)), типу объекта (AppleTalk-файл-сервер или AFPServer) и к зоне, в которой объект размещается (зона Сан-Хосе (San Jose Zone)). Опция deny other-nbps запрещает доступ ко всем другим именованным ресурсам.

```
San-Jose#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+2.
San-Jose(config)#access-list 601 permit nbp 1 object Engineering Publio
San-Jose(config)#access-list 601 permit nbp 1 type AFPServer
San-Jose(config)#access-list 601 permit nbp 1 zone San Jose Zone
San-Jose(config)#access-list 601 deny other-nbps
San-Jose(config)#^Z
```

Фильтрация по имени зоны позволяет ставить фильтр как на запросы об имени зоны, так и на распространение информации об именах зон. Как эти фильтры устанавливаются, будет показано в следующем разделе. Оба типа фильтрации по именам зон реализуются путем добавления к команде задания списка доступа протокола AppleTalk параметра в виде ключевого слова zone. В примере ниже задается фильтр по имени зоны на маршрутизаторе в Сингапуре, который запрещает доступ к зоне с именем Operations Zone (Производственная зона), одновременно разрешая доступ к остальным зонам посредством ключевого слова additional-zones:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#access-list 605 deny zone Operations    "
Singapore(config)#access-list 605 permit additional-zones
Singapore(config)#^Z
```

Наложение списков доступа

Задав критерии фильтрации списка доступа протокола AppleTalk, необходимо наложить их на один или несколько интерфейсов, чтобы могла выполняться фильтрация пакетов. Наложение списков доступа к интерфейсу может выполняться либо в выходящем направлении, либо во входящем. При входящем направлении перемещения пакетов они поступают в маршрутизатор через интерфейс. При выходящем направлении пакеты покидают маршрутизатор и поступают на интерфейс.

Списки доступа протокола AppleTalk, заданные как NBP-фильтры, накладываются с помощью субкоманды конфигурирования интерфейса ОС 1OS appletalk access-group. Параметром этой команды выступает ключевое слово in или out, при этом, если ключевое слово не указывается, по умолчанию подразумевается наличие слова out. В примере ниже выполняется наложение ранее заданного списка доступа протокола AppleTalk под номером 601 на интерфейсы глобальной сети маршрутизатора в Сан-Хосе, тем самым разрешается доступ из остальных частей сети компании ZIP только к открытому серверу технического департамента:

```
San-Jose#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
San-Jose(config)#interface serial 0/0
San-Jose(config-if)#appletalk access-group 601
San-Jose(config-if)#interface serial 1/0
San-Jose(config-if)#appletalk access-group 601
San-Jose(config-if)#^Z
```

Чтобы понять, как списки доступа протокола AppleTalk, заданные в качестве фильтров имен зон, накладываются на запросы имени зоны и на распространение имен зон, рассмотрим принципы управления такими именами.

На маршрутизаторе имена зон отображаются на номера сетей с помощью протокола обмена информацией о зонах (Zone Information Protocol — ZIP). Когда маршрутизатор принимает в свою таблицу маршрутизации объявление о новой сети, протокол ZIP заносит сеть в таблицу информации о зонах (Zone Information Table — ZIT) и посылает наружу широковещательный ZIP-запрос об информации относительно зон, которые отображаются на адрес новой сети. Подобным образом протокол ZIP может строить полный список всех зон, которые соответствуют адресам сетей, полученным из протоколов RTMP и EIGRP.

Первичными получателями ZIP-информации являются пользователи рабочих станций. Как только пользователь компьютера Macintosh производства компании Apple открывает окно выбора (Chooser), в сегмент локальной сети посылаются широковещательные пакеты запроса списка зон протокола ZIP. Ответить списком имеющихся зон может любой AppleTalk-маршрутизатор, стоящий в этом сегменте локальной сети.

Примечание

Не путайте протокол обмена информацией о зонах Zone Information Protocol (ZIP) со взятой в качестве примера сетью компании Zoom Integrated Products (ZIP).

Помня о роли протокола ZIP, рассмотрим применение фильтров имен зон протокола AppleTalk. Для фильтрации распространения имен зон от одного маршрутизатора к другому используется субкоманда конфигурирования интерфейса ОС IOS appletalk zip-reply-filter. Такой фильтр работает, заставляя маршрутизатор отвечать на ZIP-запросы об отображении сети на имя зоны только теми именами зон, которые разрешены в списке доступа. В результате, команда appletalk zip-reply-filter накладывается только на ответные пакеты, которые являются выходящими для конфигурируемого интерфейса. В представленном ниже примере интерфейс Ethernet расположенного в Сингапуре маршрутизатора конфигурируется заданным ранее списком доступа 605 протокола AppleTalk так, чтобы ни один неизвестный маршрутизатор не узнал о зоне с именем Operations Zone:

```
Singapore #configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interface ethernet 0
Singapore(config-if)#appletalk zip-reply-filter 605
Singapore(config-if)#^Z
```

Для того чтобы пользователи не узнали об определенных зонах, используется субкоманда конфигурирования интерфейса ОС IOS appletalk getzonelist-filter. Фильтр работает, заставляя маршрутизатор отвечать на ZIP-запросы списков зон только теми именами зон, которые разрешены списком доступа. Как и для команды appletalk zip-reply-filter, единственными фильтруемыми ответными пакетами являются выходящие пакеты интерфейса, который конфигурируется командой getzonelist-filter. В показанном ниже примере интерфейс Ethernet расположенного в Сингапуре маршрутизатора конфигурируется ранее заданным списком доступа 605 протокола AppleTalk так, чтобы ни одна рабочая станция конечного пользователя при просмотре наличных ресурсов в окне выбора не узнала о зоне с именем Operations Zone:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#interface ethernet 0
Singapore(config-if)#appletalk getzonelist-filter 605
Singapore(config-if)#^Z
```

Совет

Как упоминалось выше, если в сегменте сети располагается несколько маршрутизаторов, то на ZIP-запросы списков зон типа GetZoneList может отвечать любой из них. Исходя из этого факта, важно, чтобы фильтрация имен зон накладывалась на все маршрутизаторы, принадлежащие одному сегменту сети, идентичным образом. Невыполнение условия идентичности приводит к тому, что пользователям будут предоставляться различные списки зон в зависимости от того, какое устройство отвечает на запрос. Также несовместимая фильтрация может привести к ситуации, когда зоны появляются и исчезают на рабочей станции пользователя каждые несколько секунд. Учитывая потенциальную несовместимость, как правило, следует накладывать фильтры имен зон только тогда, когда все маршрутизаторы работают под управлением ОС IOS, если только маршрутизаторы, которые не используют ОС IOS, не обладают аналогичными возможностями фильтрации.

Просмотреть поведение списков доступа и верифицировать правильность их конфигурации можно с помощью команд ОС IOS режима EXEC show access-list и show appletalk access-list. Первая команда показывает все списки доступа, заданные на маршрутизаторе, а вторая — только списки доступа протокола AppleTalk. Параметром каждой команды может быть номер списка доступа. И команда будет выводить информацию о содержании только этого списка. Если параметр не указывается, то выводятся сведения обо всех списках. Ниже показан результат выполнения команды show appletalk access-list на сингапурском маршрутизаторе для рассмотренного ранее списка доступа:

```
Singapore#show appletalk access-lists
AppleTalk access list 605:
   deny zone Operations
   permit additional-zones
```

Команда ОС IOS режима EXEC show appletalk interface показывает, где и для выполнения какого типа фильтрации на интерфейсе накладывается список доступа протокола AppleTalk. Последние две строчки показанного ниже результата исполнения этой команды на маршрутизаторе в Сингапуре указывают, что список доступа протокола AppleTalk под номером 605 наложен как zip-reply-фильтр и как getzonelist-фильтр:

```
Singapore#show appletalk interface ethernet 0
EthernetO is up, line protocol is up
   AppleTalk cable range is 4001-4010
   AppleTalk address is 4008.30,Valid
   AppleTalk zone is "Manufacturing "
   AppleTalk address gleaning is disabled
   AppleTalk route cache is enabled
   AppleTalk GetZoneList filter is 605
   AppleTalk Zip Reply filter is 605
```

Конфигурирование основных служб удаленного доступа по коммутируемым каналам связи протокола AppleTalk

В этой главе рассматриваются возможности ОС IOS по маршрутизации с использованием протокола AppleTalk. ОС 1OS компании Cisco также позволяет осуществлять удаленный доступ AppleTalk-клиентов, что аналогично функциональным возможностям, описанным в предыдущей главе для работы на коммутируемых каналах связи протокола IP. Удаленный доступ по протоколу AppleTalk позволяет использовать службы протокола AppleTalk в

условиях, когда пользователи не имеют физического соединения с выделенным каналом сегмента локальной сети.

В рамках ОС IOS возможность удаленного доступа по протоколу AppleTalk реализуется по асинхронным коммутируемым линиям и в рамках ISDN-сети. В данной главе рассматриваются специфические команды протокола AppleTalk, обычно используемые для конфигурирования служб доступа клиентов к сети по асинхронным коммутируемым каналам связи посредством AppleTalk-протокола удаленного доступа (AppleTalk Remote Access Protocol — ARAP) и AppleTalk-протокола управления (AppleTalk Control Protocol — ATCP) из протокола двухточечной связи (Point-to-Point Protocol — PPP). AppleTalk-доступ в рамках стандарта ISDN обычно используется при маршрутизации по типу с запросом по вызову, но эта тема лежит за пределами предмета данной книга.

Как было показано в главе 4 в описании конфигурирования IP-служб удаленного доступа по коммутируемым каналам связи, удаленный доступ организуется путем настройки конфигурации асинхронной линии, что разрешает пользователям работу с AAA-службами, и конфигурирования специфичных для протокола опций. Для протокола AppleTalk конфигурирование асинхронной линии связи аналогично конфигурированию, показанному в главе 4 для протокола IP.

Только протокол ARAP требует дополнительных команд конфигурирования асинхронной линии. AppleTalk-клиенты, использующие протокол ARAP, требуют конфигурирования дополнительных AAA-служб, а пользователи, работающие с протоколом канального уровня PPP, используют конфигурацию AAA-служб, ранее рассмотренную для протокола IP и обсуждаемую в главе 7, "Основы администрирования и управления". И удаленные пользователи, работающие с протоколом ARAP, и удаленные пользователи, работающие с протоколом AppleTalk PPP, требуют наложения специфичных для протокола команд конфигурирования на групповой асинхронный интерфейс сервера доступа.

Рассмотрим дополнительные команды конфигурирования, необходимые для поддержки удаленных клиентов, работающих с протоколом ARAP. Для активации ARAP-служб удаленного доступа по коммутируемым каналам связи нужны команды конфигурирования асинхронной линии. Эти команды разрешают работу протокола ARAP, задают метод аутентификации в рамках этого протокола и определяют способ вызова протокола ARAP во время сеанса удаленного доступа.

Субкоманда конфигурирования линии ОС IOS arap enable является первой из этих трех команд, которая разрешает работу протокола ARAP на коммутируемых линиях. Субкоманда конфигурирования линии ОС IOS arap authentication default инструктирует сервер доступа использовать метод аутентификации протокола ARAP по умолчанию, сконфигурированный через службу AAA. И наконец, субкоманда конфигурирования линии ОС IOS autoselect arap конфигурирует в сервере доступа автоматическое распознавание попытки удаленного пользователя подключиться с использованием протокола ARAP. В примере ниже показано доконфигурирование сервера доступа Sing2511, на котором ранее были сконфигурированы IP-службы удаленного доступа, субкомандами конфигурирования линии протокола ARAP:

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Sing2511(config)#line 1 16
Sing2511(config-line)#arap enable
Sing2511(config-line)#arap authentication default
Sing2511(config-line)#autoselect arap
Sing2511(config-line)#^Z
```

Для верификации личности удаленных пользователей, обращающихся через протокол ARAP, требуются дополнительные команды AAA-аутентификации. Чтобы задать критерии идентификации ARAP-пользователей, используется команда глобального конфигурирования aaa authentication arap. В качестве параметра этой команды выступают название метода и список методов аутентификации. Как и для протокола PPP, аутентификация в рамках

протокола ARAP может выполняться с использованием локального имени пользователя или сервера аутентификации, например, с помощью системы управления доступом через контроллер доступа к терминалу TACACS+ (Terminal Access Controller Access Control System). Управление гостевыми регистрациями также может задаваться с помощью ключевого слова auth-guest, которое определяет, что гостевые регистрации в протоколе ARAP допускаются, только если пользователь во время сеанса удаленного доступа предварительно был освидетельствован для работы в режиме EXEC ОС IOS.

Удаленным пользователям, работающим с протоколом ARAP, также необходимо сообщить AppleTalk-номер сети и зоны, к которым они приписываются во время сеанса удаленного доступа по коммутируемой линии связи. Для задания ARAP-номера сети и имени зоны используется команда глобального конфигурирования ОС IOS arap network.

В приведенном ниже примере на находящемся в Сингапуре сервере доступа с именем Sing2511 конфигурируется метод выполнения аутентификации и вводится информация протокола AppleTalk, что в результате открывает удаленным ARAP-пользователям доступ к сети AppleTalk. ARAP-пользователи освидетельствуются для работы с базой данных локальных имен пользователей сервера доступа и приписываются к AppleTalk-сети 2500 в зоне с именем Mac-dailup Zone ("Мак-зона" с удаленным доступом):

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Sing2511(config)#aaa authentication arap default auth-guest local
Sing2511(config)#arap network 2500 Mac-dialup
Sing2511(config)#^Z
```

Для того чтобы разрешить удаленным пользователям доступ к службам сети AppleTalk *с* использованием протоколов ATCP и PPP, нужны всего две протокольные команды в дополнение к командам конфигурирования протокола PPP и линии, которые описывались ранее в главе 4 при обсуждении конфигурирования IP-служб удаленного доступа. Как и при работе протокола ARAP, AppleTalk PPP-клиенты должны иметь номер AppleTalk-сети и имя зоны, к которым они могут приписываться. Хотя имя зоны и номер сети протокола ARAP могут быть такими же, для создания PPP-номера сети и имени зоны удаленных пользователей используется отдельная команда ОС IOS.

После назначения PPP-номера сети и имени зоны для удаленных пользователей на групповом асинхронном интерфейсе активируются службы AppleTalk PPP-клиента. Для определения PPP-номера сети и имени зоны используется команда глобального конфигурирования ОС IOS appletalk virtual-net. Эти значения указываются в качестве параметра данной команды. Субкоманда конфигурирования интерфейса ОС IOS appletalk client-mode активирует службы удаленного доступа протокола PPP на интерфейсе, для которого она применяется. При активации режима клиента на интерфейсе отключается AppleTalk-маршрутизация, и пакеты актуализации маршрутной информации перестают отправляться. Ниже показан пример конфигурирования находящегося в Сингапуре сервера доступа с именем Sing2511 на поддержку удаленных AppleTalk PPP-клиентов, которые приписываются к AppleTalk-сети 2501 и зоне Zone Mac-dialup:

```
Sing2511#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands one per line.  End with CTRL+Z.
Sing2511(config)#apple virtual-net 2501 Mac-dialup
Sing2511(config)#interface group-as nc 1
Sing2511(config-if)#appletalk client-mode
Sing2511(config-if)#^Z
```

Верификация взаимодействия в сети с протоколом AppleTalk и устранение неполадок

ОС IOS обладает многочисленными инструментальными средствами для обнаружения причин возникновения проблем с установлением связи в сети AppleTalk, ошибок конфигурации сети и

проблем с протоколами динамической маршрутизации. В данном разделе рассматриваются команды ОС IOS режима EXEC show, отладочные команды debug и диагностические команды, которые облегчают идентификацию проблем в сети.

Команда ОС IOS режима EXEC show appletalk interface является полезным инструментом для идентификации ошибок конфигурирования номера сети и имени зоны, а также для контроля за процессом инициализации интерфейса. Вторая строка выводимой этой командой информации показывает текущий статус инициализации и также информирует о любых ошибках конфигурирования. Ниже показан пример ошибки конфигурирования, вызванной пропуском информации об имени зоны на интерфейсе Ethernet 0 маршрутизатора компании ZIP SF-2:э

```
SF-J#show appletalk interface ethernet 0
EthernetO is up line protocol is up
AppleTalk node down Port configuration error
AppleTalk able range is 151-200
AppleTalk address is 198.72 Invalid
AppleTalk zone is not set.
AppleTalk address gleaning is disabled
AppleTalk route  cache is enabled
```

Команда ОС IOS режима EXEC show appletalk nbp позволяет определить номер сети, связанный с конкретным именованным ресурсом. Ею выводится значение адреса *сеть.узел*, связанное с именем, зарегистрированным в службе NBP. Таким образом, администратор сети может проверить, имеет ли именованный ресурс ожидаемый номер *сеть.узел* или нет. Ниже показана выдержка из результата исполнения команды show appletalk nbp на маршрутизаторе компании ZIP SF-1, из которой видно, как его интерфейсы регистрируются службой NBP:

```
SF-l#show appletalk nbp
Net Adr Skt  Name                Type        Zone
2   12  254  SF-l.FddiO        ciscoRouter     SF Zone
22  7   254  SF-l.EthernetO    ciscoRouter     Operations
```

Чтобы решить проблему с установкой связи, следует знать, отвечает ли станция, к которой подключается интерфейс локальной сети. Для проверки способности маршрутизатора осуществлять преобразование сетевого AppleTalk-адреса в MAC-адрес используется команда ОС IOS режима EXEC show appletalk arp. Эта команда может воспринимать в качестве параметра конкретный AppleTalk-адрес *сеть.узел*. Если параметр не вводится, показываются все записи Apple Talk ARP-таблицы. Выводимые этой командой данные включают отображение AppleTalk-адреса на MAC-адрес, время записи в таблице и интерфейс, с которым связана запись в ARP-таблице. (Маршрутизатор по истечении четырех часов автоматически убирает ARP-запись из своей ARP-таблицы.) Ниже приведен пример исполнения команды show appletalk arp на маршрутизаторе компании ZIP SF-1:

```
SF-l#show appletalk arp
Address Age(min)   Type      Hardware Addr          Encap       Interface
2.12    -         Hardware    0000.OcOc.34dl.0000     SNAP         FddiO
9.159   -         Hardware    000.OcOc.23dl.0000      SNAP         Ethernetl
5.20    -         Dynamic     0000.030c.Ilc4.0000     SNAP         FddiO
```

Подобно протоколу TCP/IP, в протоколе AppleTalk реализуется протокол эхо-пакетов запрос/ответ, называемый эхо-протоколом AppleTalk (AppleTalk Echo Protocol — AEP). Протокол AEP позволяет AppleTalk-станции посылать эхо-запрос станции пункта назначения. Когда станция принимает запрос, она посылает станции-отправителю эхо-ответ. Этот простой протокол дает сетевому администратору возможность проверять достижимость AppleTalk-серверов, принтеров и других устройств. В ОС IOS протокол AEP реализуется в виде команды ping appletalk. В дополнение к основной информации о достижимости команда ping appletalk предоставляет также данные о приблизительном времени,

которое требуется эхо-запросу и ответу, чтобы достичь станции пункта назначения и вернуться оттуда. В следующем примере команда ОС IOS режима EXEC ping appletalk, выполняемая на маршрутизаторе SF-1, посылает пять AEP-запросов размером 100 байт по заданному AppleTalk-адресу:

```
SF-l#ping appletalk 5.20
Type escape sequence to abort.
Sending 5  100-byte AppleTalk Echos to 5.20  timeout is 2 seconds:
! ! ! ! !
Success rate is 100 percent (5/5)round-trip min/avg/max =  1/2/4  ms
```

Маршрутизатор посылает пять эхо-запросов AEP и восклицательными знаками (!) сообщает, что все ответы получены. Он также сообщает о количестве попыток отправки эхо-запросов и количестве принятых эхо-ответов. Затем он рассчитывает процент успешных пингов. Также рассчитывается минимальное, максимальное и среднее время реакции.

В табл. 5.4 показаны различные ответные символы, которые могут быть получены в результате исполнения команды ping в сети AppleTalk.

Таблица 5.4. Ответные символы команды ping

Символ	Значение
!	Каждый восклицательный знак указывает на получение ответа (эха) от устройства с заданным адресом
.	Каждая точка указывает, что сетевой сервер отключился по превышению времени, ожидая ответ от устройства с заданным адресом
B	Эхо-ответ, полученный от устройства с заданным адресом, плохой или имеет искаженную форму
C	Был принят ответ с плохой контрольной суммой
E	При передаче эхо-пакета по заданному адресу не удалось найти MAC-адрес
R	Передача пакета по заданному адресу потерпела неудачу из-за отсутствия маршрута к нему

Команда ping appletalk, как и ее IP-аналог, имеет привилегированную и непривилегированную версии. В пользовательском режиме EXEC непривилегированная версия позволяет пользователю только задавать AppleTalk-адрес. Привилегированная версия, доступная в полнофункциональном режиме EXEC, позволяет модифицировать параметры эхо-запросов, включая количество запросов, размер посылаемых пакетов, значение временного предела ожидания и многие другие величины. Ниже показан пример привилегированной версии команды ping appletalk, исполненной на маршрутизаторе SF-1; размер пакета запроса был увеличен до 500 байт:

```
SF-l#ping appletalk
Target AppleTalk address:   5.20
Repeat count   [5]:
Datagram size   [100]:   500
Timeout  in seconds   [2] :
Verbose   [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5 500-byte AppleTalk Echos to 5.20 timeout is 2 seconds:
! ! ! ! !
```

```
Success rate is 100 percent (5/5)  round-trip min/avg/max = 1/4/6 ms
```

Общую информацию о производительности и функционировании протокола AppleTalk на маршрутизаторе компании Cisco можно получить, воспользовавшись Двумя различными командами ОС IOS режима EXEC. Команда show appletalk traffic имеет в своем составе счетчики такой информации, как общее количество пакете в, посланных и принятых маршрутизатором, количество посланных и принятых широковещательных пакетов, а также она предоставляет статистические данные работы протоколов RTMP и EIGRP и сведения о том, посылал и получал ли маршрутизатор AppleTalk эхо-пакеты. Счетчики команды show appletalk traffic являются коммулятивными. Сброс их показаний может осуществляться привилегированной командой ОС IOS режима EXEC clear appletalk traffic либо путем выполнения перезагрузки или выключения-включения питания маршрутизатора. Ниже приведен пример информации, выводимой командой **show** appletalk traffic, выполненной на маршрутизаторе компании ZIP в Сингапуре:

```
Singapore#show appletalk traffic
AppleTalk statistics:
Rcvd: 90 total, 0 checksum errors, 0 bad hop count
      45 local destination, 0 access denied, 0 fast access denied
      0 for MacIP, 0 bad MacIP, 0 no client
      0 port disabled, 0 no listener
      0 ignored, 0 martians Beast: 0 received, 18766 sent
Sent: 18766 generated, 0 forwarded, 0 fast forwarded, 45 loopback
      0 forwarded from MacIP, 0 MacIP failures
      25 encapsulation failed, 0 no route, 0 no source
DDP:  135 long, 0 short, 0 macip, 0 bad size
NBP:  30 received, 0 invalid, 0 proxies
      0 replies sent, 55 forwards, 25 lookups, 0 failures
RTMP: 0 received, 0 requests, 0 invalid, 0 ignored
      17624 sent, 0 replies ATP:   0 received
ZIP:  0 received, 20 sent, 0 netinfo
Echo: 40 received, 0 discarded, 0 illegal
      20 generated, 20 replies sent
Responder:  0 received, 0 illegal, 0 unknown
      0 replies sent,0 failures
AARP: 0 requests, 0 replies, 0 probes
      0 martians, 0 bad encapsulation, 0 unknown
      153 sent, 0 failures, 0 delays, 25 drops
Lost:  0 no buffers
Unknown: 0 packets
Discarded: 0 wrong encapsulation, 0 bad SNAP discriminator
AURP:0 Open Requests, 0 Router Downs
      0 Routing Information sent, 0 Routing Information received
      0 Zone Information sent, 0 Zone Information received
      0 Get Zone Nets sent, 0 Get Zone Nets received
      0 Get Domain Zone List sent, 0 Get Domain Zone List received
      0 bad sequence
EIGRP: 0 received, 0 hellos, 0 updates, 0 replies, 0 queries, 1097 sent,
            0 hellos, 0 updates, 0 replies, 0 queries
```

Второй командой ОС IOS, которая обеспечивает общую информацию о работе протокола AppleTalk, является команда show appletalk globals. Эта команда позволяет получать данные об установках различных конфигурационных опций протоколов динамической маршрутизации, количестве сетевых маршрутов и зон в сети AppleTalk, а также она выводит информацию о том, как будут обрабатываться пакеты с ошибками при их поступлении на маршрутизатор. Данная команда полезна, если необходимо удостовериться, что желаемые опции сконфигурированы правильно и работают на маршрутизаторах ожидаемым образом. В примере ниже показан результат исполнения команды show appletalk globals на маршрутизаторе компании ZIP SF-1:

```
SF-l#show appletalk globals
AppleTalk global information:
Internet is incompatible with older, AT Phasel,  routers.
There are 16 routes in the internet.
There are 11 zones defined.
Logging of significant AppleTalk events is disabled.
ZIP resends queries every 10 seconds.
RTMP updates are sent every 10 seconds.
RTMP entries are considered BAD after 20 seconds.
RTMP entries are discarded after 60 seconds.
AARP probe retransmit count: 10, interval: 200 msec.
AARP request retransmit count: 5, interval: 1000 msec.
DDP datagrams will be checksummed.
RTMP datagrams will be strictly checked.
RTMP routes may not be propagated without zones.
Routes will be distributed between routing protocols.
Routing between local devices on an interface will not be performed.
IPTalk uses the udp base port of 768 (Default).
EIGRP router id is: 2500
EIGRP maximum active time is 3 minutes
Alternate node address format will not be displayed.
Access control of any networks of a zone hides the zone.
```

Кроме представленных в данной главе команд, связанных с операциями по поиску и устранению неполадок и верификации, в режиме EXEC ОС IOS существует еще множество привилегированных отладочных команд debug, призванных помочь определить качество работы протокола AppleTalk на маршрутизаторе. Эти команды debug позволяют получать как общую, так и подробную диагностическую информацию, которая может помочь в устранении неисправностей, а также при проверке работы маршрутизатора, протоколов маршрутизации и других функций. Например, команда debug appletalk errors позволяет изолировать ошибки в конфигурировании сетевых адресов и имен зон на интерфейсах маршрутизатора. Самые распространенные команды debug для протокола AppleTalk приведены в табл. 5.5.

Таблица 5.5. Отладочные команды для протокола AppleTalk

Команда	Описание
`debug appletalk arp`	Показывает данные о сгенерированных AARP-запросах и посланных маршрутизатору ответах, а также данные о деятельности модуля управления возрастом записей таблицы адресов протокола AARP (AARP ager)
`debug appletalk eigrp-packet`	Показывает содержание пакетов протокола AppleTalk EIGRP, отправленных и принятых маршрутизатором
`debug appletalk eigrp-update`	Показывает деятельность протокола AppleTalk EIGRP по актуализации маршрутной информации маршрутизатора
`debug appletalk errors`	Выдает информацию об ошибках, возникающих при работе протокола AppleTalk
`debug appletalk events`	Демонстрирует важные события протокола AppleTalk, которые имели место на маршрутизаторе
`debug appletalk nbp`	Показывает деятельность службы NBP на маршрутизаторе
`debug appletalk packet`	Выдает AppleTalk-адреса отправителей и получателей пакетов, коммутируемых маршрутизатором. Как и команда debug ip packet, эта команда может вызвать перегрузку маршрутизатора, поэтому при ее применении следует проявлять осторожность. Предполагается, что отладка на уровне пакетов ограничивается конкретным интерфейсом
`debug appletalk routing`	Выводит данные об изменениях, имевших место в таблице маршрутизации в результате добавления и удаления маршрутов для протоколов EIGRP и RTMP

`debug appletalk rtmp`	Выводит данные об изменениях, имевших место в таблице маршрутизации в результате добавления и удаления только RTMP-маршрутов
`debug appletalk zip`	Показывает данные о деятельности на маршрутизаторе протокола обмена информацией о зонах ZIP

Резюме

В данной главе на примере сети компании ZIP рассмотрена базовая конфигурация, необходимая для работы с протоколом AppleTalk. Хотя эти основные команды и функции могут обеспечить работоспособность AppleTalk-сети, существует много других более совершенных технических функций, применение которых способно значительно улучшить работу и масштабируемость сети AppleTalk. Монографии и Web-ресурсы, приведенные в разделе "Дополнительная литература", являются прекрасными учебными материалами для понимания, реализации и устранения ошибок в работе этих функций. Ключевыми моментами в данной главе являются следующие положения.

- Система адресации протокола AppleTalk фазы 1 предусматривает использование только одного номера сети для идентификации сетевого сегмента. Система адресации фазы *2* предусматривает для идентификации одной или нескольких сетей либо одного номера сети, либо непрерывной последовательности номеров в форме *начало-конец*..
- Нерасширенный адрес фазы *2* может поддерживать только один сетевой адрес, тогда как расширенный адрес фазы *2* способен поддерживать несколько номеров сетей.
- Номера узлов назначаются динамически протоколом AARP.
- Номера локальных сетей и зоны могут конфигурироваться вручную или с помощью концепции маршрутизатора посева и режима поиска.
- Протоколами динамической маршрутизации, предлагаемыми ОС IOS компании Cisco для работы с протоколом AppleTalk, являются протоколы RTMP и EIGRP. Протокол RTMP конфигурируется по умолчанию, если разрешена маршрутизация по протоколу AppleTalk командой `appletalk routing`. Протокол EIGRP используется для снижения потребления полосы пропускания каналов локальной сети.
- Из-за гибкой и динамической природы адресов протокола AppleTalk использование их в списках доступа в качестве основы для фильтрации не рекомендуется. Фильтровать надо имена служб, зарегистрированных службой NBP, и запросы на рассылку имен зон.
- В протоколе AppleTalk команды формирования списка доступа не зависят от порядка их следования, как в протоколах IP и IPX. Если команды конфликтуют или перекрываются, то стоящая раньше команда отбрасывается, и реализуется команда, стоящая позднее.
- Каждый список доступа должен включать метод обработки пакетов данных и пакетов актуализации маршрутной информации, которые не соответствуют ни одному оператору управления доступом списка доступа. В противном случае такие пакеты данных и пакеты актуализации автоматически не пропускаются или уничтожаются.
- Для проверки конфигураций и устранения неполадок в сети AppleTalk существуют разнообразные команды show, debug и ping (см. табл. 5.3, 5.5 и 5.6).

Таблица 5.6. Сводная таблица команд режима EXEC для протокола AppleTalk

Команда	Функция
clear appletelk route	Выполняет очистку всей таблицы маршрутизации или, если задан, удаляет конкретный маршрут
clear appletalk traffic	обнуляет счетчики команды show appletalk traffic
ping appletalk *сеть.узел*	Осуществляет проверку с целью определения достижимости и способности отвечать указанного AppleTalk-адреса
show appletalk access-list	Показывает все списки доступа протокола AppleTalk или со-

	держание конкретного списка, если он задан
show appletalk globals	Дает общую информацию о конфигурации и работе протокола AppleTalk
show appletalk interface brief	Показывает краткие сводные данные об AppleTalk-адресах и статусе интерфейса для всех имеющихся на устройстве интерфейсов
show appletalk interface *интерфейс*	Показывает все параметры, связанные с AppleTalk-конфигурацией интерфейса
show appletalk nbp	Показывает номер сети, связанный с конкретным именованным ресурсом
show appletalk neighbors	Приводит список соседей в сети AppleTalk, информация о которых поступила в процессе динамической маршрутизации
show appletalk route *сетевой адрес*	Показывает таблицу маршрутизации целиком или, если он задан, конкретный маршрут
show appletalk static	Показывает сконфигурированные статические маршруты
show appletalk traffic	Выводит общие статистические данные о работе протокола AppleTalk на маршрутизаторе
show appletalk zone	Показывает список всех AppleTalk-зон, известных маршрутизатору

Таблица 5.7. Сводная таблица команд конфигурирования для протокола AppleTaik

aaa authentication arap *списочный метод*	Определяет, что протокол ARAP должен аутентифицироваться с помощью списочного AAA-метода
access-list	Создает нумерованный список доступа и связанные с ним критерии фильтрации
access-list номер [permit|deny] additional zones	Запрещает или разрешает прохождение пакетов данных и пакетов актуализации маршрутной информации, которые не удовлетворяют ни одному оператору управления доступом на основе имен зон из списка доступа
access-list номер [permit I deny] other-access	Запрещает или разрешает прохождение пакетов данных и пакетов актуализации маршрутной информации, которые не удовлетворяют ни одному оператору управления доступом на основе номеров сетей или кабельных диапазонов из списка доступа
access-list номер [permit I deny] other-nbps	Запрещает или разрешает прохождение пакетов данных и пакетов актуализации маршрутной информации, которые не удовлетворяют ни одному оператору управления доступом на основе имен службы NBP
appletalk access-group *номер*	Накладывает указанный список доступа на задачу фильтрации входящих и выходящих пакетов на интерфейсе
appletalk address *сеть.узел*	Задает номер сети в системе адресации фазы 1 интерфейсу локальной или глобальной сети
appletalk cable-range *начало-конец*	Задает кабельный диапазон в системе адресации фазы 2 интерфейсу локальной или глобальной сети
appletalk client-mode	Разрешает работу AppleTaik PPP-служб удаленного доступа на интерфейсе, в отношении которого используется
appletalk discovery	Реализует динамическое конфигурирование сетевого адреса и имени (имен) зоны (зон) в режиме поиска
appletalk getzonelist-filter *номер*	Накладывает указанный список доступа на задачу фильтрации выходящих ответов на запросы списка зон протокола ZIP, которые отсылаются рабочим станциям и серверам

appletalk maximum-paths *номер*	Разрешает маршрутизатору хранение в таблице маршрутизации протокола AppleTaik заданного количества маршрутов равной стоимости
appletalk protocol [eigrp|rtmp]	Определяет, какой протокол маршрутизации (EIGRP или RTMP) следует использовать на конкретном интерфейсе в AppleTalk-сетях, где разрешена работа протокола EIGRP
appletalk route-redistribution	Включает режим редистрибуции маршрутов между протоколами EIGRP и RTMP
appletalk routing	Разрешает на маршрутизаторе выполнение маршрутизации по протоколу AppleTaik
appletalk routing eigrp *автономная-система*	Разрешает на маршрутизаторе работу протокола Appletalk EIGRP
appletalk rtmp-stub	Конфигурирует маршрутизатор на отправку только укороченных пакетов актуализации маршрутной информации с того интерфейса, для которого применяется
appletalk static	Конфигурирует статический AppleTalk-маршрут
apple talk virtual-net	Устанавливает PPP-номер сети и имя зоны для удаленных пользователей, работающих по коммутируемым каналам связи
appletalk zip-reply-filter *номер*	Накладывает указанный список доступа на задачу фильтрации, выходящих ответов с ZIP-именем зоны, отсылаемых другим маршрутизаторам
appletalk zone *имя*	Конфигурирует на интерфейсе имя зоны
arap authentication default	Задает выполнение аутентификации в рамках протокола ARAP до разрешения сетевым службам на начало работы. Между сервером доступа и удаленным клиентом используется протокол аутентификации по умолчанию
arap enable	Разрешает работу протокола ARAP на асинхронных линиях
arap network	Задает ARAP-номер сети и имя зоны для удаленных пользователей
autoselect arap	Задает автообнаружение протокола ARAP на асинхронных линиях, переведенных в интерактивный режим
dialer map appletalk	Отображает AppleTalk-адрес сеть.узел на имя системы и телефонный номер для ISDN-вызовов соединения
frame-relay map appletalk	Статически отображает AppleTalk-адрес сеть узел на DLCI-идентификатор протокола Frame Relay
map-group	Присваивает интерфейсу названную групповую карту отображения для использования на нем при отображении AppleTalk-адре-сов на адреса канального уровня протокола ATM
map-list	Создает именованный список карт отображения для конфигурирования отображения AppleTalk-адресов на постоянные или коммутируемые виртуальные каналы системы адресации протокола ATM
x25 map appletalk	Статически отображает AppleTalk-адрес на адрес протокола X121

Дополнительная литература

1. Cisco Systems. *Troubleshooting Internetworking Systems: AppleTalk Connectivity.*
(В данном руководстве описываются способы устранения неполадок для множества протоколов и сетевых технологий. Эту книгу можно получить непосредственно в компании Cisco, а также в интерактивной версии по адресу

www.cisco.com/univercd/cc/td/doc/cisintwk/tis_doc/76523.htm.
2. Sidhu, G., R. Andrews, and A. Oppenheimer. *Inside AppleTalk*, 2nd Edition. Reading, Massachusetts: Addison-Wesley, 1990.
3. Vandersluis, K. and A. Eissa. *Troubleshooting Macintosh Networks: A Comprehencive Guide to Troubleshooting and Debugging Macintosh Networks.* Indianapolis, Indiana: IDG Books Worldwide, 1993.

Глава 6

Ключевые темы этой главы

Система адресации и структура адреса в протоколе IPX Основы структуры адресов протокола IPX и структура сети.

Конфигурирование IPX-адресов. Обзор схемы адресации протокола IPX, а также примеры конфигурирования адресов для интерфейсов глобальных и локальных сетей различных типов

Конфигурирование IPX-маршрутизации. Основы конфигурирования маршрутизации по протоколу IPX с использованием статических маршрутов и верификация IPX-маршрутизации.

Конфигурирование протоколов маршрутизации, работающих с протоколом IPX. Характеристики протоколов динамической маршрутизации IPX RIP и NLSP и примеры базовых конфигураций.

Конфигурирование фильтрации в протоколе IPX с применением списков доступа. Управление доступом в сети и защита информации с помощью команд access-list и ipx access-group.

Конфигурирование основных служб удаленного доступа по коммутируемым каналам связи протокола IPX. Установки для организации взаимодействия с IPX-клиентом по асинхронным коммутируемым каналам связи.

Верификация взаимодействия в сети с протоколом IPX и устранение неполадок Идентификация проблем связи с помощью команд show, ping и debug.

Конфигурирование переадресации IPX-пакетов типа 20. Опции конфигурирования ОС JOS для организации переадресации IPX-пакетов типа 20.

Основы IPX

В конце 1970-х годов компания Xerox создала сетевой протокол под названием XNS (Xerox Network Systems — Сетевые системы Xerox), который широко использовался различными поставщиками сетевых решений, включая компанию Novell, Inc. В начале 1980-х компания Novell внесла в этот протокол несколько изменений, переименовав его в протокол межсетевого обмена пакетами (Internet Packet Exchange Protocol — IPX), который ввела в качестве составной части своего продукта NetWare. Из группы протоколов XNS происходит и протокол транспортного уровня из состава продукта NetWare — протокол последовательного обмена пакетами (Streams Packet Exchange — SPX) Продукт Novell NetWare работает поверх группы протоколов IPX/SPX, равно как и поверх группы протоколов TCP/IP.

Продукт NetWare представляет собой группу протоколов для коллективного использования ресурсов (главным образом принт- и файл-серверов) рабочими станциями за счет реализации клиент-серверной модели работы. Компания Novell позиционирует NetWare в качестве сетевой операционной системы, поскольку она обеспечивает конечным пользователям доступ к ресурсам, находящимся в локальной или глобальной сети Продукт NetWare, являясь доминирующей сетевой операционной системой, широко используется во многих сетевых комплексах.

На рис. 6.1 показано несколько протоколов, которые наиболее часто используются в составе группы протоколов NetWare. В данной главе не будут рассматриваться все эти протоколы. Основное внимание будет уделено описанию протоколов сетевого и транспортного уровней, а именно: протоколу IPX, IPX-протоколу обмена информацией о маршрутизации (IPX Routing Information Protocol — IPX RIP), протоколу сервиса связи (NetWare Link State Protocol — NLSP), протоколу объявлений услуг (Service Advertisement Protocol — SAP) и протоколу SPX. В показанных на рис. 6.1 протоколах упоминаются и другие технологии межсетевого взаимодействия, с которыми читатель, вероятно, знаком.

Примечание

he все выпуски ОС IOS компании Cisco поддерживают протокол IPX. Следует удостовериться, что версия ОС IOS, исполняемая на вашем маршрутизаторе, поддерживает набор протоколов для настольных систем *Desktop Protocol Suite.*

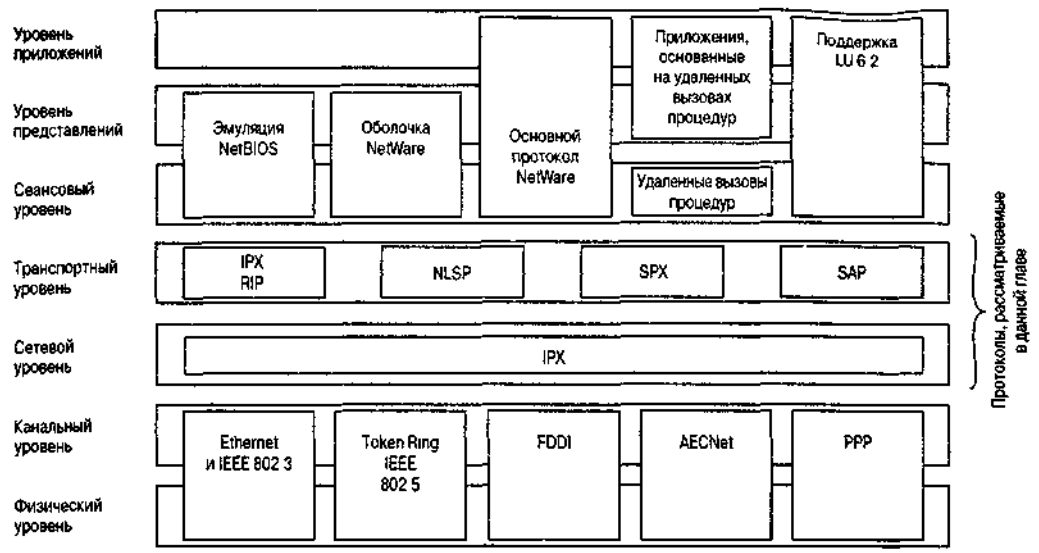


Рис. 6 1. Протоколы группы протоколов IPX

Система адресации и структура адреса в протоколе IPX

Протокол IPX представляет собой протокол сетевого уровня со своей собственной структурой адресации. В этом разделе рассматривается структура адресов протокола IPX, которой должен отвечать адрес каждого IPX-клиента (иногда в документации на ОС NetWare называемого рабочей

станцией) или сервера, чтобы те имели возможность обмениваться данными с другими IPX-устройствами, находящимися в сетевом комплексе.

IPX-адрес имеет две составляющие: 32-разрядную *сетевую* составляющую, которая используется для данного сегмента локальной или глобальной сети, и 48-разрядную *узловую* составляющую, которая уникальным образом идентифицирует клиента или сервер. Большинство узлов определяют этот уникальный номер, считывая 48-разрядный адрес канального уровня (уровня 2 модели OSI) на своем интерфейсе локальной сети. Как мы увидим, протокол IPX не требует, чтобы канальный адрес устройства совпадал с узловой составляющей, но, как правило, эти числа совпадают.

Выражаемые вместе в виде пары чисел *сеть.узел,* эти две составляющие записываются в шестнадцатеричной форме. Двухуровневая иерархия структуры IPX-адреса делает такую схему адресации масштабируемой для сетевых систем, но все же не такой масштабируемой, как структура IP-адресов с многоуровневой иерархией.

Администратор сети назначает номер сегменту IPX-сети точно так же, как выбирает IP-подсеть для данного сегмента локальной или глобальной сети. Все IPX-клиенты, IPX-серверы и маршрутизаторы компании Cisco, находящиеся в одном сегменте локальной или глобальной сети, должны иметь одинаковый номер сети.

Серверы NetWare имеют внутренние IPX-номера сети, которые отличаются от IPX-номеров сети любого интерфейса локальной или глобальной сети. Этот внутренний IPX-номер сети используется в качестве номера сети источника для служб ОС NetWare на сервере. Процесс объявления сетевых услуг будет рассмотрен в этой главе позднее при обсуждении протокола SAP В конфигурацию маршрутизатора компании Cisco внутренний IPX-номер сети может быть внесен с помощью команды глобального конфигурирования ipx internal-network. Концепция внутреннего номера сети обсуждается в разделе "Конфигурирование протокола NLSP".

Необходимо, чтобы каждый IPX-сервер или клиент имели в сегменте локальной или глобальной сети свой уникальный номер узла. Обычно IPX-клиенты извлекают этот номер, считывая 48-разрядный адрес канального уровня собственных интерфейсов локальной сети и затем используя его в качестве своего уникального адреса узла уже на сетевом уровне. Хотя адрес интерфейса локальной сети канального уровня совпадает с IPX-адресом узла, не следует делать вывод, что клиент использует эти два адреса одинаковым образом. Канальный адрес используется в процессе инкапсуляции на канальном уровне, например, в пакетах протоколов Ethernet или TokenRing. Адрес IPX-узла является второй частью IPX-адреса *сеть.узел* сетевого уровня для данного клиента. IPX-клиент в сети 10 с адресом интерфейса Ethernet канального уровня 0802.044d.d88f будет известен в IPX-сети под номером 10. 0802.044d.d88f как раз благодаря считыванию 48-разрядного канального адреса интерфейса локальной сети.

Использование канального адреса для определения уникального 48-разрядного адреса IPX-узла вовсе не является требованием протокола IPX. Можно иметь адрес узла, который не совпадает с канальным адресом, лишь бы он был уникальным для данного сегмента сети. Например, как мы уже видели, работающее под управлением ОС IOS устройство может иметь несколько интерфейсов локальной сети. После того как активируется функция IPX-маршрутизации, такое устройство выбирает канальный адрес первого интерфейса локальной сети в качестве уникального адреса узла для всех сегментов IPX-сети. Теперь представим себе, что канальный адрес интерфейса Ethernet 0 маршрутизатора имеет вид 0000.0cll.12ab. Если Ethernet 0 является первым интерфейсом локальной сети на данном маршрутизаторе, и маршрутизатор подключен к IPX-сети 10 и IPX-сети 20, то в IPX-сети 10 он будет виден как устройство с адресом 10.0000.Ocll.12ab, а в IPX-сети 20 — как устройство с адресом 20.0000.Ocll.12ab.

На устройстве, работающем под управлением ОС IOS, активирует IPX-маршрутизацию (и протокол IPX RIP, который будет рассматриваться ниже) команда глобального конфигурирования ОС IOS ipx routing. Когда эта команда вводится в конфигурацию, устройство автоматически выбирает номер IPX-узла на основе данных своего первого интерфейса локальной сети. В примере ниже активируется функция IPX-маршрутизации на маршрутизаторе компании ZIP SF-2:

```
SF-2#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands one per line.  End with CTRL+Z.
```

```
SF-2(config)#ipx routing
SF-2 (config)#^Z
```

Примечание

Если устройство не имеет интерфейса локальной сети, необходимо сконфигурировать уникальный адрес IPX-узла, введя его в качестве параметра команды ipx routing. Адрес узла должен иметь 12 десятичных разрядов и быть уникальным номером для IPX-сетей, подключенных к маршрутизатору.

Использование адреса канального уровня для определения адреса IPX-узла упрощает работу администратора сети, так как в этом случае IPX-клиенты не нуждаются в ручном конфигурировании. Кроме того, такое отображение адреса канального уровня на адрес сетевого уровня может исключить потребность в отдельном протоколе, например ARP, для отображения адресов между этими двумя уровнями. Данный вопрос обсуждался в главе 4, "Основы TCP/IP".

Конфигурирование IPX-адресов

В данном разделе рассматривается задача конфигурирования IPX-адресов на интерфейсах локальной и глобальной сети маршрутизаторов компании Cisco. Также здесь рассматривается конфигурирование используемых в среде протокола IPX четырех методов инкапсуляции на интерфейсах локальной сети.

Конфигурирование интерфейсов локальной сети

Все маршрутизаторы компании Cisco, которые осуществляют маршрутизацию по протоколу IPX, обладают уникальным IPX-адресом *сеть.узел* в каждом из подключенных к ним сегментов локальной сети. Этот адрес позволяет маршрутизатору знать, какие сети подключены к каким интерфейсам, и куда следует посылать пакеты для этих сетей.

Назначение сетевого IPX-адреса интерфейсам локальной и глобальной сети выполняется с помощью интерфейсной субкоманды ОС IOS ipx network. Адрес IPX-узла, как уже упоминалось ранее в этой главе, устанавливается командой глобального конфигурирования ipx routing. В приведенном ниже примере осуществляется конфигурирование IPX-адресов на каждом из трех интерфейсов локальной сети маршрутизатора SF-2:

```
SF-2#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands one per line.  End with CTRL+Z.
SF-2(config)#interface  ethernet 0
SF-2(config-if)#ipx network 200
SF-2(config-if)interface ethernet 1
SF-2(config-if)#ipx network 150
SF-2(config-if)#interface fddi 0
SF-2(config-if)#ipx network 10
SF-2(config-if )#^Z
```

Инкапсуляция на интерфейсе локальной сети в протоколе IPX

Как показано из рис. 6.1, протокол IPX работает поверх различных протоколов канального уровня. Первоначально протокол IPX был создан для работы над протоколом Ethernet. Затем по мере изобретения новых протоколов канального уровня, например, IEEE 802.3, IEEE 802.5 и FDDI, он был доработан для поддержки инкапсуляции и этих протоколов канального уровня. В результате, различные версии продукта NetWare поддерживают различные протоколы канального уровня и связанные с ними методы инкапсуляции. В IPX для большинства новых технологий локальных сетей используется единый протокол канального уровня, но он содержит четыре метода инкапсуляции для сегментов с локальной сетью Ethernet.

Методы инкапсуляции пакетов локальных сетей имеют разные названия в протоколе IPX и

ОС IOS компании Cisco. В табл. 6.1 приведено отображение названий типов кадров протокола IPX на синтаксис обозначения методов инкапсуляции в ОС IOS.

Таблица 6.1. Терминология инкапсуляции в протоколе IPX и синтаксис обозначения методов инкапсуляции в ОС IOS

Тип кадра в протоколе IPX	Название метода инкапсуляции в ОС IOS компании Cisco
Ethernet_802 2	sap
Ethernet_802.3	novell-ether
Ethernet_II	arpa
Ethernet_Snap	snap
Token-Ring	sap
Token-Ring_Snap	snap
Fddi_Snap	snap
Fddi_802 2	sap
Fddi Raw	novell-fddi

Примечание

По умолчанию на маршрутизаторах компании Cisco для всех интерфейсов Ethernet типом инкапсуляции ОС IOS является novell-ether. Соответственно, для интерфейсов Token Ring по умолчанию используется метод инкапсуляции sap, а для интерфейсов FDDI — snap.

Метод инкапсуляции ОС IOS sap по умолчанию используется ОС NetWare вереи 4.0. На интерфейсах Ethernet кадр этого типа использует стандартный заголовок протокола IEEE 802.3, за которым следует заголовок протокола управления логической связью (LLC) стандарта IEEE 802.2 (также известный под названием *точки доступа сервису,* или SAP). Заголовок протокола IEEE 802.2 LLC обеспечивает на уровне канала средство для определения протокола сетевого уровня в кадре протокола канального уровня. На интерфейсах Token Ring инкапсуляция по методу sap, являющемуся в данном случае методом инкапсуляции по умолчанию, предусматривает использована стандартного заголовка протокола IEEE 802.5, за которым следует заголовок протоколе IEEE 802.2 LLC. Аналогично, на интерфейсах FDDI этот тип кадра включает стандартный заголовок протокола FDDI, за которым следует заголовок протокола 1EE 802.2 LLC.

Примечание

Не путайте метод инкапсуляции IEEE 802.2 LLC SAP (где SAP обозначает точку доступа к сервису) с протоколом SAP (протокол объявления служб) ОС NetWare. Протокол NetWare SAP рассматривается в этой главе ниже.

Метод инкапсуляции novell-ether в ОС IOS компании Cisco аналогичен инкапсуляции по методу Ethernet_802.3 компании Novell и работает только на интерфейсах Ethernet. Тип кадра novell-ether предусматривает наличие стандартного заголовка протокола ШЕЕ 802.3, за которым стоит заголовок протокола IPX с полем контрольной суммы, установленным в шестнадцатеричное значение FFFF. Метод инкапсуляции novell-ether используется по умолчанию в ОС NetWare версии 3.11 ив ОС IOS на маршрутизаторах компании Cisco.

Для интерфейсов Ethernet, которые должны обрабатывать трафик протоколов TCP/IP и IPX, следует использовать метод инкапсуляции Novell Ethernet_II (в ОС IOS компании Cisco он называется агра). Этот метод использует просто заголовок

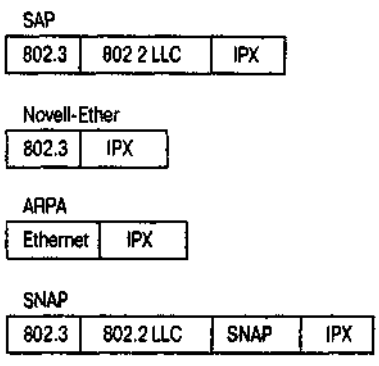


Рис. 6.2. Форматы инкапсуляции для протоколов канального уровня протокола IPX

протокола Ethernet, за которым стоит заголовок протокола IPX.

Метод инкапсуляции snap в ОС IOS на интерфейсе Ethernet использует стандартный заголовок протокола IEEE 802.3, за которым следует заголовок протокола IEEE 802.2 SNAP LLC. Протокол доступа к подсетям SNAP является стандартным методом инкапсуляции дейтаграмм сетевого уровня в IEEE-протоколах. На интерфейсах Token Ring и FDDI кадр SNAP-типа содержит стандартный заголовок протокола IEEE 802.5 или FDDI, за которым стоит заголовок IEEE 802.2 SNAP LLC.

На интерфейсах FDDI инкапсуляция по методу novell-fddi в обозначении ОС IOS компании Cisco соответствует инкапсуляции по методу Fddi_Raw в обозначении компании Novell. Этот тип кадра содержит стандартный заголовок протокола FDDI, за которым следует заголовок протокола IPX с полем контрольной суммы, установленным в шестнадцатеричное значение FFFF.

Резюмируем сказанное выше: на интерфейсах Ethernet возможны четыре метода инкапсуляции (sap, arpa, novell-ether и snap), три — на интерфейсе FDDI (sap, snap и novell-fddi) и два — на интерфейсе Token Ring (sap и snap). На рис. 6.2 показаны четыре схемы инкапсуляции для интерфейса Ethernet.

Примечание

Хотя в протоколе IPX существует несколько методов инкапсуляции канального уровня, очень часто необходимый метод определяет тот выпуск ОС NetWare, который исполняется в сети (NetWare 3.11 или NetWare 4.0). Для нормального взаимодействия в рамках данного сегмента локальной IPX-сети все устройства должны использовать один метод инкапсуляции. Это касается NetWare-клиентов, NetWare-серверов и маршрутизаторов компании Cisco.

Конфигурирование инкапсуляции

Для конфигурирования метода инкапсуляции на интерфейсе локальной сети используется команда ipx network с опцией encapsulation. В примере ниже показано конфигурирование метода инкапсуляции snap на интерфейсе Ethernet 0 маршрутизатора сети компании ZIP SF-2:

```
SF-2#configure terminal
Configuring from terminal memory or network [terminal]?
Enter configuration commands  one per line.  End with CTRL+Z.
SF-2(config)#interface ethernet 0
SF-2(config-if)#ipx network 200 encapsulation snap
SF-2(config-if)#^Z
```

В некоторых ситуациях бывает необходимым работать с несколькими методами инкапсуляции канального уровня ОС NetWare на одном интерфейсе локальной сети и в одно время. Например, может оказаться необходимым перевести некоторых IPX-клиентов с ОС NetWare 3.11 на ОС NetWare 4.0, использующих различные методы инкапсуляции канального уровня. Обычно различные методы инкапсуляции, используемые клиентом и сервером, не позволяют клиентам общаться с серверами, которые работают под управлением другой версии ОС NetWare. Однако благодаря применению в одном сегменте локальной IPX-сети двух различных методов инкапсуляции маршрутизатор компании Cisco позволяет осуществлять взаимодействие между клиентами и серверами, исполняющими различные версии ОС NetWare.

При одновременной работе с несколькими методами инкапсуляции следует назначить на интерфейсе маршрутизатора уникальные номера сети для каждого метода инкапсуляции канального уровня. В этом случае одна сеть становится первичной IPX-сетью, а вторая — вторичной IPX-сетью. При этом обе они приписаны к одному физическому интерфейсу. Для назначения вторичных сетей на интерфейсе локальной сети, работающем *с* различными методами инкапсуляции, используется команда ipx network с опцией secondary. В примере ниже на интерфейсе Ethernet 0 маршрутизатора сети компании ZIP SF-2 назначается инкапсуляция по методу АКРА:

```
SF-2#configure
Configuring from terminal memory or network [terminal]?
```

```
Enter onfiguration commands one per line.  End with CTRL+Z.
SF-2(config)#interface ethernet 0
SF-2(config-if)#ipx network 201 encapsulation arpa secondary
SF-2(config-if)#^Z
```

Конфигурирование интерфейсов глобальной сети

Адресация в глобальной сети для протокола IPX, которая аналогична адресации в локальной сети, конфигурируется с помощью субкоманды конфигурирования интерфейса ipx network. В данном разделе рассматривается назначение IPX-номеров сети двухточечным и многоточечным интерфейсам глобальной сети. В главе 3, "Основы интерфейсов устройств Cisco", отмечалось, что для работы на интерфейсе глобальной сети специальные методы инкапсуляции (например, X.25 или Frame Relay) должны конфигурироваться в явном виде. Это же относится и к методам инкапсуляции для глобальной сети, используемым протоколом IPX.

Адресация двухточечных интерфейсов глобальной сети

Как говорилось в главе 4 при рассмотрении протокола IP, двухточечный интерфейс глобальной сети соединяет только два устройства. Для маршрутизации IPX-пакетов через двухточечный интерфейс глобальной сети взаимодействующие интерфейсы обоих маршрутизаторов должны быть сконфигурированы на один IPX-номер сети. Как и при использовании интерфейса локальной сети, каждое устройство должно иметь на интерфейсе глобальной сети уникальный IPX-номер узла.

IPX-номер сети на двухточечном интерфейсе глобальной сети может конфигурироваться с помощью субкоманды конфигурирования интерфейса ipx network. Ниже приведен пример назначения IPX-номера сети двухточечным интерфейсам глобальной сети (два подынтерфейса Frame Relay и один HDLC-интерфейс) маршрутизатора Seoul- 1:

```
Seoul–l#configure
Configuring from terminal memory or network [terminal]?
Enter  configuration commands  one per line.  End with CTRL+Z.
Seoul-1 (config)#interface serial 0.16 point-to-point
Seoul-1 (config-if ) #ipx network 2901
Seoul-1 (config-if ) #interface serial 0.17 point-to-point
Seoul-1 (config-if) #ipx network 2902
Seoul-1 (config-if ) #interface serial 1
Seoul-1 (config-if ) #ipx network 1901
Seoul-1 (config-if) #^Z
```

Адресация многоточечных интерфейсов глобальной сети

Вопросы адресации многоточечных интерфейсов глобальной сети рассматривались в главе 4 на примере протокола IP. Как и протокол IP, протокол IPX может использоваться со многими различными многоточечными интерфейсами глобальной сети, включая Frame Relay, X.25, ISDN и ATM. Каждый из этих многоточечных интерфейсов может конфигурироваться на маршрутизацию IPX-пакетов с помощью субкоманды конфигурирования интерфейса ipx network. А отображение адресов конкретного канального уровня на IPX- номер сети конфигурируется по-разному для каждого протокола глобальной сети.

При использовании многоточечных интерфейсов Frame Relay маршрутизатору необходимо отображать DLCI- идентификаторы на IPX-номер *сетъ.узел.* Такое отображение может динамически выполнять функция обратного разрешения адресов Inverse ARP протокола Frame Relay. В качестве альтернативы используется субкоманда конфигурирования интерфейса frame-relay map ipx и статическое отображение DLCI-адреса протокола Frame Relay на IPX-номер *сеть. узел,* достижимый через многоточечный интерфейс глобальной сети.

Адресация многоточечных интерфейсов глобальной сети X.25 подобна адресации интерфейсов Frame Relay в том, что в обоих случаях для статического отображения используются субкоманды конфигурирования интерфейса. Интерфейсы X.25 должны иметь отображения своих IPX-адресов на адреса протокола X. 121, которые используются для

настройки виртуальных каналов между системами. Каждый виртуальный канал идентифицируется адресом протокола X. 121, используемым для установления соединения. Чтобы выполнить статическое отображение IPX-адреса на адрес протокола X.121, на многоточечном интерфейсе глобальной сети следует использовать субкоманду конфигурирования интерфейса x25 map ipx.

Адресация многоточечных интерфейсов ISDN тоже требует применения команд статического отображения. Однако для интерфейсов ISDN команды отображения нужны только тогда, когда устройство хочет установить соединение по вызову с другим устройством. Для отображения IPX-адресов на имена систем и телефонные номера, которые используются для соединения по вызову через интерфейс ISDN, применяется субкоманда конфигурирования интерфейса OC IOS dialer map ipx.

Отображение между канальными адресами протокола ATM, каковыми являются идентификаторы виртуальных путей/виртуальных каналов (VPI/VCI-адреса), и IPX-номером сети на многоточечном интерфейсе ATM зависит от типа используемых ATM-протоколов и виртуальных каналов В протоколе IPX можно использовать LLC/SNAP инкапсуляцию ATM-пакетов как для постоянных, так и для коммутируемых виртуальных каналов. При использовании метода постоянных виртуальных каналов в ATM-сети устанавливается постоянный виртуальный канал, и пакеты идентифицируются как адресованные на IPX-адрес на другом конце конкретного виртуального канала. При использовании метода коммутируемых виртуальных каналов IPX-пакеты идентифицируются как направленные на статически заданный канальный ATM-адрес. Когда маршрутизатор запрашивает соединение с ATM-адресом для конкретного IPX-адреса, ATM-коммутатор устанавливает по требованию виртуальный канал.

Если LLC/SNAP-инкапсуляция используется при работе с постоянными виртуальными каналами, то субкоманда конфигурирования интерфейса OC IOS map-group команда глобального конфигурирования OC IOS map-list используются для отображения IPX-адресов на конкретный постоянный виртуальный канал. Если LLC/SNAI инкапсуляция используется при работе с коммутируемыми виртуальными каналами, то субкоманда конфигурирования интерфейса OC IOS map-group и команда глобальной конфигурирования OC IOS map-list используются для отображения IPX-адресов и адреса точек доступа к сетевым службам (NSAP — network service access point), которые в свою очередь, используются для идентификации удаленных устройств в ATM-сети.

Проверка конфигурации IPX-адресов

Верификация IPX-адресов и других IPX-атрибутов, назначенных интерфейсам, может выполняться с помощью команды режима EXEC show ipx interface. Эта команда обеспечивает получение полной картины всех параметров, связанных с конфигурацией протокола IPX на всех интерфейсах. Если в качестве параметра данной команды вводится название конкретного интерфейса, то выдается только та информация, которая относится к этому интерфейсу. Ниже показан результат исполнения команды show i] interface ethernet 0 на маршрутизаторе сети компании ZIP SF-2:

```
SF-2#show ipx interface ethernet 0
EthernetO is up line protocol is up
  IPX address is 200.0000.OcOc.llbb 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 1010
  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 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
RIP interpacket delay is 55 ms
IPX accounting is disabled
IPX fast switching is  configured (enabled)
RIP packets received 6
RIP packets sent 1861
SAP packets received 330
SAP packets sent 4
```

В первой строке результата показан административный и рабочий статус интерфейса. Вторая строка показывает IPX-адрес *сеть.узел* и метод инкапсуляции в протоколе IPX. В выводимом результате также показывается статус различных IPX-фильтров и списков доступа, некоторые из них рассматриваются в этой главе.

Команда ОС IOS режима EXEC show ipx interface имеет опцию, которая позволяет увидеть краткий обзор информации об IPX-адресах и статусах всех имеющихся в устройстве интерфейсов. Такая резюмирующая версия результата может быть получена с помощью команды show ipx interface brief. Ниже приводится результат исполнения команды show ipx interface brief на маршрутизаторе компании ZIP SF-2:

```
SF-2#show ipx interface brief
Interface    IPX Network         Encapsulation         Status      IPX State
Ethernet0     200                NOVELL-ETHER          up          [up]
Ethernet1     150                NOVELL-ETHER          up          [up]
Fddi0         0                  SNAP                  up          [up]
Loopbackl    unassigned    not  config'd  up          n/a
```

В приведенном выше результате показан IPX-номер сети, назначенный каждому интерфейсу, название метода инкапсуляции и рабочий статус для каждого интерфейса.

В дополнение к проверке конфигурации протокола IPX на самом интерфейсе можно также просматривать статические и динамические отображения IPX-адресов на адреса канального уровня в различных средах многоточечных глобальных сетей. Для этого используются команды ОС IOS режима EXEC show frame-relay map, show atm map, show x25 map и show dialer maps.

Конфигурирование IPX-маршрутизации

Назначение IPX-адресов *сеть.узел* работающим под управлением ОС IOS устройствам и интерфейсам является необходимым условием для прокладки маршрутов IPX-пакетам. Вторым жизненно важным условием является процесс IPX-маршрутизации. Для обеспечения полнофункционального общения, как и в IP-сетях, маршрутизаторы должны осуществлять IPX-маршрутизацию и иметь маршруты к IPX-сетям сетевых комплексов. Чтобы определять, где находятся IPX-сети, маршрутизаторы используют таблицу маршрутизации, которая создается алгоритмами, называемыми протоколами маршрутизации.

В рамках протокола IPX протоколы маршрутизации могут быть либо статическими, либо динамическими. При использовании статических протоколов конфигурирование таблицы IPX-маршрутизации информацией о сетевых путях осуществляется вручную. Динамические же протоколы полагаются на маршрутизаторы, которые сами объявляют информацию о различных IPX-сетях, к которым они подсоединены. Протокол IPX работает с тремя протоколами динамической маршрутизации, которые рассматриваются в разделе "Конфигурирование протоколов маршрутизации, работающих с протоколом IPX".

Команды конфигурирования IPX-маршрутизации

Как упоминалось в этой главе ранее, активация маршрутизации по протоколу 1P> осуществляется с помощью команды глобального конфигурирования ipx routing После активации обработки по протоколу IPX маршрутизатор строит таблицу, используемую в процессе маршрутизации. По умолчанию после того, как интерфейс локальной или глобальной сети конфигурируется IPX-адресом, и он переводится в рабочее состояние, сетевой IPX-адрес этого интерфейса помещается в таблицу маршрутизации. В таблицу маршрутизации помещаются данные по всем активным интерфейсам, подключенным к маршрутизатору. Если в сети находится только один маршрутизатор, то он обладает информацией обо всех подключенных к нему IPX сетях, и нет необходимости в конфигурировании статической или динамически маршрутизации. Только когда сеть содержит несколько маршрутизаторов, необходимы записи в таблицу статической или динамической маршрутизации.

Чтобы просмотреть таблицу IPX-маршрутизации, можно воспользоваться командой ОС IOS режима EXEC show ipx route. Если ввести эту команду без указания параметров, она покажет содержание всей таблицы IPX-маршрутизации. В пример ниже представлен результат для маршрутизатора сети компании ZIP SF-2, который имеет только подключенные и находящиеся в активном состоянии интерфейсы, и дополнительных записей в таблице маршрутизации нет:

```
SF-2#show ipx route
Codes: C - Connected primary network c - Connected  secondary network
       S - Static F - Floating static L - Local (internal)  W - IPXWAN R - RIP E -
       EIGRP N - NLSP X - External A - Aggregate s - seconds u - uses U - Per-user
       static
       3 Total IPX routes. Up to 1 parallel paths and 16 hops allowed.
No default route known.
C        10  (NOVELL-FDDI)   FdO
C       150 (NOVELL-ETHER)   Etl
C       200 (NOVELL-ETHER)   EtO
```

Команда show ipx route предоставляет администратору сети полезные данные является ключевым инструментом для определения путей, которыми проходят IP" пакеты по сети. По своей форме результат исполнения этой команды подобен результату исполнения рассмотренной в главе 4 команды show ip route, которая показывает содержание таблицы IP-маршрутизации.

Первый раздел выводимого результата представляет собой легенду первого столб таблицы. Он говорит о том, откуда были получены данные о маршруте. Каждая трех последних строк в этой таблице IPX-маршрутизации показывает один маршрут до IPX-сети, способ получения данных о нем, метод инкапсуляции пакетов локальной сети в протоколе IPX и название интерфейса, связанного с маршрутом. Буква "C' первом столбце свидетельствует о том, что все три маршрута известны из активных подключенных первичных IPX-сетей. Команда show ipx route будет рассмотрен; разделе "Проверка конфигурации IPX-маршрутизации".

Конфигурирование статической маршрутизации

В главе 4 рассматривались различные причины использования статических] маршрутов. То же относится и к статическим IPX-маршрутам. Для конфигурировав статических IPX-маршрутов в таблице IPX-маршрутизации можно использовать команду глобального конфигурирования ipx route.

Проверка конфигурации IPX-маршрутизации

Как отмечалось ранее, командой для проверки конфигурации IPX-маршрутизации является команда ОС IOS режима EXEC show ipx route. В данном разделе будут рассмотрены и другие команды, которые помогают в верификации и управлении конфигурацией таблицы IPX-маршрутизации.

Команда show ip route является инструментом, который используется для просмотра состояния таблицы IPX-маршрутизации. Сконфигурированы ли статические маршруты или работают протоколы динамической маршрутизации, эта команда показывает, есть ли в маршрутизаторе те маршруты, которые были сконфигурированы или, как ожидается, будут

определены в процессе обучения. Протоколы динамической IPX-маршрутизации рассматриваются в следующем разделе. Ниже приводится выдержка из результата исполнения команды show ip route на маршрутизаторе компании ZIP SF-2:

```
SF-2#show ipx route
Codes:  C - Connected primary network   - Connected secondary network
        S - Static  F - Floating static L - Local (internal)  W - IPXWAN R - RIP
        E - EIGRP N - NLSP X - External A - Aggregate, s -seconds u - uses
        4 Total IPX routes. Up to 1 parallel paths and 16 hops allowed.
No default route known.
C         10  (NOVELL-FDDI)   FdO
C        150 (NOVELL-ETHER)   Etl
C        200 (NOVELL-ETHER)   EtO
R 100 [02/01] via 100.0000.Ic2c.23bb, 19s, FdO
```

Из показанного выше видно, что есть маршруты к IPX-сетям, непосредственно подключенным к маршрутизатору SF-2, и маршрут к IPX-сети 100, данные о котором были динамически получены через протокол IPX RIP от маршрутизатора SF-1.

Точно так же, как при использовании команды show ip route, задав номер сети, с помощью команды show ipx route можно посмотреть конкретный маршрут, используя привилегированную команду режима EXEC clear ipx route, можно удалять IPX-маршруты из таблицы маршрутизации. При выполнении отладочных работ можно воспользоваться этой командой, чтобы удалить маршрут, а затем с помощью команды show ipx route проверить, обучится ли маршрутизатор этому маршруту.

Конфигурирование протоколов маршрутизации, работающих с протоколом IPX

В главе 4 уже обсуждались моменты, которые следует учитывать при выборе протокола динамической маршрутизации.

- Топология сети.
- Формирование сводных адресов и маршрутов.
- Скорость сходимости.
- Критерии отбора маршрутов.
- Масштабируемость (расширяемость).
- Легкость внедрения.
- Средства защиты.

ОС IOS позволяет работать с несколькими протоколами динамической IPX-маршрутизации. Выбирая оптимальный протокол для сети, следует учитывать приведенные выше критерии.

Однако, прежде чем углубиться в изучение отдельных протоколов динамической маршрутизации, следует рассмотреть протокол SAP, который является динамическим протоколом служб и неразрывно связан с протоколами динамической IPX-маршрутизации. После протокола SAP будут рассмотрены протоколы динамической IPX-маршрутизации IPX RIP, NLSP и IPX EIGRP.

Протокол SAP

Протокол объявлений услуг (Service Advertisement Protocol — SAP) является закрытым протоколом компании Novell, который осуществляет объявление сервисов ОС NetWare в IPX-сети. Сервис представляет собой ресурс, например, сервис по работе с файлами или услуги печати, которыми может захотеть воспользоваться IPX-клиент. Всем услугам присвоен тип, который выражается шестнадцатеричным числом. Некоторые типы заданы компанией Novell, тогда как номера других присвоены поставщиками этих услуг для ОС NetWare. Например, SAP тип 4 является стандартным типом сервиса работы с файлами ОС NetWare, SAP тип 7 — стандартный тип сервиса для принтеров.

По умолчанию серверы, работающие с ОС NetWare, делают широковещательную рассылку SAP-пакетов каждые 60 секунд, объявляя об известных услугах. Каждый работающий с ОС NetWare сервер узнает о SAP-службах во многом так же, как узнает информацию протоколов динамической маршрутизации и затем строит табличное представление этой информации, называемое SAP-таблицей.

Маршрутизаторы компании Cisco по умолчанию разрешают работу протокола SAP для всех интерфейсов, сконфигурированных под протокол IPX. SAP-таблицу маршрутизатор строит на основе информации протокола SAP, получаемой от NetWare-серверов и других маршрутизаторов. Для просмотра SAP-таблицы маршрутизатора компании Cisco используется команда ОС IOS режима EXEC show ipx servers. В примере ниже показан результат исполнения команды show ipx servers на маршрутизаторе компании ZIP SF-1:

```
SF-l#show ipx servers
Codes: S - Static, P - Periodic, E - EIGRP, N - NLSP, H - Holddown, + = detail 2 Total
IPX Servers
Table ordering is based on routing andserver info
Type Name     Net Address       PortRoute Hops  Itf
P    4 SF-MAIN 100.0001.0002.0006:04512/01  1     EtO
P    4 SF-ENG  100.0809.0001.0002:04512/01  1     EtO
```

Этот результат говорит о том, что маршрутизатор SF-1 узнал о двух IPX-серверах, каждый из которых предлагает услугу работы с файлами (о чем свидетельствует цифра 4 в первой части имени сервера). Для каждого IPX-сервиса показывается IPX-адрес сервера, предлагающего сервис, метрику IPX-маршрута сервиса и интерфейс, на котором маршрутизатор слышит сервис. В этом примере оба сервиса идентифицированы как периодические, что означает получение информации о них через протокол SAP (который объявляет об услугах через регулярные периодические интервалы времени). Другие методы получения информации об IPX-сервисе будут рассмотрены в последующем материале данной главы в разделах "Конфигурирование протокола NLSP" и "Конфигурирование протокола IPX EIGRP".

Точно так же, как возможны статические IPX-маршруты, существуют и статические записи в SAP-таблице. Статические записи в SAP-таблице задаются с помощью команды глобального конфигурирования ipx sap. Эти записи полезны в сетевых средах, которые используют коммутируемый или резервный коммутируемый канал.

Когда сервер или маршрутизатор сформируют SAP-таблицу, они смогут отвечать NetWare-клиентам, которые нуждаются в тех или иных услугах. Такие клиенты посылают IPX-сообщения с запросом о ближайшем сервере (так называемые сообщения IPX Get Neareast Server — GNS), пытаясь найти сервер, который может предоставить необходимые им услуги. NetWare-серверы, которые обладают такой информацией, могут ответить клиенту, давая конкретный IPX-адрес. Маршрутизаторы компании Cisco тоже могут отвечать клиентам IPX-адресом сервиса, если этот сервис прописан в SAP-таблице маршрутизатора. Если маршрутизатор компании Cisco слышит GNS-сообщение в том сегменте локальной сети, в котором, как известно, сервис существует, то он на GNS-сообщение не отвечает.

Примечание

Ближайший сервер — это сервер, который предоставляет сервис и в SAP-таблице имеет кратчайший маршрут. Если критерию удовлетворяют несколько серверов, то маршрутизатор компании Cisco отвечает адресом сервера, которого он слышал последним. Это может привести к тому, что несколько NetWare-клиентов получат в ответе на свой GNS-запрос адрес одного и того же сервера. Подобная ситуация не выглядит оптимальной, если в IPX-сети имеется несколько серверов, предоставляющих одинаковый сервис в целях балансировки нагрузки от запросов клиентов.

В этом случае следует воспользоваться командой глобального конфигурирования ipx gns-round-robin, которая заставляет маршрутизатор при ответах на GNS-запросы циклически двигаться по списку подходящих серверов. В следующем разделе представлена информация о фильтрации GNS-ответов, посылаемых маршрутизатором через конкретные

Фильтры сообщений протокола SAP

ОС IOS компании Cisco позволяет администратору сети организовать фильтрацию на основе того, какие SAP-услуги устройство объявляет из своей SAP-таблицы или заносит в нее. Такая фильтрация сообщений протокола SAP широко используется в сетевых комплексах для ограничения объема входящего и исходящего трафика протокола SAP на маршрутизаторе.

Во многих IPX-сетях фильтры сообщений протокола SAP используются для снижения количества SAP-сообщений, посылаемых через интерфейсы глобальной сети, чем уменьшается нагрузка по трафику. Фильтрация же принимаемых объявлений протокола SAP может снизить количество IPX-услуг, которое маршрутизатор держит в своей оперативной памяти, и обеспечить некоторую степень защиты сети. Ограниченная защита сети достигается за счет того, что устройству, работающему под управлением ОС IOS, не разрешается предоставлять данные из SAP-таблицы по тем услугам, которые хотят оставаться "припрятанными" в IPX-сети. Дополнительная информация о фильтрации IPX-пакетов приводится в этой главе в разделе "Конфигурирование фильтрации в протоколе IPX с применением списков доступа".

Чтобы создать SAP-фильтры по IPX-адресам или типу SAP-сервиса, можно использовать команду глобального конфигурирования access-list. Фильтры сообщений протокола SAP используют списки доступа с номерами от 1000 по 1099. Аналогично спискам доступа в протоколах IP и Apple Talk, эти списки доступа позволяют использовать *подстановочную* или *безразличную* маски. Эта возможность позволяет одной командой глобального конфигурирования access-list представить сразу несколько IPX-адресов.

В приведенном ниже примере на маршрутизаторе компании ZIP в Сан-Хосе создается SAP-фильтр, который разрешает объявление услуг, предоставляемых единственным NetWare-сервером с адресом 10. 0000.0000. a0b0:

```
San-Jose#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands one per line.  End with CTRL+Z.
San-Jose(config)#access-list 1000 permitlO.0000.0000.aObO
San-Jose(config)#access-list 1000 deny -1
San-Jose(config)#^Z
```

Примечание

При построении фильтров сообщений протокола SAP (и списков доступа протокола IPX для фильтрации пакетов, что рассматривается далее в этой главе) номер IPX-сети -1 обозначает все IPX-сети. Таким образом, в приведенном выше примере вторая строчка списка доступа 1000 запрещает все SAP-сообщения. Аналогично спискам доступа протокола IP, в списках доступа протокола IPX последняя строка с оператором запрещения подразумевается всегда. Показанное здесь ее явное присутствие имеет целью проиллюстрировать применение номера IPX-сети -1.

После конфигурирования фильтра сообщений протокола SAP его необходимо наложить на определенный интерфейс устройства, работающего под управлением O< IOS. С помощью субкоманд конфигурирования интерфейса ipx input-sap-filte и ipx output-sap-filter можно поинтерфейсно осуществлять фильтрацию сообщений протокола SAP, которые принимаются или посылаются устройством, соответственно. Наложим SAP-фильтр, использующий список доступа 1000, на все исходящие объявления протокола SAP на интерфейсе Serial 0 маршрутизатора в Сан-Хосе:

```
San-Jose#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
San-Jose(config)#interface serial 0
San-Jose(config-if)#ipx output-sap-filter 1000
```

```
San-Jose(config)#^Z
```

В качестве другого примера можно построить SAP-фильтр, который на интерфейсе глобальной сети разрешает объявления только об услугах работы с файлами и печати ото всех серверов. В примере ниже строится SAP-фильтр, разрешающий только работу с файлами (тип 4) и печать (тип 7). Этот фильтр накладывается на исходящие объявления на интерфейсе Serial 0 маршрутизатора сети компании ZIP в Сан-Хосе:

```
San-Jose#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
San-Jose(config)#access-list 1005 permit -1 4
San-Jose(config)#access-list 1005 permit -1 7
San-Jose(config)#interface serial 0
San-Jose(config-if)#ipx output-sap-filter 1005
San-Jose(config-if)#^Z
```

Еще один тип SAP-фильтра разрешает или запрещает услуги ОС NetWare на основе IPX-адреса маршрутизатора. Одно из применений фильтров такого типа состоит в сокрытии всех услуг, источником которых выступает заданный маршрутизатор. Накладывает подобный SAP-фильтр маршрутизатора на заданный интерфейс команда конфигурирования интерфейса ipx router-sap-filter. В приведенном ниже примере маршрутизаторный SAP-фильтр накладывается на интерфейс FDDI O/O маршрутизатора сети компании ZIP SF-Core-1, чтобы скрыть все услуги ОС NetWare сервера технического департамента:

```
SF-Core-l# configure
Configuring from terminal,   memory,   or network   [terminal]?
Enter configuration commands,   one per line.   End with CTRL+Z.
SF-Core-1(config)#access-list 1001 permit aa.0207.0104.0874
SF-Core-1(config)#interface fddi  0/0
SF-Core-1(config-if)#ipx router-sap-filter 1001
SF-Core-1(config-if)#^Z
```

ОС IOS также позволяет на поинтерфейсной основе осуществлять фильтрацию услуг из SAP-таблицы, годящихся в качестве отклика на GNS-запросы, посылаемые NetWare-клиентами. GNS-фильтры на выходе интерфейса используются, когда надо не допустить идентификации конкретных серверов в качестве ближайших, или когда необходимо, чтобы все GNS-запросы обрабатывались конкретным сервером. В примере, приведенном ниже, для маршрутизатора SF-Core-1 задается список доступа протокола IPX, который разрешает упоминание в ответах на GNS-запросы только одного NetWare-сервера. Этот список доступа накладывается как выходной GNS-фильтр на интерфейсе FDDI O/O маршрутизатора SF-Core-1 с помощью субкоманды конфигурирования интерфейса ipx output-gns-filter:

```
SF-Core-1#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
SF-Core-1(config)#access-list 1010 permit aa.0207.0104.0874
SF-Core-1(config)#interface fddi 0/0
SF-Core-1(config-if)#ipx output-gns-filter 1010
SF-Core-1(config-if)#^Z
```

Конфигурирование протокола IPX RIP

IPX RIP является протоколом динамической маршрутизации ОС NetWare, аналогичным по функции протоколу IP RIP. Этот протокол использует метод вектора расстояния, образует и управляет таблицами IPX-маршрутизации между 1PX-маршрутизаторами и NetWare-серверами. В главе 4 уже обсуждались протокол IP RIP и свойства протоколов маршрутизации на основе метода вектора расстояния. Протокол IPX RIP относится к классу протоколов внутренних

шлюзов (IGP). Для протокола IPX не существует протоколов внешних шлюзов (EGP), поскольку ОС NetWare работает всегда только во внутренних сетях предприятия и никогда — в сетях общего пользования типа Internet. Работа протокола IPX RIP разрешается на всех интерфейсах по умолчанию со вводом команды глобального конфигурирования ipx routing.

Данный протокол был первым протоколом динамической маршрутизации для IPX-сетей, и поэтому в нем нет таких развитых технических возможностей современных протоколов динамической маршрутизации, как сведение адресов и маршрутов, скорость сходимости, критерии выбора маршрута и масштабируемость. Как будет видно далее из материала данного раздела, некоторые из этих вопросов решаются протоколами NLSP и IPX EIGRP, являющимися более современными протоколами динамической маршрутизации для протокола IPX.

Протокол IP RIP в качестве метрики маршрутизации использует счет переходов, а н протоколе IPX RIP для принятия решений о выборе маршрута применяется другая метрика, известная под названием *тактов системных часов*. Такт системных часов эквивалентен одной восьмой секунды. Метрика пункта назначения в тактах системных часов измеряется путем проверки полосы пропускания на интерфейсе, необходимой для достижения этого пункта назначения. В результате исполнения команды show ip route на маршрутизаторе SF-2 маршрут до IPX-сети 100 имеет метрику в два такта системных часов и один переход, что отражено в таблице IPX-маршрутизации в виде записи [02/01]:

```
SF-2#show ipx route
Codes:   C - Connected primary network, c - Connected secondary
           network, S -Static, F - Floating static, L - Local (internal), W -IPXWAN,
           R - RIP, E - EIGRP, N - NLSP, X - External, A -Aggregate, s - seconds, u
           - uses
4 Total IPX routes. Up to 1 parallel paths and 16 hops allowed.
No default route known.
C          10  (NOVELL-FDDI),  FdO
C         150 (NOVELL-ETHER),  Etl
C         200 (NOVELL-ETHER),  EtO
R 100 [02/01] via 100.0000.Ic2c..23bb, 19s, FdO
```

Если количество тактов системных часов до достижения пункта назначения равно для нескольких маршрутов, присутствующих в таблице маршрутизации протокола IP) RIP, то, чтобы разорвать этот узел, маршрутизатор использует тот маршрут, у которого наименьшее количество переходов между маршрутизаторами. Как и протокол IP RIF протокол IPX RIP имеет по умолчанию максимальное количество переходов — 16. Подобно всем маршрутным протоколам, обслуживаемым ОС IOS, если таблица маршрутазации протокола IPX RIP содержит пути равной стоимости (если такты системных часов и межмаршрутизаторные переходы увязаны), то маршрутизатор распределяет нагрузку трафика к пункту назначения между всеми имеющимися такими путями.

Примечание

По умолчанию маршрутизатор, использующий ОС IOS, не обучается нескольким параллельным IPX-путям равной стоимости до конкретного пункта назначения. Маршрутизатор воспринимает один путь до пункта назначения и отбрасывает всю информацию об альтернативных параллельных путях равной стоимости, о чем и свидетельствует фраза из результата исполнения команды show ipx route: "Up to 1 parallel
paths and 16 hops allowed" ("Допускается не более 1 параллельного пути и 16 переходов"). Такое поведение по умолчанию основано на реализации некоторых клиентов и услуг ОС NetWare, которые не могут обрабатывать IPX-пакеты, поступающие неупорядоченным образом. А это может иметь место, когда нагрузка распределяется между параллельными путями равной стоимости.

Чтобы разрешить маршрутизатору помещать в таблицу IPX-маршрутизации пути равной стоимости, используется команда глобального конфигурирования ipx maximum-paths.

Например, команда ipx maximum-paths 2 позволяет маршрутизатору знать о двух путях равной стоимости до данного пункта назначения. Количество путей равной стоимости, разрешаемое маршрутизатору, зависит от топологии IPX-сети.

По умолчанию маршрутизаторы компании Cisco распределяют нагрузку на попакетной основе между всеми параллельными путями равной стоимости до IPX-адреса пункта назначения. Однако может оказаться необходимым, чтобы все пакеты для каждого уникального IPX-адреса пункта назначения проходили по одному и тому же пути, даже если существует несколько путей равной стоимости. Для активации такой функции используется команда глобального конфигурирования ОС IDS ipx per-host-load-share.

Конфигурирование протокола NLSP

Протокол NLSP представляет собой протокол внутренних шлюзов на основе метода учета состояния канала для IPX-сетей. Этот протокол, базирующийся на протоколе "промежуточная система — промежуточная система" (IS-IS), обладает техническими характеристиками, аналогичными характеристикам других протоколов с учетом состояния канала, например OSPF. Подобно другим протоколам этого вида, он тоже поддерживает иерархическую адресацию и быструю сходимость.

Для агрегации и сведения номеров IPX-сетей протокол NLSP использует технику иерархической маршрутизации. Агрегация и сведение маршрутов полезны в больших IPX-сетях по тем же причинам, по которым они полезны в больших IP-сетях.

Первым уровнем маршрутизации по протоколу NLSP считается *область*. Областью в протоколе NLSP называется логическая группа IPX-адресов сетей; концептуально она схожа с областью в протоколе OSPF, которая представляет собой группу IP-сетей и подсетей. Таким образом, *NLSP-маршрутизация уровня 1* существует в области. NLSP-обмен между областями называется *маршрутизацией уровня 2*. Все области с NLSP-маршрутизаторами, обменивающиеся данными в рамках маршрутизации уровня 2, могут быть объединены в иерархическую группу, называемую *доменом маршрутизации*. NLSP-обмен между доменами маршрутизации называется *маршрутизацией уровня 3*. На рис. 6.3 показана сетевая система, использующая протокол NLSP.

Протокол NLSP требует, чтобы на маршрутизаторе был сконфигурирован внутренний номер IPX-сети. Сделать это можно с помощью команды глобального конфигурирования ОС IOS ipx internal-network, что уже отмечалось ранее в разделе "Система адресации и структура адреса в протоколе IPX".

Чтобы разрешить исполнение протокола NLSP, используется команда глобального конфигурирования ОС IOS ipx router nlsp. Эта команда требует использования в качестве параметра тэга, обозначающего NLSP-процесс в ОС IOS. Чтобы ввести определение множества номеров сетей, являющихся частью существующей на данный момент NLSP-области, используется команда глобального конфигурирования ОС IOS area-address. Команда area-address имеет в своем составе две опции: IPX-адрес сети и маску. Маска показывает, какая часть номера области идентифицирует область, а какая — отдельные сети в этой области. Хотя в сети компании ZIP не применяется протокол NLSP, приведенный ниже пример показывает активацию протокола NLSP на маршрутизаторе в Сингапуре. Последующее применение команды area-address описывает область из 16 сетей с номерами, лежащими в диапазоне от 4000 до 400F:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#ip router nlsp 1
Singapore(config-ipx-router)#area-address 4000 FFFO
Singapore(config-ipx-router)#^Z
```

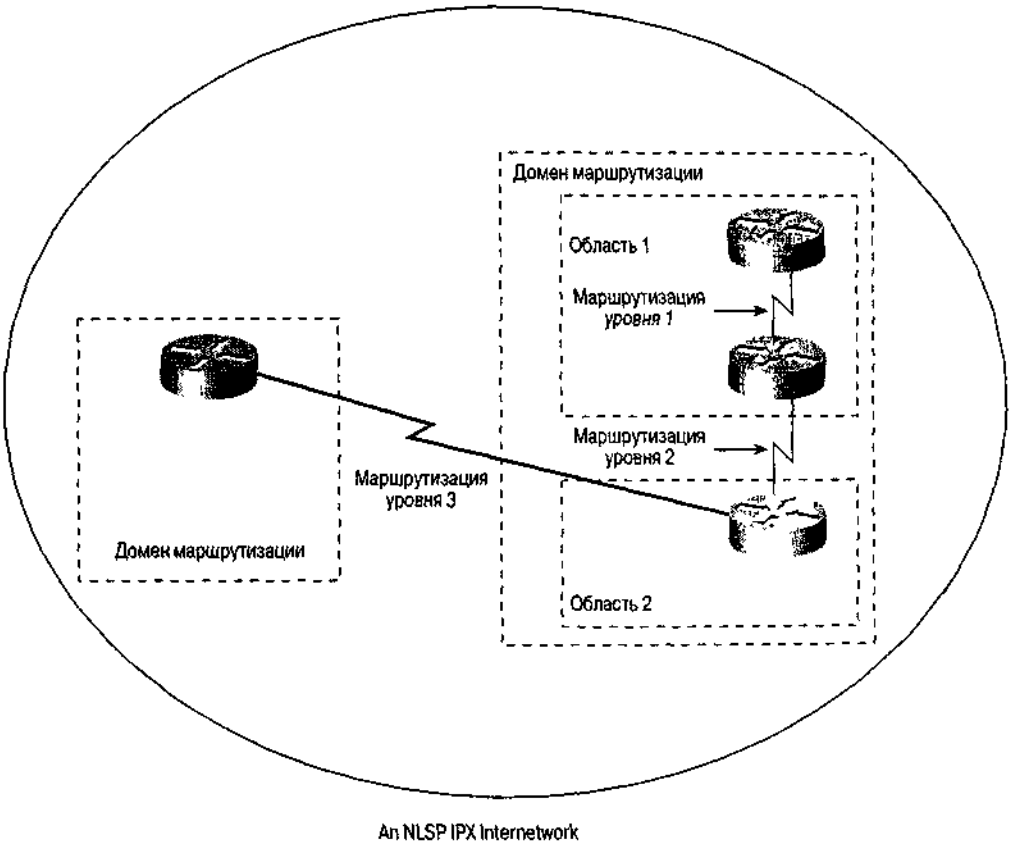


Рис. 6.3. Иерархическая структура IPX-сети, использующей протокол NLSP

Протокол NLSP должен активироваться на поинтерфейсной основе с помощь интерфейсной субкоманды ОС IOS ipx nlsp enable. Эта команда конфигурирования задает тэг процесса протокола NLSP, который будет использоваться при отправке маршрутной информации на данный интерфейс. В примере ниже на интерфесе Ethernet 0 маршрутизатора в Сингапуре задается использование NLSP-процесса 1:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+z.
Singapore(config)#interface ethernet 0
Singapore(config-if)#ipx nlsp 1 enable
Singapore(config-if)#^Z
```

Конфигурирование протокола IPX EIGRP

Протокол EIGRP может использоваться в качестве протокола динамической *IPX*-маршрутизации. Как было показано в предыдущих главах, протокол EIGRP обладает характеристиками как протоколов, основанных на методе вектора расстояния (пакеты актуализации маршрутной информации отсылаются только соседям), так и протоколов, основанных на методе учета состояния канала (пакеты актуализации содержат частичные инкрементальные обновления маршрутной информации и время сходимости уменьшено). Для активации работы протокола EIGRP в рамках протокола 1P> используется команда глобального конфигурирования ipx router eigrp. Эта команда требует указывать номер автономной системы, идентифицирующий процесс! протокола EIGRP. В одном административном IPX-домене номер автономной систе-1 мы должен быть единым для всех маршрутизаторов, общающихся с помощью протокола IPX EIGRP.

Субкоманда network связывает номер IPX-сети с протоколом EIGRP, инструктируя! его передавать маршрутную информацию об IPX-сети с этим номером. В примере ниже! осуществляется активация протокола IPX EIGRP на маршрутизаторе Сингапуре с использованием номера автономной системы 25000. Протоколу EIGRP сообщается о необходимости пересылать

маршрутную информацию об IPX-сетях 4010 и 2902:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#ipx router eigrp 25000
Singapore(config)ttnetwork 4010
Singapore(config-ipx-router)#network 2902
Singapore(config-ipx-router)#^Z
```

Совет

Протокол EIGRP можно активировать сразу во всех IPX-сетях, воспользовавшись командой ipx router eigrp вместе с субкомандой network all.

При использовании протокола IPX EIGRP можно заставить ОС IOS посылать SAP-сообщения периодически или только тогда, когда происходят изменения в SAP-таблице. По умолчанию протокол EIGRP посылает SAP-сообщения периодически на интерфейсы локальных сетей, разрешая тем самым поступать объявлениям протокола SAP IPX-серверам и клиентам. Периодические SAP-сообщения посылаются по умолчанию также на любой интерфейс, на котором отсутствуют работающие с протоколом EIGRP маршрутизаторы, так как интерфейс не может соединяться с IPX-серверами и клиентами.

Если же на интерфейсе находятся только маршрутизаторы, которые работают с протоколом EIGRP, то можно так сконфигурировать протокол, что SAP-сообщения будут посылаться только в случае возникновения изменений в SAP-таблице. Эта функция может помочь в снижении трафика на интерфейсах глобальной сети, соединяющих устройства, работающие под управлением ОС IOS, что достигается за счет исключения периодической рассылки SAP-сообщений, значительно потребляющей полосу пропускания. Если на интерфейсе глобальной сети присутствует маршрутизатор, работающий протоколом EIGRP, ОС IOS по умолчанию посылает пакеты актуализации маршрутной информации протокола SAP только при изменениях в SAP-таблице.

Чтобы отправлять SAP-сообщения только при изменениях в SAP-таблице, следует воспользоваться субкомандой конфигурирования интерфейса ОС IOS ipx sap incremental-eigrp. Эта команда требует в качестве параметра номер автономно системы для протокола EIGRP. Из результата исполнения команды ОС IOS режим EXEC show ipx servers видно, получены данные об IPX-сервисе из периодически пакетов актуализации маршрутной информации протокола SAP или из протокол EIGRP.

Конфигурирование фильтрации в протоколе IPX с применением списков доступа

Средства фильтрации IPX-пакетов ОС IOS компании Cisco позволяют администратору сети ограничивать доступ к определенным системам, сегментам сети, диапазону адресов и услугам на основе разнообразных критериев. Как и SAP-фильтрация, IP/ фильтрация осуществляется с помощью списков доступа. Фильтры протокола SAP накладывают списки доступа на посылаемые и принимаемые SAP-сообщения. При филь рации IPX-пакетов списки доступа используются для запрещения или разрешения прохождения маршрутизируемого IPX-трафика через определенный интерфейс.

Задание списков доступа

Стандартный список доступа протокола IPX, который нумеруется числами от 8 до 899, позволяет ограничивать поток пакетов на основе IPX-адресов отправителя получателя. Диапазон адресов может задаваться с помощью подстановочных или безразличных масок.

Расширенные списки доступа протокола IPX, нумеруемые числами от 900 до % обладают такими же возможностями по фильтрации, как и стандартные списки доступа. Но дополнительно они позволяют фильтровать на базе протоколов ОС NetWare (например, RIP, SAP и SPX), а также на основе номеров IPX-разъемов. IPX-разъемов используются для идентификации услуг ОС NetWare верхнего уровня. Работу списка доступа можно протоколировать, воспользовавшись в качестве параметра ключевым словом log. Протоколирование будет рассмотрено более подробно в главе 7, "Основы администрирования и управления".

В примере ниже на маршрутизаторе компании ZIP SF-2 конфигурируется стандартный список доступа протокола IPX, который разрешает пакетам, источником которых является IPX-сеть 10, достигать IPX-сети назначения 200:

```
SF-2#configure
Configuring from terminal, memory, or network [terminal]? Enter configuration
commands, one per line. End with CTRL+Z.
SF-2(config)#access-list 800 permit 10 200
SF-2(config)#^Z
```

Как и спискам протокола IP, спискам доступа протокола IPX можно давать имена. Наделение протокола возможностью работы с именованными списками доступа означает, что для идентификации списка доступа можно назначать не номера, а произвольную цепочку символов. Командой для создания именованных списков доступа протокола IPX является команда глобального конфигурирования ОС IOS ipx access-list. Используя именованные списки доступа, можно создавать стандартные, расширенные или SAP-фильтры. В примере ниже предыдущему нумерованному списку доступа протокола IPX на маршрутизаторе сети компании ZIP SF-2 присваивается имя pass-marketing ("пропускать для подразделения маркетинга"):

```
SF-2 #configure
Configuring from terminal, memory, or network [terminal] ?
Enter configuration commands, one per line. End with CTRL+Z.
SF-2(config)#ipx access-list standard pass-marketing
SF-2(config-ipx-std-nacl)#permit 10 200
SF-2(config-ipx-std-nacl)#^Z
```

Наложение списков доступа

Для того чтобы пакеты могли фильтроваться, после задания критериев списка доступа его необходимо наложить на один или несколько интерфейсов. Список доступа может накладываться на интерфейс либо во входящем, либо в исходящем направлении. Входящее направление подразумевает, что пакеты поступают в маршрутизатор из интерфейса. Исходящее направление означает, что пакеты выходят из маршрутизатора и поступают на интерфейс. Список доступа накладывается с помощью субкоманды конфигурирования интерфейса ОС IOS ipx access-group. В качестве параметра команда воспринимает ключевые слова in или out ("внутрь" или "наружу"), при этом, если ключевое слово не вводится, по умолчанию подразумевается наличие слова out. В примере ниже заданный в предыдущем разделе стандартный список доступа 800 накладывается на интерфейс FDDI 0 маршрутизатора сети компании ZIP SF-1:

```
SF-l#configure
Configuring from terminal,  memory,   or network   [terminal]?
Enter configuration commands,   one per line.   End with CTRL+Z.
SF-1(config)#interface fddi  0
SF-1(config-if)#ipx access-group 800 out
SF-1(config-if)#^Z
```

Просмотр поведения списка доступа и проверка правильности его конфигурирования возможны с помощью команд ОС IOS режима EXEC show access-lists и show ipx access-lists. Первая команда показывает все списки доступа, заданные на маршрутизаторе, а вторая — только заданные на маршрутизаторе списки доступа протокола IPX. Каждая из команд может иметь

параметром номер списка доступа, и тогда выводится содержание только этого списка. Если параметр не указывается, то выводятся данные обо всех списках доступа. Ниже приведен результат исполнения команды show ipx access-lists на маршрутизаторе компании ZIP SF-1 для рассмотренных ранее примеров создания списков доступа:

```
SF-l#show ipx access-lists
IPX standard access list 800
permit 10 200
IPX standard access list pass-marketing permit 10 200
```

Установлен ли на интерфейсе список доступа, показывает команда ОС IOS режима EXEC show ipx interface. В восьмой строке приведенного ниже результата испо нения этой команды на маршрутизаторе SF-1 показано, что стандартный список до тупа протокола IPX наложен на исходящие IPX-пакеты:

```
SF-2#show ipx interface fddi 0
FddiO is up, line protocol is up
  IPX address is 10.0000.OcOc.llbb, SNAP [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 800
  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
  IPX accounting is disabled
  IPX fast switching is configured (enabled)
  RIP packets received 54353, RIP packets sent 214343
  SAP packets received 94554422, SAP packets sent 93492324
```

Конфигурирование основных служб удаленного доступа по коммутируемым каналам связи протокола IPX

В этой главе рассматриваются возможности маршрутизации с помощью протокола IPX, реализованные в ОС 1OS компании Cisco. Эта операционная система также позволяет осуществлять удаленный доступ IPX-клиентов во многом с теми же функциями, которые описывались в разделах предыдущих глав, посвященных удаленному доступу по коммутируемым каналам связи в рамках протоколов IP и AppleTalk. Функция удаленного доступа в протоколе IPX дает пользователям возможность получать у< ОС NetWare даже тогда, когда они физически не подключены к выделенным каналам сегмента локальной сети.

ОС IOS позволяет осуществлять удаленный доступ по асинхронным коммутируемым каналам связи и по ISDN-каналам. В этой главе рассматриваются специфические для протокола IPX команды, обычно используемые для IPX-клиентов, работающих с асинхронными коммутируемыми каналами. IPX-доступ по каналам ISDN обычно

используется при маршрутизации между маршрутизаторами с установкой соединения по требованию — тема, которая выходит за рамки настоящей книги.

Как было сказано раньше, настройка удаленного доступа состоит из установки конфигурации асинхронной линии, при которой активируются AAA-службы для пользователей, и конфигурирования опций, специфических для протокола. Конфигурирование асинхронной линии для протокола IPX выполняется точно так же, как для протокола IP, что рассматривалось в главе 4. IPX-клиенты используют в качестве протокола канального уровня протокол PPP, делая конфигурирование AAA-служб абсолютно идентичным их конфигурированию для ранее рассмотренных сетевых протоколов. Дальнейшее обсуждение этого вопроса проводится в главе 7.

Первым шагом в добавлении специфических для протокола IPX опций, связанных с доступом по асинхронным коммутируемым каналам связи, является присвоение IPX-адреса интерфейсу Loopback 0, для чего используется субкоманда конфигурирования интерфейса ОС IOS ipx network (рассмотренная выше). Этот адрес становится номером IPX-сети, который будут использовать удаленные IPX-клиенты. Затем с помощью субкоманды конфигурирования интерфейса ОС IOS ipx ppp-client loopback групповому асинхронному интерфейсу назначается IPX-номер сети интерфейса Loopback 0. Конфигурирование сервера доступа sing2511 сети компании ZIP выглядит так:

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Sing2511(config)#interface loopback 0
Sing2511(config-if)#ipx network 2500
Sing2511(config-if)#interface group-asyncl
Sing2511(config-if)#ipx ppp-client loopback 0
Sing2511(config-if)#^Z
```

Удаленным IPX-клиентам нет необходимости получать информацию протоколов IPX RIP и SAP. Чтобы исключить отправку на асинхронные интерфейсы пакетов актуализации маршрутной информации с нормальной периодичностью в 60 секунд, можно воспользоваться субкомандой конфигурирования интерфейса ipx update interval. Эта команда требует в качестве параметра слово sap или rip и значение в секундах частоты отправки соответствующих пакетов актуализации на интерфейс. В примере ниже сервер доступа Sing2511 конфигурируется на отправку пакетов актуализации от протоколов IPX RIP и SAP каждые 10 часов (36 000 секунд). При установке командой ipx update interval такого высокого значения интервала предполагается, что IPX-клиент не будет оставаться подключенным в течение 10 часов.

```
Sing2511#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Sing2511(config)#interface group-asyncl
Sing2511(config-if)#ipx update interval sap 36000
Sing2511(config-if)#ipx update interval rip 36000
Sing2511(config-if)#^Z
```

Верификация взаимодействия в сети с протоколом IPX и устранение неполадок

Полезным инструментом для идентификации проблем взаимодействия в IPX-о является эхо-тестирование с помощью специальных IPX-пакетов, или пингов. При работе с протоколом IPX используются два различных типа пингов. Первый — эхо-пакеты Cisco (специальная разработка компании Cisco); на такие эхо-пакеты отвечают только устройства, работающие под управлением ОС IOS. Второй — стандартные эхо-пакеты разработки компании Novell, которые поддерживаются устройств; работающими под ОС IOS, и NetWare-серверами, на которых исполняется протокол

NLSP, отвечающий спецификации версии 1.0 или более поздней.

Использующее ОС IOS устройство, находясь в непривилегированном ре» EXEC, может посылать эхо-пакеты Cisco, для чего необходимо воспользоваться командой ОС IOS режима EXEC ping ipx. По этой команде посылаются пять байтных эхо-пакетов Cisco в формате протокола IPX по заданному IPX-адресу, это показано в примере ниже для маршрутизатора SF-Core-1:

```
SF-Core-l#ping ipx 10.0000.OcOc.23ce
Type escape sequence to abort.
Sending 5,   100-byte IPX cisco Echoes to 10.0000.OcOc.23ce,timeout  is 2
seconds:
! ! ! ! !
Success  rate is 100 percent   (5/5),   round-trip min/avg/max =1/1/4 ms
```

Из этого примера видно, что было отослано пять IPX-эхо-пакетов Cisco и пол но пять откликов от проверяемого адреса. В табл. 6.2 показаны значения символов, выводимых маршрутизатором для каждого посланного IPX-пинга.

Таблица 6.2. Символы, выводимые в ответ на команду ipx ping

!	Получен ответ от исследуемого адреса
.	Сетевой сервер превысил временной предел, ожидая ответ от исследуемого адреса
U	Получено сообщение об ошибке недостижимости IPX-пункта назначения
C	Принят пакет с сообщением о перегрузке в IPX-сети
I	Пользователь вручную прервал тест
?	Принят IPX-пакет неизвестного типа
&	Превышено время жизни IPX-пакета

Команда ОС IOS ping в привилегированном режиме EXEC может использовать для отправки либо эхо-пакетов Cisco, либо стандартных эхо-пакетов компании N> Команда ping в привилегированном режиме также позволяет задавать множество характеристик посылаемых эхо-пакетов, включая количество повторений, размер пакетов и временной предел ожидания эхо-ответа. В примере ниже маршрута: компании ZIP SF-Core-1 отправляет IPX-пинг с помощью команды ping в привилегированном режиме:

```
SF-Core-l#ping
Protocol [ip]:ipx
Target IPX address:10.0000.OcOc.23ce
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Verbose [n]:
Novell Standard Echo [n]:
Type escape sequence to abort.
Sending 5 100-byte IPX echoes to 10.0000.OcOc.23ce,timeout is 2 seconds.
! ! ! ! !
Success rate is 100 percent (5/5)
```

Общую статистику работы протокола IPX на маршрутизаторе компании Cisco можно получить, воспользовавшись командой show ipx traffic. В ее состав входят счетчики таких данных, как общее количество посланных и принятых маршрутизатором пакетов, количество принятых и отосланных широковещательных пакетов, статистика протоколов SAP, IPX RIP, EIGRP и NLSP, а также информация о том, посылал или принимал маршрутизатор IPX-эхо-пакеты. Кумулятивные счетчики команды show ipx traffic обнуляются только при перезагрузке маршрутизатора или выключении-включении питания. Ниже показан пример результата исполнения команды show ipx traffic на маршрутизаторе компании ZIP SF-Core-1:

```
SF-Core-l#show ipx traffic
System Traffic for 0.0000.0000.0001 System-Name:zipnet
```

```
Rcvd: 603143 total, 94947 format errors, 0 checksum errors, 0 bad hop
           count,
        0 packets pitched, 401 local destination, 0 multicast
Beast:  406 received, 6352 sent
Sent:   6355 generated, 0 forwarded
         0 encapsulation failed, 19 no route
SAP:   368 SAP requests, 0 SAP replies, 2 servers
         0 SAP Nearest Name requests, 0 replies
         0 SAP General Name requests, 0 replies
         27 SAP advertisements received, 138 sent
         20 SAP flash updates sent, 0 SAP format errors
RIP:   6 RIP requests, 0 RIP replies, 5 routes
         5629 RIP advertisements received, 6139 sent
0 RIP flash updates sent, 0 RIP format errors
Echo:   Rcvd 0 requests, 0 replies
Sent 0 requests, 0 replies
         0 unknown: 0 no socket, 0 filtered, 0 no helper
         0 SAPs throttled, freed NDB len 0
Watchdog:
         0 packets received, 0 replies spoofed
Queue lengths:
         IPX input: 0, SAP 0, RIP 0, GNS 0
         SAP throttling length: 0/(no limit), 0 nets pending lost route reply
Delayed process creation: 0
EIGRP:  Total received 0, sent 0
         Updates received 0, sent 0
         Queries received 0, sent 0
         Replies received 0, sent 0
SAPs received 0, sent 0
NLSP:   Level-1 Helios received 0, sent 0
```

В дополнение к командам верификации, поиска и устранения неисправное представленным в настоящем разделе, в привилегированном режиме EXEC ОС существует множество отладочных команд debug, призванных оценить работоспособность протокола IPX на маршрутизаторе. Эти команды debug обеспечивают получение как общей, так и подробной диагностической информации, которая может мочь при устранении неполадок и проверке работы маршрутизатора, протоколов маршрутизации и других функций. Самые распространенные команды debug, используемые для протокола IPX, сведены в табл 6.3.

Таблица 6.3. Команды debug для протокола IPX

Команда	Описание
debug ipx eigrp	Выводит содержание пакетов протокола IPX EIGRP, посылаемых и получаемых маршрутизатором
debug ipx nlsp	Показывает деятельность протокола NLSP, исполняемого на маршрутизаторе
debug ipx packet	Выводит данные об IPX-адресах отправителей и получателей паке маршрутизируемых маршрутизатором
debug ipx routing	Показывает изменения в таблице IPX-маршрутизации, явившиеся результатом добавлений и удалений маршрутов
debug ipx sap	Выводит информацию об объявлениях протокола SAP, отправлен и принятых маршрутизатором

Конфигурирование переадресации IPX-пакетов типа 20

Многие приложения в среде ОС NetWare используют сетевую базовую сие ввода/вывода (NetBIOS), чтобы запрашивать службы IPX-серверов. Эти услуги в чают начало и окончание сеанса и передачу информации.

На NetWare-клиенте NetBIOS-приложение, используя протокол IPX, осуществляет

широковещательную рассылку пакетов типа 20 во все IPX-сети, пытаясь получить информацию об именованных узлах в сети. Система NetBIOS воспринимает именованные узлы в качестве ресурсов сети. Таким образом, чтобы общаться такими ресурсами, NetWare-клиенты должны отображать эти именованные узлы на IPX-адреса.

Для отображения именованных узлов на IPX-адреса система NetBIOS использует механизм протокола IPX. Однако, как было доказано в этой книге, маршрутизаторы компании Cisco no умолчанию блокируют все широковещательные пакеты сетевого уровня, включая и IPX-пакеты рассылки типа 20. Если маршрутизатор не переадресовывает пакеты рассылки типа 20, а NetWare-клиенту, исполняющему приложение которое использует NetBIOS, необходимо пройти маршрутизатор, чтобы получит! формацию об именованном узле в сети, то такой клиент не имеет возможности связаться с сервером.

Интерфейсная субкоманда ОС IOS ipx type-20-propagation дает маршрутизатору инструкцию на прием и переадресацию пакетов рассылки типа 20 на другие IPX-интерфейсы, у которых тоже входит в конфигурацию эта субкоманда. Более того, ОС IOS пытается переадресовывать IPX-пакеты рассылки типа 20 интеллектуальным образом: она не размещает эти пакеты на интерфейсах, которые стоят на пути маршрута к интерфейсу исходного отправителя.

Вместо переадресации IPX-пакетов рассылки типа 20 в несколько сегментов сети можно переадресовывать эти пакеты на конкретный сетевой IPX-адрес, тем самым потенциально снижая количество широковещательных пакетов, посылаемых по IPX-сети. Переадресацию IPX-пакетов рассылки типа 20 на конкретный IPX-адрес разрешает команда глобального конфигурирования ОС IOS ipx type-20-helpered. Интерфейсная субкоманда ОС IOS ipx helper-address задает тот конкретный IPX-адрес, на который следует переадресовывать пакеты типа 20. Команды ipx type-20-helpered и ipx type-20-propagation являются взаимоисключающими. ОС IOS должна либо переадресовывать пакеты рассылки типа 20 другим аналогично сконфигурированным интерфейсам, либо переадресовывать их на IPX-адрес.

В примере конфигурации ниже все IPX-пакеты типа 20 на маршрутизаторе сети компании ZIP в Сингапуре переадресовываются через интерфейс Ethernet 0 на конкретный IPX-сервер в Сан-Франциско с IPX-адресом aa.0005. 0112.0474:

```
Singapore#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CTRL+Z.
Singapore(config)#ipx type-20-helpered
Singapore(config)#interface ethernet 0
Singapore(config-if)#ipx helper-address aa.0005.0112.0474
Singapore(config-if)#^Z
```

Резюме

В данной главе рассмотрены главные моменты, связанные с работой группы протоколов, входящих в состав протокола IPX, основные команды для поднятия IPX-сети, а также некоторые дополнительные команды, часто применяемые в крупных IPX-сетях. Конечно, эта глава не превратит читателя в эксперта по IPX-сетям, но она поднимет его уровень и сделает дееспособным. Основные концептуальные положения этой главы выглядят следующим образом.

- IPX-адрес имеет форму *сеть.узел*, где сеть — это 32-разрядный номер, назначаемый сегменту локальной или глобальной сети, а узел — 48-разрядный но мер, назначаемый клиенту или серверу. Сетевая часть адреса назначается администратором сети. Узловая часть часто совпадает с 48-разрядным адресом устройства на канальном уровне.
- Для того чтобы NetWare-клиенты, NetWare-серверы и маршрутизаторы компании Cisco нормально общались в рамках сегмента локальной IPX-сети, все они должны работать с одним и тем же методом IPX-инкапсуляции. Наиболее часто выбор метода инкапсуляции диктуется используемой версией ОС NetWare.

- Как и IP-маршрутизация, IPX-маршрутизация может конфигурироваться вручную или с помощью протоколов динамической маршрутизации. Для протокола IPX таковыми являются протоколы RIP, NLSP и EIGRP. Работа протокола RIP разрешается на всех IPX-интерфейсах по умолчанию сразу после применения команды глобального конфигурирования ipx routing.
- SAP представляет собой протокол динамических услуг, который объявляет услуги, имеющиеся в IPX-сети. Он конфигурируется по умолчанию на сконфигурированных под работу с протоколом IPX интерфейсах. Для ограничения трафика посылаемых и принимаемых маршрутизатором пакетов протокола SAP могут быть использованы SAP-фильтры.
- Чтобы позволить маршрутизаторам компании Cisco принимать и отсылать широковещательные пакеты системы NetBIOS, необходимо воспользоваться командой ipx type-20-propagation или ipx type-20-helpered.
- Для проверки конфигураций и устранения неполадок в IPX-сети используются команды show, debug и ping. Кроме команд, приведенных в табл. 6.4, с соответствующих команд еще можно найти в табл. 6.5.

Таблица 6.4. Сводная таблица команд режима EXEC для конфигурирования протокола IPX

Команда	Описание
clear ipx route	Очищает всю таблицу IPX-маршрутизации или, если он задан, конкретный маршрут
ping *сеть. узел*	Проверяет указанный IPX-адрес на предмет его достижимости и способности отвечать
ping ipx *сеть.узел*	В привилегированном режиме используется для отправки либо эхо-пакетов Cisco, либо стандартных эхо-пакетов компании Novell по указанному IPX-адресу для проверки его достижимости и способности отвечать
show ipx access-list	Показывает все списки доступа протокола IPX, которые даны на маршрутизаторе
show ipx interface brief	Показывает краткую сводную информацию об IPX-сети и статусе всех имеющихся на устройстве интерфейсов
show ipx interface *интерфейс*	Показывает все параметры, связанные с конфигурацией протокола IPX на интерфейс
show ipx route	Выводит таблицу IPX-маршрутизации маршрутизатора
show ipx route *сеть, узел*	Показывает маршрутную информацию для заданного IPX-маршрута
show ipx servers	Показывает список всех известных на текущий момент IPX-серверов
show ipx traffic	Выводит общие статистические данные о работе протокола IPX на маршрутизаторе

Таблица 6.5. Сводная таблица команд конфигурирования для IPX-сетей

Команда	Описание
access-list	Создает нумерованный список доступа и связанные с ним критерии фильтрации
area-address *адрес маска*	Задает префикс адреса области и маску для протокола NLSP
dialer map ipx	Статически отображает IPX-адрес на имена систем и телефонные номера для ISDN-вызовов
frame-relay map ipx	Отображает IPX-адрес на DLCI-идентификатор протокола Frame Relay
ipx access-group *список* [in | out]	Накладывает указанный список доступа на задачу фильтрации входящих или исходящих пакетов на интерфейсе
ipx access-list {extended I sap I standard} *имя*	Назначает именованный список доступа протокола IPX и связанные с ним критерии фильтрации

ipx gns-round-robin	Оговаривает использование метода циклического отбора по списку при выборе подходящих серверов из нескольких, когда маршрутизатор отвечает на GNS-запросы
ipx input-sap-filter *список*	Интерфейсная субкоманда, инструктирующая маршрутизатор фильтровать входящие SAP-пакеты на основе критериев конкретного списка доступа
ipx internal-network *сеть*	Задает внутренний номер сети на маршрутизаторе для протокола NLSP
ipx maximum paths *коли-чество*	Конфигурирует маршрутизатор на разрешение содержать в таблице IPX-маршрутизации заданное количество путей равной стоимости
ipx network *сеть* [encapsulation | secondary]	Задает IPX-сеть для этого интерфейса Как вариант, задает метод инкапсуляции (например, snap и агра), используемый на данном интерфейсе, и определяет, является сеть для данного интерфейса первичной или вторичной
ipx output-gns-filter *список*	Интерфейсная субкоманда, инструктирующая маршрутизатор фильтровать исходящие из маршрутизатора GNS-пакеты на основе критериев заданного списка доступа
ipx output-sap-filter *список*	Интерфейсная субкоманда, инструктирующая маршрутизатор фильтровать исходящие из маршрутизатора SAP-пакеты на основе критериев заданного списка доступа
ipx ppp-client loopback	Интерфейсная субкоманда, которая назначает IPX-номер интерфейсу обратной петли для использования IPX PPP-клиентами
ipx route	Конфигурирует статический IPX-маршрут
ipx router eigrp *автономная система*	Разрешает использовать протокол EIGRP в качестве процесса маршрутизации протокола IPX
ipx router nlsp тэг	Разрешает использовать заданный процесс протокола NLSP в качестве процесса маршрутизации протокола 1P>
ipx router-sap-filter	Накладывает фильтр на все объявления протокола SAP на основе критериев заданного списка доступа
ipx routing	Разрешает IPX-маршрутизацию на маршрутизаторе
ipx sap	Задает записи в статической SAP-таблице
ipx sap-incremental-eigrp	конфигурирует маршрутизатор таким образом, чтобы он посылал SAP-сообщения только при изменениях в SAP-табл
ipx update interval {rip I sap) *секунды*	Интерфейсная субкоманда, изменяющая интервал отправки IPX RIP или SAP-пакетов до заданного количества секунд
map group	Назначает именованную группу отображений интерфейс для использования при отображении на интерфейсе IPX адресов на ATM-адреса канального уровня
map list	Создает именованный список отображений для конфигурирования отображения IPX-адресов на постоянные или кс мутируемые виртуальные каналы ATM-системы адресаи
network *сеть*	Связывает номер IPX-сети с протоколом EIGRP
x25 map ipx	Статически отображает IPX-адрес на адрес протокола X

Дополнительная литература

Предмет данной главы более подробно рассматривается в следующих монографиях.

1. Currid, C. and A. Currid. *Novell's Introduction to Networking.* Foster City, Califc IDG Books Worldwide, 1997.
2. Heywood, D. *Novell's Guide to TCP/IP and Intranetware.* Foster City, Califc IDG Books Worldwide, 1997.

3. Siyan, K..S. et al. *Novell Intranetware Professional Reference.* Indianapolis, Inc New Riders Publishing, 1997.

Глава 7

Ключевые темы этой главы

- **Основы управления** доступом. Основы конфигурирования средств управления доступом к устройству с использованием в ОС IOS протоколов RADIUS и TACACS+.
- **Основы предотвращения атак** Основные моменты, связанные с настройкой функций ОС IOS для предотвращения некоторых распространенных в сети Internet атак отказов в обслуживании (DoS).
- **Основы управления сетью.** Краткий обзор простого протокола управлению сетью (Simple Network Management Protocol — SNMP) и его конфигурирование в ОС IOS компании Cisco.
- **Основы управления временем.** Настройка протокола системы сетевого времени Network Time Protocol и системные часы в устройствах компании Cisco

Основы администрировать и управления

В этой главе рассматриваются основные вопросы, связанные с управлением IOS, которые существенны для создания надежных и эффективных сетей передачи данных Эти вопросы включают управление доступом к устройствам компании Cisco, протоколирование системной деятельности, предотвращение атак, конфигурирование протоколов управления сетью и синхронизацию времени и даты на устройства тающих под управлением ОС IOS компании Cisco.

Основы управления доступом

ОС IOS компании Cisco предлагает ряд механизмов и протоколов, которые помогают в управлении доступностью устройств Эти базовые механизмы управления доступом могут оказать помощь в ограничении круга тех, кто обращается к устройствам сети, а также того, что они делают на каждом из устройств Таким образом обеспечивается безопасность сети и создается протокол любых изменений в сети.

Подключение к виртуальному терминалу с использованием протокола Telnet и оболочки

Общими методами доступа к устройству, работающему под управлением являются подключение через порт консоли (как описано в главе 2) или подключение по каналам виртуального терминала (vty). Каналы виртуального те представляют собой программное обеспечение, которое дает возможность подключаться к маршрутизатору по сети данных Работающее под управлением устройство также поддерживает пять одновременных сеансов через каналы виртуального терминала

Использование клиента протокола Telnet и клиента защищенной оболочки Shell (SSH) — вот два наиболее общеупотребительных метода подключения виртуального терминала Для создания незащищенного соединения с серверным программным обеспечением, работающим на канале виртуального терминал клиент использует стандартный протокол, описанный в Запросе на комментарий № 854 По умолчанию все основанные на ОС 1OS устройства имеют Telnet- сервер активированным на всех каналах виртуального терминала; методы защиты этих каналов будут рассматриваться в следующем разделе "Активация SSH-сервера".

SSH представляет собой протокол, который обеспечивает защищенное и шифрованное соединение между SSH-клиентом и сервером, работающим на канале виртуального терминала. Это соединение по своим функциональным характеристикам подобно соединению протокола Telnet. В отличие от Telnet-сервера, SSH-сервер не является активированным по умолчанию на каналах виртуального терминала. Активация SSH-сервера обсуждается в следующем разделе.

Чтобы выбрать, Telnet- или SSH-клиент использовать в конкретной локальной системе, обратитесь за помощью к системному администратору. Исполняющее ОС IOS устройство может играть роль либо Telnet-клиента, либо SSH-клиента, для этого в строке приглашения режима EXEC вводится команда telnet или ssh.

Примечание

В настоящее время существуют две версии протокола SSH: SSH версии 1 и SSH версии 2. На данный момент ОС IOS поддерживает только протокол SSH версии 1.

SSH-клиенты и серверы могут обеспечить аутентификацию пользователя с помощью системы шифрования по открытому ключу, изобретенной Ривестом (Rivest), Шамиром (Shamir) и Аделманом (Adelman) (система RSA). Однако реализованная в SSH-клиенте RSA-аутентификация пользователя не поддерживается в SSH-сервере для ОС IOS компании Cisco. ОС IOS осуществляет аутентификацию пользователей только с применением комбинации из идентификатора пользователя и пароля. Хотя SSH-сервер ОС IOS использует метод RSA для

генерации пары ключей, которые затем применяются при установке шифрованного сеанса с клиентом, что будет описано в следующем разделе.

Протокол SSH обеспечивает защиту соединения между клиентом и сервером за счет применения алгоритмов шифрования стандарта DES (56-разрядная длина ключа) или Triple DES (168-разрядная длина ключа). Следует помнить, что не все версии ОС IOS поддерживают стандарты DES или Triple DES. Поэтому необходимо воспользоваться командой show version и проверить, поддерживает ли версия, исполняемая на устройстве, эти алгоритмы шифрования.

Примечание

Экспорт некоторых алгоритмов шифрования (и шифрование данных с 56-разрядным ключом входит *в* их число) контролируется правительством Соединенных Штатов Америки. Использование этих алгоритмов, а равно и поддерживающей их версии ОС IOS, требует экспортной лицензии.

Активация SSH-сервера

Чтобы активировать SSH-сервер и позволить SSH-клиентам подключаться к каналам виртуального терминала, работающее с ОС IOS устройство должно иметь соответствующим образом сконфигурированные имя хост-машины и имя домена. Как обсуждалось ранее, эти параметры конфигурируются *с* помощью команд глобального конфигурирования hostname и ip domain-name.

Для конфигурирования SSH-сервера необходимо сгенерировать пару RSA-ключей используемых для шифрования сеанса между клиентом и сервером. Генерация пары ключей на устройстве, работающем с ОС IOS, осуществляется с помощью команды глобального конфигурирования crypto key generate rsa. После генерации пары RSA-ключей для устройства активация SSH-сервера на каналах виртуального терминала происходит автоматически. Удаление RSA-ключа выполняется с помощью команды глобального конфигурирования crypto key zeroize rsa, при этом автоматически деактивируется и SSH-сервер.

Примечание

В результатах, выводимых командами show running-config или show startup-config, команды глобального конфигурирования crypto key generate rsa казано не будет.

Активирует SSH-сервер на всех каналах виртуального терминала команда гло! ного конфигурирования ip ssh**:**

```
SF-1#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands one per line. End with CNTL/Z.
SF-1(config)#crypto key generate rsa
SF-1(config)#ip hss
SF-1(config)#^Z
```

Проверка конфигурации протокола SSH

Для просмотра открытого RSA-ключа, используемого протоколом SSH, при! ется команда режима EXEC show crypto key mypubkey rsa:

```
SF-l>show crypto key mypubkey rsa
% Key pair was generated at: 19:01:46 EOT Aug 7 2000 Key name: SF-l.zipnet. om
Usage: General Purpose Key Key Data:
305C300D 06092A86 4886F70D 01010105 00034BOO 30480241 OOC6F6D1 CCBF8B9A
6D3E451F C362DD75 866F084B 04F43C95 OB68BA44 OB8D5B8C 35264CFA 04B8B532
OFF6473C 4768C46F CD820DAF B7CA8C75 4977CF6E 7ED1ACE3 FF020301 0001
% Key pair was generated at: 23:14:52 EOT Aug 29 2000
Key name: SF-l.zipnet. om.server
Usage: Encryption Key
Key Data:
```

```
307C300D   06092A86   4886F70D   01010105   00036BOO   30680261   OOC5D98C   E628790E
17BOBA2B   C31C9521   8543AE24   F19E0988   BF2901DC   11D723EF   3512DD29   C28DBC53
8112755C   307AC527   14B955FO   AODD29AD   AE53BAOO   4D84657B   4C605E8E   6EBDDB6E
4FB98167   8616F964   E067604A   F852A27D   1F9B7AFF   3EC73F5C   75020301   0001
```

Более того, на устройстве, которое работает под управлением ОС IOS, можно с помощью команды show ip ssh посмотреть активные SSH-сеансы:

```
SF-l#show ip ssh
Connection    Version   Encryption   State  Username
    0           1.5       3DES         6      admin
```

Защита порта консоли и виртуальных терминалов

На уровне отдельных устройств, работающих с ОС IOS, можно устанавливать пароль для доступа к порту консоли, для чего следует воспользоваться основной командой ОС IOS line console 0 и субкомандой password. Для каналов виртуального терминала добавить пароли можно с помощью основной команды line vty о 4 и субкоманды password.

Используя субкоманду access-class команды line, можно задавать список IP-адресов, которые будут иметь возможность подключаться или быть достижимыми через терминальные каналы устройства, работающего с ОС 1OS. Далее, с помощью ключевого слова in или out можно задавать наложение класса доступа в отношении входящих или исходящих сеансов. Эта субкоманда использует список доступа, квалифицирующий IP-адреса до начала каких-либо входящих или исходящих сеансов. Субкоманда access-class может быть применена для разрешения выхода в каналы виртуального терминала исполняющего ОС IOS устройства только с рабочих станций администратора сети, что является дополнительным методом защиты доступа к устройству.

В примере ниже маршрутизатор SF-1 конфигурируется паролем Zipmein для консоли и виртуального терминала:

```
SF-1#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands one per line. End with CNTL/Z.
SF-1(config)#line console 0
SF-1(config)#password Zipmein
SF-1(config)#line vty 0 4
SF-1 (config)#password Zipmein
SF-1(config)#^Z
```

В рабочей конфигурации и конфигурации запуска пароли консоли и виртуального терминала хранятся в виде открытого текста. Если нужно зашифровать все пароли, выводимые на экран какой бы то ни было командой режима EXEC (например, show running-config или show startup-config), воспользуйтесь командой глобального конфигурирования service password-encryption. В результате исполнения этой команды пароли в незашифрованном виде нельзя будет увидеть ни через одну команду режима EXEC. Забытый пароль можно восстановить с помощью задокументированной для каждого типа устройств процедуры компании Cisco.

Альтернативой конфигурированию паролей на каждом устройстве с целью контроля за доступом является использование в сети протокола управления доступом. Такие протоколы управления доступом выполняют три функции: аутентификацию, авторизацию и учет, которые известны под коллективным названием AAA. (От англ, authentication, authorization, accounting. — *Прим. перев.)* *Аутентификация* — это процесс идентификации и проверки личности пользователя. В рамках ОС IOS возможны несколько методов аутентификации пользователя, включая использование комбинации имени пользователя и пароля или передачу уникального ключа. Процесс *авторизации* определяет то, что пользователь может делать после успешной аутентификации, например, он может получить доступ к определенным сетевым устройствам и хост-машинам. Функция *учета* представляет собой метод регистрации того, что пользователь делает или сделал.

AAA-функции требуют наличия двух составляющих: клиента, который функционирует на

устройстве, работающем под ОС IOS компании Cisco, и серверного программного обеспечения для управления доступом, которое обычно исполняется на сетевой рабочей станции. Наиболее общеупотребительными протоколами, используемыми для обеспечения связи между AAA-клиентом на устройстве компании Cisco серверным программным обеспечением для управления доступом, являются служба удаленной аутентификации пользователей, устанавливающих соединение по телефонным линиям (The Remote Authentication Dial-In User Service — RADIUS) и система управления доступом на основе применения контроллера управления доступом к терминалу (Terminal Access Controller Access Control System — TACACS+).

Предположим, что пользователь с помощью Telnet-приложения подключается маршрутизатору, в конфигурации которого отсутствует протокол управления доступом. Пользователь немедленно получает приглашение ввести пароль канала виртуального терминала в следующем виде:

```
% telnet Singapore
Trying...
Password:
```

Введя правильный пароль, пользователь получает доступ к режиму EXEC маршрутизатора. Такой пользователь не является предметом аутентификации или авторизации и может выполнять любую задачу (включая вход в привилегированный режим если известен пароль). Более того, пользователь, выполняющий такое действие, регистрируется в журнале. Очевидно, что такая открытая политика неприемлема почти во всех сетях. Единственным исключением могут быть лаборатории или испытательные полигоны, когда неконтролируемый доступ к устройству многих пользователей не оказывает существенного влияния на степень защиты, конфигурацию *v* производительность сети.

Если устройство, работающее под управлением ОС IOS, имеет настройки на использование протокола управления доступом, то оно приглашает пользователя ввести имя и пароль:

```
% telnet Singapore
Trying...
Username: allan
Password:
```

При использовании протокола управления доступом устройство, работающее с ' IOS, выполняет следующие действия.

1. Получая внешний запрос на установление соединения по протоколу Telnet, клиент управления доступом в устройстве предлагает ввести имя пользователя и пароль.
2. Клиент управления доступом опрашивает пользователя и затем в виде запроса на аутентификацию посылает комбинацию из имени пользователя и пароля сервер управления доступом.
3. Сервер управления доступом выполняет аутентификацию комбинации имени пользователя и пароля. Эта комбинация либо проходит аутентификацию, либо нет, при этом клиенту отсылается назад соответствующее сообщение. Сервер может также дать клиенту информацию о степени авторизации пользовать Сервер открывает транзакцию.
4. Клиент управления доступом принимает или отвергает комбинацию имени пользователя и пароля. Если комбинация принимается, то пользователь получает право доступа к системе и авторизуется на выполнение действий, определенных в авторизационной информации, переданной сервером.

Эта последовательность взаимодействий между клиентом и сервером управления доступом показана на рис. 7.1.

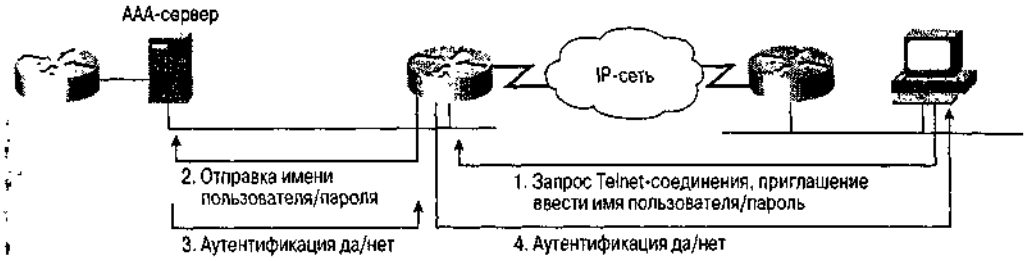


Рис. 7.1. В роли AAA-клиента устройство, работающее под управлением ОС IOS, обменивается информацией с AAA-сервером для решения задачи управления доступом

Активация AAA-служб

Чтобы активировать все AAA-службы в ОС IOS, необходимо воспользоваться командой глобального конфигурирования aaa new-model.

Затем, используя команды глобального конфигурирования aaa authentication, aaa authorization и aaa accounting, можно активировать AAA-клиент с конкретной конфигурацией аутентификации, авторизации и учета. Каждая из AAA-команд конфигурируется с помощью списков методов. Список методов представляет собой сконфигурированный список, описывающий AAA-методы, которые будет пытаться в порядке следования применить клиент для аутентификации пользователя, авторизации его деятельности и учета действий. Например, с помощью списков методов можно задать несколько механизмов аутентификации в попытке все-таки аутентифицировать пользователя, если начальный метод потерпит неудачу. Устройство с ОС IOS пытается использовать для аутентификации пользователя первый метод из перечисленных в списке. Если этот метод не дает отклика, устройство пробует применить следующий метод аутентификации из приведенных в списке. Это продолжается до тех пор, пока не произойдет успешного завершения общения по одному из методов аутентификации, указанному в списке, или пока не будут использованы все заданные методы. Списки методов авторизации и учета работают аналогично тому, как было описано выше для списка методов аутентификации.

Примечание

Устройство с ОС IOS пытается использовать следующий метод из списка методов '"только в том случае, если оно не может обмениваться данными по предыдущему методу. Например, если какой-либо метод аутентификации дал ответ, но аутентификация пользователя не прошла, то следующий метод аутентификации не используется.

Двумя наиболее употребительными AAA-протоколами являются RADIUS и TACACS+, описание которых будет приведено ниже. С помощью команд глобального конфигурирования aaa authentication, aaa authorization и aaa accounting использование в качестве метода протокола RADIUS можно задать, применив о group radius, а протокола TACACS+ — опцию group tacacs+.

Команда aaa authentication задает протоколы аутентификации с помощью упорядоченного списка методов, которые устройство может пытаться использовать для верификации доступа. Команда aaa authorization позволяет задавать выполнение авторизации по каждой команде режима EXEC или только в начале сеансов режима EXEC или сетевых сеансов (например сеансов протокола PPP). Она также позволяет задавать протокол, используемый при выполнении этих задач. В свою очередь, команда aaa accounting определяет события, после которых производится отправка серверу отчетных сообщений, например, в начале или конце каждого (пользователя либо после каждой команды. Эта команда также задает тип учета, выполняемого AAA-клиентом. Можно вести учет деятельности системы IOS, связанных с сетью служб (например, PPP или ARAP) и EXEC-сеансов. Для пересылки учетной информации от AAA-клиенту к AAA-серверу можно использовать как протокол TACACS+, так и протокол RADIUS.

В примере ниже выполняется конфигурирование AAA-процессов на маршрутизаторе в Сингапуре. С помощью команды глобального конфигурирования aaa authentic login осуществляется активация AAA-аутентификации сеансов регистрации в системе. Первым протоколом аутентификации в списке методов стоит TACACS+. Если протокола TACACS+ не способен установить контакт с сервером для выполнения аутентификации, устройство будет выполнять ее с помощью второго метода — команд глобального конфигурирования enable secret или enable password. Этот список методов виден в команде aaa authentication login как опция group tacacs+, за корой следует опция enable.

Совет

Не полагайтесь в аутентификации сеансов регистрации на устройствах с IOS исключительно на AAA-протокол. Наличие второго метода аутентификации сеансов регистрации гарантирует, что доступ к устройству можно получить всегда, даже если сервер недоступен.

При конфигурировании команд aaa authorization и aaa accounting и используется та же логика, которая применялась для команды aaa authenticate. Используя в команде глобального конфигурирования aaa authorization опции exec и network, можно задать различные методы авторизации для сеансов режима Exec и сетевых сеансов (например сеансов протокола PPP). Обозначающее метод ключевое слово if-authenticated говорит AAA-клиенту, чтобы тот в случае успешной аутентификации сеанса выдавал авторизацию.

Наконец, учетные сообщения по всем EXEC-сеансам выдаются только после окончания использования ими протокола TACACS+, в свою очередь, используемого командой глобального конфигурирования aaa accounting.

```
Singapore#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands  one per line. End with CNTL/Z.
Singapore(config)#aaa new-model
Singapore(config)#aaa authentication login default group tacacs+ enable
Singapore(config)#aaa authorization exec group tacacs+ if-authenticated
Singapore(config)#aaa authorization network group radius if-authenticated
Singapore(config)#aaa accounting exec stop-only group tacacs+
Singapore(config)#^Z
```

В этом примере опция group tacacs+ инструктирует устройство с ОС IOS связываться с TACACS+-сервером, задаваемым командой глобального конфигурирования tacacs-server host, что обсуждается в разделе "Протокол TACACS+". Используя команду глобального конфигурирования aaa server group и субкоманду server, можно вводить определения своих собственных групп AAA-серверов с задаваемым пользователем именем группы. Задаваемая пользователем группа AAA-серверов полезна в тех случаях, когда есть группа пользователей, работающих с одним AAA-сервером, и другая группа пользователей, которые работают с другим AAA-сервером. Эти две группы могут использовать или не использовать один и тот же AAA-протокол (скажем, RADIUS). До изобретения групп AAA-серверов все пользователи для каждого метода могли использовать только один набор AAA-серверов. Чаще всего группы AAA-серверов применяются для аутентификации удаленных, устанавливающих соединение по коммутируемым каналам пользователей с помощью одного RADIUS-сервера и аутентификации сетевых администраторов — с помощью другого.

В последующих разделах описывается процедура задания RADIUS- и TACACS+-серверов для AAA-клиента.

Протокол RADIUS

Впервые спецификация протокола RADIUS была опубликована компанией Livingston Enterprises, Inc., где он был определен в качестве протокола обмена AAA-информацией между RADIUS-клиентом и сервером. Протокол RADIUS является открытым протоколом; множество

разнообразных сетевых устройств имеют клиентскую часть протокола RADIUS. RADIUS-сервер представляет собой рабочую станцию, на которой исполняется программное обеспечение серверной части протокола RADIUS от поставщика или какой-либо компании, например, Livingston, Merit или Microsoft. Задать IP-адрес RADIUS-сервера, с которым будет общаться клиент из ОС IOS, можно с помощью команды глобального конфигурирования radius-server host.

При аутентификации протокол RADIUS шифрует пароли, посылаемые между клиентом и сервером. Для такого шифрования необходимо сконфигурировать на RADIUS-сервере и в ОС IOS секретную цепочку. Чтобы сконфигурировать эту цепочку в клиенте ОС IOS, следует воспользоваться командой глобального конфигурирования radius-server key.

Маршрутизатор сети компании ZIP в Сан-Хосе конфигурируется адресом RADIUS-сервера и ключом шифрования следующим образом:

```
San Jose#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands  one per line. End with CNTL/Z.
San Jose(config)#radius-server host 131.108.110.33
San Jose(config)#radius-server key Radius4Me
San Jose(config)#^Z
```

Протокол TACACS+

TACACS+ представляет собой AAA-протокол, который концептуально подобен протоколу RADIUS. TACACS+ — это третья ревизия протокола TACACS. Вторая ревизия называлась Extended TACACS или XTACACS (расширенный протокол TACACS). Протокол TACACS+ является протоколом собственной разработки компании Cisco, и все устройства, работающие с ОС IOS, имеют родной TACACS+-клиент.

Серверное программное обеспечение протокола TACACS+ доступно из многих источников, включая компанию Cisco (в продукте CiscoSecure) и других поставщиков, и для многих аппаратных платформ рабочих станций. Задать IP-адрес TACACS+-сервера, с которым будет общаться клиент из ОС IOS, можно с помощью команды глобального конфигурирования tacacs-server host.

Протокол TACACS+ шифрует всю коммуникацию между клиентом и сервером. Для такого шифрования сообщений необходимо сконфигурировать на TACACS+-сервере и в ОС IOS секретную цепочку. Чтобы сконфигурировать эту цепочку в клиенте ОС IOS, следует воспользоваться командой глобального конфигурирования tacacs-server key.

Маршрутизатор сети компании ZIP SF-1 конфигурируется адресом TACACS+-сервера и ключом шифрования следующим образом:

```
SF-Core-l#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands one per line. End with CNTL/Z.
SF-Core-l(config)#tacacs-server host 131.108.110.33
SF-Core-1(config)#tacacs-server key ZIPSecure
SF-Core-l(config)#^Z
```

Сравнение протоколов RADIUS и TACACS+

Различий между протоколами RADIUS и TACACS+ достаточно много, но выполняемые ими функции, по сути, одинаковы. Протокол RADIUS, являющийся стандартом, использует на транспортном уровне протокол UDP. Протокол же TACACS+, являясь частной разработкой, применяет на транспортном уровне протокол TCP. Протокол RADIUS хорошо работает только в IP-средах, тогда как протокол TACACS+ полезен в многопротокольных средах. В настоящее время протоколом RADIUS поддерживается больше количество атрибутов, и он позволяет передавать клиенту и серверу больше информации, чем протокол TACACS-K Наконец, RADIUS шифрует только пароль, пересылаемый между клиентом и сервером, тогда как TACACS+ шифрует всю пересылаемую информацию.

Многие поставщики, поддерживающие тот или иной протокол, яростно спорят о преимуществах "своего" протокола. Компания Cisco поддерживает оба протокола. Если сеть в

значительной степени гетерогенна, то лучше всего выбрать протокол RADIUS, так как его поддерживают многие поставщики. Если сеть использует главным образом устройства компании Cisco, то, скорее всего, правильным решением будет применение протокола TACACS+.

Основы предотвращения атак

Имеющиеся в ОС IOS функции TCP-перехвата и одноадресной пересылки по обратному пути позволяют сконфигурировать некоторые базовые средства защиты от двух типов атак отказов в обслуживании: заполнение сети пакетами TCP SYN и подделка IP-адреса отправителя.

Атака отказов в обслуживании представляет собой ситуацию, когда хакер переполняет сетевые ресурсы трафиком, который не повреждает данные, но использует достаточный объем ресурсов сети, чтобы она не могла выполнять свою основную задачу. Например, атака заполнением пакетами TCP SYN (синхронизации) возникает, когда хакер заполняет сервер большим количеством TCP SYN-запросов (используемых для инициализации TCP-соединений) из некорректного IP-адреса отправителя. Каждый из этих запросов имеет недостижимый IP-адрес отправителя, т.е. соединения не могут быть установлены. Большое количество неустановленных открытых соединений переполняет сервер и может привести к тому, что он будет отказывать в обслуживании корректных запросов, не давая пользователям подключиться к серверу.

TCP-перехват

Функция TCP-перехвата помогает предотвратить заполнение сети SYN-запросами путем перехвата и проверки достоверности запросов на установление TCP-соединений при их прохождении через маршрутизатор. Функция TCP-перехвата может работать на перехват входящих TCP SYN-сообщений или отслеживать TCP-соединения, когда маршрутизатор пересылает их.

В режиме перехвата маршрутизатор активно перехватывает каждый входящий TCP SYN-запрос и отвечает за реальный сервер-получатель пакетом подтверждения TCP ACK и пакетом SYN. Это является первым шагом в стандартном процессе установления TCP-соединения, называемом *трехсторонним рукопожатием*. Затем маршрутизатор ожидает получения пакета TCP ACK на второй пакет TCP SYN от отправителя. После получения подтверждения ACK маршрутизатор устанавливает правильное TCP-соединение с отправителем и завершает трехстороннее рукопожатие. Затем маршрутизатор посылает начальный пакет TCP SYN реальному серверу-получателю и выполняет второе трехстороннее рукопожатие. После этого маршрутизатор прозрачным образом объединяет эти два TCP-соединения, пересылая пакеты между двумя соединениями в течение всего времени жизни соединения.

Режим перехвата функции TCP помогает не допустить атаки заполнением пакетами TCP SYN, так как пакеты от недостижимой хост-машины никогда не попадут серверу-получателю. Сконфигурировать маршрутизатор на перехват запросов можно путем применения расширенного IP-списка доступа, который позволяет задать подлежащие перехвату маршрутизатором запросы.

Чтобы не перехватывать каждое TCP-соединение, можно сделать так, что функция TCP-перехвата будет наблюдать за запросами на соединение при их пересылке маршрутизатором. Если в течение сконфигурированного временного интервала TCP-соединение не будет инициализировано, то программное обеспечение ОС IOS перехватит и оборвет попытку такого соединения.

Функция TCP-перехвата конфигурируется с помощью команды глобального конфигурирования ip tcp intercept mode. Команда глобального конфигурирования ip tcp intercept list назначает расширенный IP-список доступа, задающий те запросы, которые маршрутизатор должен перехватывать. Команда ip tcp intercept watch-timeout задает количество секунд, допускаемых маршрутизатором до сброса TCP-соединения, не завершившего процесс корректного трехстороннего рукопожатия с сервером-получателем. По умолчанию маршрутизатор будет сбрасывать TCP-соединение, если трехстороннее рукопожатие не завершится за 30 секунд. В примере ниже маршрутизатор SF-Core-1 конфигурируется на наблюдение за всеми TCP-соединениями из сета 131.108.0.0 и сброс соединений, которые не устанавливаются за 15 секунд:

```
SF-Core-l#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands one per line. End with CNTL/Z.
SF-Core-1(config)#access-list 120 permit ip any 131.108.0.0 0.0.255.25
SF-Core-1(config)#ip tcp intercept mode watch
SF-Core-1(config)tip tcp intercept list 120
SF-Core-1(config)lip tcp intercept watch -timeout 15
SF-Core-1(config)#^Z
```

Команда режима EXEC show tcp intercept connections выводит на экран незавершенные и установленные TCP-соединения. Другая команда режима E) show tcp intercept statistics показывает статистические данные о поведении функции TCP-перехвата.

Одноадресная пересылка по обратному пути

Функция одноадресной пересылки по обратному пути (unicast reverse path forwarding) может помочь в предотвращении атак отказов в обслуживании из-за фальшивых IP-адресов отправителя (иногда называемых *IP-спуфингом).* При атаке на сеть фальшивыми IP-адресами отправителя используются искаженные IP-адреса отправителя или быстро меняющийся IP-адрес отправителя. Если сеть подвергается атаке такими искаженными IP-адресами отправителя или набором быстро изменяющихся IP-адресов отправителя, то может оказаться невозможным сконфигурировать список доступа, который остановит атаку.

Примечание

Функция одноадресной пересылки по обратному пути доступна на устройстве только в том случае, если при этом используется механизм экспресс-переадресации Cisco (Cisco Express Forwarding — CEF). CEF представляет собой сложный механизм, используемый для переадресации пакетов и построения таблиц IP-маршрутизации. В настоящее время механизм CEF работает только в некоторых старших моделях устройств, использующих ОС IOS.

Функция одноадресной пересылки по обратному пути решает эту проблему за счет автоматического уничтожения пакетов, которые не имеют поддающегося верификации IP-адреса отправителя. Проверяя, есть ли адрес и интерфейс маршрутизатора отправителя в таблице IP-маршрутизации, и согласуется ли этот интерфейс с тем, на который этот пакет был принят, маршрутизатор верифицирует IP-адрес отправителя. Принятый маршрут и маршрут в обратном направлении к IP-адресу отправителя, как он показан в таблице маршрутизации, должны быть симметричными. Маршрут симметричен, если пакет поступает на интерфейс маршрутизатора, стоящего в одном из наилучших путей возврата к отправителю пакета, при этом не является ограничением точное соответствие с интерфейсом маршрутизатора отправителя, что позволяет использовать такие методики маршрутизации, как балансировка нагрузки по путям равной стоимости

Если нет маршрута обратного пути на тот же интерфейс отправителя или пути возврата для того пути, с которого пакет был принят, то это, вероятно, означает, что адрес отправителя был модифицирован или подделан, и пакет уничтожается. Верификация достижимости IP-адреса отправителя с помощью обратного пути, на который будет переадресовываться пакет, помогает не допустить подделки IP-адреса отправителя.

Функция одноадресной пересылки по обратному пути может использоваться в сети с любой конфигурацией, в которой есть только один путь, по которому можно взаимодействовать с сетью извне. Если такой путь единственный, даже если существует несколько путей разделения нагрузки, то маршрутизация в сети почти всегда симметрична. Такая конфигурация часто возникает в точке выхода восходящего потока данных сети к сети Internet Не следует использовать функцию одноадресной переадресации по обратному пути во внутренней сети организации, когда существуют несколько различных маршрутов к IP-адресам получателей

Конфигурирование функции одноадресной пересылки по обратному пути осуществляется с помощью единственной интерфейсной субкоманды ip verify unicast reverse-path. В

обыкновенной среде эта команда используется в отношении только того интерфейса (или интерфейсов, если это среда с разделением нагрузки) маршрутизатора, через который проходит восходящий поток данных в сеть Internet.

Устройства, работающие под управлением ОС IOS, имеют возможность вести журнал сообщений о деятельности в системе Эти регистрируемые в журнале сообщения могут быть полезны при отслеживании действий в системе, ошибок и извещений. При ведении журнала используются восемь уровней извещающих сообщений, которые сведены в табл. 7.1.

Таблица 7.1. Регистрируемые в журнале сообщения ОС IOS

Уровень	**Описание**
Уровень 0 — аварийные	Система стала непригодной для использования
Уровень 1 — тревожные	Требуется немедленное действие для восстановления стабильности системы
Уровень 2 — критические	Сложились критические условия, которые могут потребовать внимания
Уровень 3 — ошибки	Возникли ошибки, которые могут помочь в отслеживании проблем
Уровень 4 — предупреждения	Сложились предпосылочные условия, но они не носят серьезного характер
Уровень 5 — извещения	Нормальные, но важные в смысловом плане условия, подразумевающие наличие извещений
Уровень 6 — информационные	Эти информационные сообщения не требуют действий
Уровень 7 — отладочные	Эти отладочные сообщения предназначены только для процесса устранения неполадок

В ОС IOS устанавливается минимальный уровень протоколируемых сообщений(в терминах серьезности), которые желательно заносить в журнал. Это делается путем указания в команде конфигурирования уровня серьезности по названию. Аварийные сообщения (уровень 0) обладают наивысшим приоритетом, тогда как отладочные (уровень 7) — наименьшим. Все сообщения с заданным уровнем серьезности и выше отсылаются в одно из четырех мест.

- Сервер системного журнала, который конфигурируется командой logging trap.
- Внутренний буфер устройства, который конфигурируется с помощью команды logging buffered.
- Порт консоли устройства, который конфигурируется с помощью команды logging console.
- Терминальные каналы устройства, который конфигурируются командой logging monitor.

Стоящая впереди команда logging является командой глобального конфигурирования, что позволяет задавать уровень сообщений, отсылаемых в каждое место ведения журнала. Сервер системного журнала — это превосходное место для ведения журнала, так как система обычно сохраняет сообщения на жестком диске. Кроме того, поскольку системный журнал представляет собой средство общего назначения, которым пользуются разнообразные программы, можно иметь один центральный источник для протоколирования сообщений от различных устройств.

Внутренний буфер устройства полезен, если отсутствует сервер системного журнала или нужно, чтобы каждое устройство вело свой отдельный журнал событий. Размер внутреннего буфера устройства по умолчанию составляет 4096 байт. Но, используя команду logging buffered, можно изменять его размер. Например, команда logging buffered 8192 задает размер внутреннего буфера устройства в 8192. Будучи полезным в некоторых ситуациях, внутренний буфер размещается в ОЗУ устройства, и поэтому его содержимое теряется при каждой перезагрузке устройства.

Пересылка журнальных сообщений на консоль или в терминальные канал устройства (включая сеансы виртуального терминала) полезна для организации немедленного извещения о критических событиях. Четыре различных места ведения журнала не являются взаимоисключающими, и можно одновременно использовать несколько средств ведения

журнала.

Примечание
Для просмотра журнальных сообщений, выводимых на канал терминала или в сеанс виртуального терминала, необходимо использовать команду режима EXEC terminal monitor. Эта команда может исполняться в привилегированном режиме.

Конфигурирование пересылки сообщений на сервер системного журнала можно выполнить командой logging trap. Для активации в ОС IOS функции пересылки журнальных сообщений в системный журнал следует воспользоваться командой глобального конфигурирования logging, чтобы задать IP-адрес хост-машины, на которой будет осуществляться ведение журнала.

Как уже упоминалось ранее, возможно протоколирование сообщений сразу в нескольких местах. Например, можно отсылать все сообщения уровня 7 и выше на сервер системного журнала. Одновременно аварийные сообщения из-за их критической природы можно отсылать на консоль устройства. В примере ниже маршрутизатор сети компании ZIP Seoul-1 конфигурируется на выполнение протоколирования как раз таким способом, как было описано выше:

```
Seoul-1#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands  one per line. End with CNTL/Z.
Seoul-1(config)#logging 131.108.110.33
Seoul-1(config)#logging trap debugging
Seoul-1(config)#logging console emergencies
Seoul-1(config)#^Z
```

Примечание
Утилита ведения системного журнала syslog протоколирует системные сообщения в текстовый файл на UNIX-станциях или станциях других типов. Чтобы вести журнал сообщений устройства с ОС IOS на сервере системного журнала, необходимо сконфигурировать соответствующий процесс системного журнала. Для этого нужно иметь на UNIX-станции роль суперпользователя, чтобы активировать функцию local? — сервис, используемый всеми устройствами, работающими под управлением ОС IOS. Будучи суперпользователем (с корневым доступом), необходимо добавить в файл */etc/syslog.* config следующую строку:

```
Local7.debug              /var/adm/router.log
```

Затем следует на UNIX-станции перезапустить служебный процесс syslog, что обычно делается с помощью такой команды:

```
% kill -HUP 'cat /etc/syslog.pid'
```

Если все работает хорошо, можно начинать вести журнал устройств с ОС IOS на этой UNIX-станции.

Если устройство с ОС IOS сконфигурировано на протоколирование во внутреннем буфере, то результаты можно просматривать с помощью команды режима EXEC show logging. Если предположить, что маршрутизатор Seoul-1 сконфигурирован на протоколирование в буфере, а также в системном журнале и на консоли, то результат исполнения команды show logging будет выглядеть следующим образом:

```
Seoul-l>show logging
Syslog logging: enabled (0 messages dropped 0 flushes  0 overruns)
      Console  logging:  level debugging    2 messages  logged
      Monitor  logging:   level debugging    2 messages  logged
      Trap logging:   level debugging    2 message lines  logged Logging to
      131.108.110.33    2 message lines  logged
      Buffer logging:   level debugging    2 messages  logged
Log Buffer   (4096 bytes):
```

```
Mar 17 17:45:56: %LINK-3-UPDOWN: Interface Serial0, changed state  to down
Mar 17 18:23:10: %LINK-3-UPDOWN: Interface Serial0, changed state  to  up
```

В выводимой информации показывается, что активирована функция протоколирования в системном журнале. Она также выводит количество сообщений, отправленных на консоль устройства, в терминальные каналы устройства (строка monitor logging) и в системный журнал. Кроме того, показывается количество сообщений, занесенных в буфер. Последние две строки показывают содержание буфера журнала с двумя запротоколированными сообщениями об изменении состояния канала (сообщения уровня 6).

Следует отметить наличие в выводимой информации временных меток. Далее в разделе "Основы управления временем" будут обсуждаться вопросы, связанны конфигурированием маршрутизатора на поддержание этой информации.

Совет

Рекомендуется активация протоколирования сообщений с отладочным уровнем (debug) хотя бы *в* одном месте ведения журнала. Это позволяет гарантировать отправку и запись всех сообщений об ошибках в устройстве, работающем под управлением ОС IOS. Большинство менеджеров сети стремятся делать установку logging trap debug, которая позволяет заносить все сообщения устройстве системный журнал.

Основы управления сетью

Управление сетью — это процесс управления отказами, контроля конфигураций, мониторинга производительности, обеспечения защиты и учета деятельности в сети передачи данных. Каждая из этих задач необходима для полного контроля над средой сети данных, которая является важной составляющей структуры организации. Форум по управлению сетями Международной организации стандартизации (ISO Network Management Forum) определил управление сетью как сумму всех действий, требующихся для управления отказами, конфигурацией, производительностью, средствами защиты и учетом данных в сети.

Платформы управления сетью представляют собой программные системы, спроектированные для выполнения действий по управлению сетью. Некоторыми примерами таких систем являются продукты OpenView компании Hewlett-Packard, Spectrum компании Cabletron, Solstice Enterprise Manager компании Sun, NetView/AIX компании IBM и CiscoWorks2000 компании Cisco. Платформы управления сетью обеспечивают программную архитектуру для приложений по сетевому управлению, которые выполняют разнообразные задачи. Их нельзя сгруппировать в одну категорию. Некоторые формируют карту сети и контролируют статус всех сетевых устройств, обеспечивают реализацию функции управления отказами. Некоторые, являясь средствами для управления производительностью, строят диаграммы использования каналов и посылают сообщения, если на интерфейсе локальной сети возникают ошибки. Другие же следят за моментами, связанными с защитой сети, и посылают сообщения по электронной почте или на алфавитно-цифровые пейджеры.

Приложения управления сетью общаются с программным обеспечением сетевых устройств, называемых *агентами*. Обмен данными между менеджером и агентом возможность менеджеру собирать стандартный набор информации, который определен в базе данных информации для управления сетью (management information Ьase-MIB). Каждая порция информации, существующая в базе данных, называется *о объектом*. База данных информации для управления сетью содержит объекты, которые нужны менеджеру для управления сетью. На рис. 7.2 показаны взаимоотношения между менеджером и агентом.

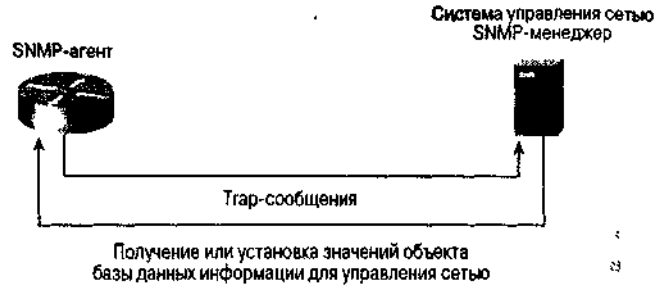


Рис. 7.2. В соответствии с протоколом управления сетью менеджер запрашивает и выполняет установки относительно информации в базе данных управления сетью, а агент посылает менеджеру захваченные сообщения, которые содержат информацию о событиях в устройстве

Существует два типа баз данных управления сетью: стандартные и собственной разработки поставщика. Стандартные базы данных, например типа MIB-II (Запрос на комментарий № 1213), обеспечивают наличие основных объектов, применимых почти ко всем устройствам в сети передачи данных. К примеру, база данных MIB-II содержит такую системную информацию об устройстве, как время нахождения в рабочем состоянии и имя, данные о трафике по интерфейсам и счетчики ошибок, а также информацию протокола IP. Технологически специализированные базы данных, которые тоже относятся к стандартным, ориентированы на конкретные протоколы, например, на Frame Relay (Запрос на комментарий № 1285) или на Token Ring (Запрос на комментарий № 1315). Они содержат объекты, связанные с конкретной технологией, используемой на сетевом устройстве. Специализированные на поставщика базы данных, которые относятся к классу собственных разработок, задают определения объектов, которые специфичны для сетевых устройств одного поставщика.

Приложения управления сетью собирают информацию в базу данных из устройств и изменяют поведение этих сетевых устройств с помощью протокола управления сетью. Стандартным и наиболее распространенным протоколом управления сетью является простой протокол управления сетью (Simple Network Management Protocol — SNMP), определение которого дано в Запросе на комментарий № 1157. Протокол SNMP использует на транспортном уровне протокол UDP и протокол IP на сетевом уровне. Существуют протоколы управления сетью собственной разработки, и некоторые поставщики реализовали их в своих сетевых устройствах.

Обмен данными между SNMP-агентом и менеджером происходит с помощью пяти типов пакетов:

- Get-Request (запрос на получение);
- Get-Next-Request (запрос на получение следующего);
- Set-Request (запрос на установку);
- Get-Response (ответ на запрос);
- Trap (перехват).

Get-Request представляет собой сообщение от менеджера агенту, запрашивающее набор конкретных объектов базы данных информации для управления сетью, например, имя устройства, местоположение, количество физических интерфейс так далее. Get-Next-Request — это сообщение от менеджера агенту с запросом дующей порции табличных данных, ссылка на которые производится из конкретной точки в базе данных информации для управления сетью. Этот тип сообщений полезен при проходах по таблицам базы данных и при извлечении данных из таких лиц, как таблица IP-маршрутизации. Сообщение типа Set-Request содержит запрос агенту на изменение значения конкретного объекта базы данных, например, на изменение статуса интерфейса устройства. Агент на каждое сообщение Get-Request, Get-Next-Request или Set-Request отвечает менеджеру сообщением Get-Response, которое содержит запрашиваемые значения объектов базы данных информации для управления сетью или показывает значение объекта, которое было изменено. Сообщение типа Trap представляет собой сообщение о событии, посылаемое менеджеру по инициативе агента.

В каждом SNMP-агенте устанавливается верификационная последовательность, называемая *цепочкой сообщества* (community string). Цепочка сообщества включает каждый запрос

менеджера на получение или установку информации в базе данных информации для управления сетью. Перед ответом агент проверяет ее. Цепочка сообщества является слабым средством аутентификации и закодирована в кодах ASCII. Не следует полагаться на нее как на единственное средство защиты доступа к SNMP-агенту. (За предложениями по улучшению степени защиты обратитесь к совету, приведенному ниже.)

Конфигурирование агента цепочкой сообщества осуществляется командой глобального конфигурирования ОС IOS snmp-server community. Опции этой команды позволяют ставить условия, чтобы цепочка сообщества применялась к сообщениям, которые имеют статус "только для чтения", или к сообщениям со статусом "чтение-запись". Сообщения Get-Request и Get-Next-Request являются сообщениями только для чтения; сообщения Set-Request относятся к сообщениям чтения-записи. Ключевыми словами, используемыми для задания условий только для чтения и для чтения-записи, являются RO и RW, соответственно. Во многих приложениях управления сетью цепочкой сообщества по умолчанию для сообщений только для чтения является public, а для сообщений с чтением-записью по умолчанию часто используется слово private. Последняя опция этой команды глобального конфигурирования задает стандартный IP-список доступа хост-машин, которым разрешается запрашивать агента с использованием достоверных цепочек сообщества.

В примере ниже маршрутизатор в Сингапуре конфигурируется цепочкой сообщества сообщений только для чтения zipnet и цепочкой сообщества для сообщений с чтением-записью ZlPprivate. Кроме того, задается список доступа access-list 2, который позволяет менеджеру сети с адресом 131.108.20.45 использовать любую из двух цепочек сообщества. Следует отметить, что номер списка доступа являете последним опционным параметром в обеих командах snmp-server community, показанных в этом примере.

```
Singapore#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands  one per line. End with CNTL/Z.
Singapore(config)#access-list 2 permit 131.108.20.45
Singapore(config)#snmp-server community  Zipnet RO 2
Singapore(config)#snmp-server community  ZlPprivate RW 2
Singapore(config)#^Z
```

Примечание

Для усиления защиты SNMP-агента на устройстве с ОС IOS предлагается устанавливать различные цепочки сообщества для доступа только на чтение и для доступа с чтением-записью. Более того, за счет применения в команде snmp-server community опции access-list рекомендуется ограничить количество хост-машин, которые могут опрашивать устройства по протоколу SNMP.

Для отправки SNMP-сообщений типа Trap необходимо выполнить конфигурирование устройства с ОС IOS. В Запросе на комментарий № 1157 определены шесть стандартных SNMP-сообщений, посылаемых всеми агентами:

- coldStart (холодный старт);
- warmStart (теплый старт);
- linkUp (канал в рабочем состоянии);
- HnkDown (канал в нерабочем состоянии);
- authenticationFailure (аутентификация не прошла);
- egpNeighborLoss (потеря EGP-соседа).

Сообщение coldStart означает, что агент был только что запущен. Сообщение warmStart указывает, что само программное обеспечение агента только что было перезапущено. На практике большинство агентов посылают только сообщение coldStart, так как обычно после включения питания устройства, на котором исполняется агент, сам агент перезапускается. Сообщения linkUp и HnkDown обращают внимание менеджера на изменение статуса канала на устройстве. Сообщение authenticationFailure указывает, что менеджер послал агенту SNMP-запрос с неправильной цепочкой сообщества. И, наконец,

сообщение egpNeighborLoss говорит менеджеру о том, что сосед протокола внешних шлюзов (EGP) стал недостижимым. Это последнее Trap-сообщение используется редко, так как протокол EGP перекрывается протоколом BGP4.

Приведенные выше шесть Trap-сообщений являются стандартными, но не единственными Trap-сообщениями SNMP, которые может посылать агент. Многие базы данных информации для управления сетью содержат определения Trap-сообщений, специфичных для протокола, например: сообщения для протоколов ISDN, Frame Relay или BGP4. На время написания этой книги ОС IOS поддерживает Trap-сообщения для разнообразных протоколов и функций IOS, включая сообщения для протоколов BGP, Frame Relay, ISDN, X.25, сообщения монитора среды и изменений конфигурации ОС IOS.

ОС IOS может быть настроена на отправку Trap-сообщений SNMP любому количеству менеджеров. Для задания IP-адреса и цепочки сообщества менеджеру, которому надо будет посылать Trap-сообщения, следует использовать команду snmp-server host. В примере ниже маршрутизатор сети компании ZIP в Сингапуре конфигурируется на отправку Trap-сообщений SNMP менеджеру с IP-адресом 131.108.20.45с использованием цепочки сообщества Zipnet. Опционные параметры команды snap-server host также задают, чтобы агент отсылал Trap-сообщения для протоколов SNMP, Frame Relay и сообщения об изменениях в конфигурации ОС IOS.

```
Singapore#configure
Configuring from terminal memory or network [terminal]?
Enter onfiguration commands cone per line. End with CNTL/Z.
Singapore(config)#snrap-server host 131.108.20.45 Zipnet snmp frame-ralay config
Singapore(config)#^Z
```

Совет

Советуем конфигурировать SNMP-агент на отправку Trap-сообщений обо всех технологиях, которые активны на устройстве. Trap-сообщения протокола SNMP обычно занимают много места, но могут дать полезную информацию для диагностики проблем в сети.

В ОС IOS в SNMP-агенте можно вручную сконфигурировать физическое местоположение и контактную персону по устройству. В этом случае приложения управления сетью могут извлекать эту информацию. Для занесения этой информации следует использовать команды глобального конфигурирования snmp-server location и snmp-server contact. Каждая из этих команд позволяет ввести текстовую строку из 255 символе описывающую местоположение и контактную персону. В примере ниже осуществляете занесение в конфигурацию маршрутизатора в Сингапуре информации о местоположении и контактной персоне:

```
Singapore # configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands one per line. End with CNTL/Z. Singapore(config)#snmp-
server location 1 Raffles Place, Singapore Singapore(config)#snmp-server contact
Allan Leinwand, allanOtelegis.i
Singapore(config)#^Z
```

Команда режима EXEC show snmp демонстрирует статистические данные протокола SNMP для заданного устройства. Эта команда полезна для наблюдения за работой протокола SNMP на устройстве. Ниже приведен результат исполнения этой команды на маршрутизаторе в Сингапуре:

```
Singapore>show snmp
Chassis: 25014624
Contact: Allan Leinwand allan@digisle.net
Location: 45 Raffles Place Singapore
4620211 SNMP packets input
      0 Bad SNMP version errors
```

```
      0 Unknown community name
      0 Illegal operation for community name supplied
      0 Encoding errors
      23493606 Number of requested variables
      0 Number of altered variables
      576553 Get-request PDUs
      4043613 Get-next PDUs
      0 Set-request PDUs 4623230 SNMP packets output
0 Too big errors (Maximum packet size 1500)
      1757 No such name errors
      0 Bad values errors
      0 General errors
      4620166 Get-response PDUs 3064 SNMP trap PDUs SNMP logging: enabled
      Logging to 131.108.20.45 0/10 3064 sent,  0 dropped.
```

В приведенном выше результате присутствуют статистические данные относительно работы протокола SNMP. В первой строке результата показан серийный номер системной платы, который находится в базе данных информации для управления сетью разработки компании Cisco. Вторая и третья строки содержат текстовые последовательности, определяющие местоположение и контактную персону по устройству в соответствии с той информацией, которая была сконфигурирована командами глобального конфигурирования snmp-server contact и snmp-server location. В начале выводимого результата стоят данные об общем количестве SNMP-пакетов на входе, общем количестве SNMP-пакетов на входе, которые были посланы с неправильной цепочкой сообщества и общем количестве SNMP-объектов (названных здесь переменными), запрошенных менеджерами. Также здесь показана разбивка принятых SNMP-пакетов по типам.

Во второй части результата содержатся данные об общем количестве SNMP-пакетов на выходе, различных сообщениях о стандартных ошибках протокола SNMP и об общем количестве отправленных ответов и Trap-сообщений. Последние две строки результата показывают, был ли агент настроен на отправку Trap-сообщений (здесь эта процедура называется "протоколирование SNMP"), IP-адреса каждого менеджера, принимающего Trap-сообщения и количество Trap-сообщений, отосланных каждому конкретному менеджеру.

Основы управления временем

ОС IOS компании Cisco позволяет устройству отслеживать текущее время и дату, используя системные часы. Системные часы запускаются в момент подачи питания на устройство и могут распространять данные о времени в различные внутренние системы, например, для регистрации времени и даты изменений конфигурации, вывода на экран времени занесения в буфер журнала сообщений и отправки времени и даты в сообщениях протокола SNMP. Только в маршрутизаторе Cisco 7000 время системных часов устанавливается аппаратным образом. Во всех других моделях системные часы устанавливаются по умолчанию на полночь 1 марта 1993 года.

После установки времени системные часы определяют надежность источника даты и времени. Если источник времени надежен, то время становится доступным другим процессам ОС IOS, в противном случае оно используется только для демонстрации. В последующих разделах показано, как сделать выбранный источник времени, например атомные часы, надежным.

Дату и время на системных часах можно посмотреть, если воспользоваться командой режима EXEC show clock:

```
SF-l>show clock
06:56:50.314   PST  Fri  Mar  30  2001
```

В маршрутизаторах серии Cisco 7000 имеется календарь, который ведет дату и время, в том числе при перезапусках системы и отказах по питанию. При перезапуске системы для установки системных часов всегда используется календарь. После этого другой протокол может изменять или обновлять показания часов. В сети, в которой нет другого авторитетного источника времени, в качестве такового может использоваться календарь, и его показания могут передаваться другим процессам (например протоколу сетевого времени Network Time Protocol, NTP, который рассматривается в разделе ниже). Увидеть текущие значения системы календаря можно,

воспользовавшись командой режима EXEC show calendar:

```
SF-l>show calendar
06:57:26 PST Fri Mar 30 2001
```

Системные часы ведут отсчет времени внутренним образом, основываясь на универсальном скоординированном времени, также называемом средним гринвичским. ОС IOS позволяет вводить в конфигурацию устройства локальный часовой пояс и, если это имеет место, счисление времени со сбережением светового дня (в синтаксисе ОС IOS это называется летним временем — summer-time). Таким образом, устройство во показывает правильное время на протяжении всего года.

Примечание

Если необходимо, чтобы устройство с ОС IOS указывало текущие дату и время в отладочных и журнальных сообщениях, используйте команду глобального *KOY* конфигурирования service time stamps. Можно выводить время с момента перезапуска устройства, дату и время по Гринвичу или в соответствии с локальные часовым поясом и время с точностью до миллисекунд. Рекомендуется использовать команды конфигурирования service timestamps log datetime localtime и service timestamps debug datetime localtime. Команда service timestamps log datetime localtime вводит дату и время в журнальные сообщения, а команда service timestamps debug datetime localtime вводит их в отладочные сообщения.

Для установки системных часов могут использоваться несколько источнике. Самыми распространенными являются следующие:

- установка вручную;
- протокол сетевого времени NTP;
- простой протокол сетевого времени SNTP.

Все они подробно рассматриваются в последующих разделах данной главы.

Конфигурирование даты и времени вручную

Если устройство, работающее под управлением ОС IOS, стоит обособление и не может использовать внешний авторитетный источник времени, то время и дата устройстве могут устанавливаться вручную. Эти установки достоверны до момента сброса и перезагрузки устройства. Службы управления временем вручную следует использовать только тогда, когда другой авторитетный источник времени недоступен.

Чтобы вручную установить часовой пояс для устройств с ОС IOS, используй команду глобального конфигурирования clock timezone. Эта команда воспринимает в качестве опций часовой пояс, в котором находится устройство, и разницу в часах между этим часовым поясом и универсальным скоординированным временем. Например, для стандартного тихоокеанского времени (Pacific Standard Time — PST), которое на восемь часов отстает от универсального скоординированного времени, вводится следующая глобальная команда: clock, timezone PST -8.

Если в часовом поясе, в котором стоит устройство, используется летнее время, воспользуйтесь командой глобального конфигурирования clock summer-time recurring. Аргументом этой команды конфигурирования является название летнего времени часового пояса, например, тихоокеанское летнее время (Pacific Daylight Time — PDT). Системные часы устанавливаются с помощью команды глобального конфигурирования clock set. В примере ниже в маршрутизаторе SF-1 устанавливается часовой пояс PST, активируется режим летнего времени (в данном случае это PDT) и выполняется установка часов на 17 марта 2001 года, 14:25:

```
SF-1#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands one per line. End with CNTL/Z.
SF-1(config)#clock timezone PST -8
SF-1(config)#clock summer-time PDT recurring
SF-1(config)#clock set 14:25 17 3 2001
```

```
SF-1(config)#^AZ
```

В маршрутизаторах серии Cisco 7000 для установки календаря вручную применяется команда глобального конфигурирования calendar set. Чтобы этот календарь играл роль достоверного источника времени и даты для других функций ОС IOS, воспользуйтесь командой глобального конфигурирования clock calendar-valid.

Протокол сетевого времени

Протокол сетевого времени (Network Time Protocol — NTP), задокументированный в Запросе на комментарий № 1305, синхронизирует время в устройствах, работающих в IP-сети передачи данных. ОС 1OS компании Cisco содержит NTP-процесс, который позволяет устройству посылать и принимать NTP-пакеты. Многие поставщики наделяют свои устройства подобными NTP-процессами, что делает этот протокол оптимальным механизмом для решения задачи синхронизации времени во всей сети.

Протокол NTP распространяет установку времени, которую он получает по сети от авторитетного источника времени. Как уже отмечалось ранее, устройство с ОС IOS может быть настроено так, чтобы оно само играло роль такого источника времени, но предпочтительнее, чтобы источником времени были атомные часы, подключенные к серверу службы времени. Для использования протокола NTP не обязательно иметь собственные атомные часы. Можно синхронизировать время с другим источником, который получает его от атомных часов.

Как и во многих часах телефонных сетей, протокол NTP измеряет расстояние между устройством, на котором он исполняется, и авторитетным источником времени в инкрементах, называемых *стратами.* Часы, являющиеся источником времени *страты 1,* подключены к атомным часам непосредственно, источник *страты 2* синхронизируется с источником *страты 1* и так далее. Вы не можете подключить свое устройство непосредственно к источнику времени *страты 1.* Однако NTP-процесс в ОС IOS компании Cisco автоматически выполняет синхронизацию с источником времени с наименьшей стратой. NTP-процесс, реализованный компанией Cisco, не подстраивает системное время устройства к времени источника, относящегося к той же страте или большей. Если протокол NTP сталкивается с источником времени, у которого время существенно отличается от времени в других устройствах сети, то он не выполняет синхронизацию с таким источником, даже если у него более низкая страта.

Одно устройство, исполняющее протокол NTP, обменивается информацией с другим NTP-устройством, образуя ассоциацию. В ОС IOS компании Cisco ассоциации конфигурируются с использованием команд глобального конфигурирования ntp server или ntp peer. *Серверная ассоциация* означает, что устройство с ОС IOS о образует ассоциацию со сконфигурированным устройством, а не наоборот. При *одноранговой ассоциации* устройства образуют ассоциацию друг с другом. Наиболее распространенным типом ассоциаций является серверная, в которой одним авторитетным источником времени для нескольких NTP-процессов на разных устройствах является сервер. На рис. 7.3 показана серверная ассоциация между NTP-клиентами и устройством, работающим под управлением ОС IOS.

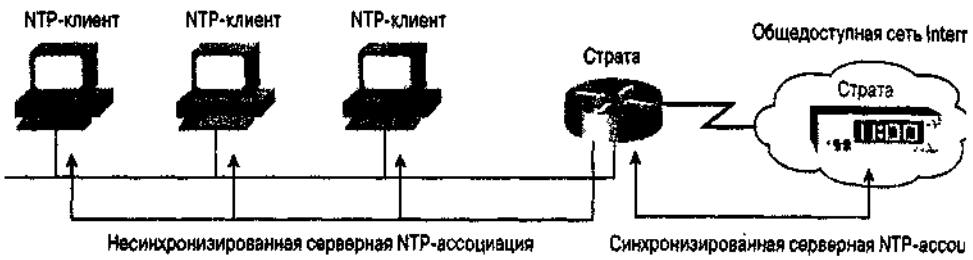


Рис. 7.3. Эти NTP-клиенты входят в серверную ассоциацию с устройством, работающим управлением ОС IOS, которое синхронизируется с авторитетным источником времени общего пользования Internet

Совет

Рекомендуется, чтобы авторитетный источник времени для обслуживания сети размещался в сети общего пользования Internet. Эти источники можно найти, используя поисковые Web-средства, и информацию о них можно регулярно обновлять (поиск по ключевому слову NTP). Общей практикой является наличие нескольких авторитетных источников времени, находящихся в различных местах, где сеть организации ИМЕЕТ возможность подключаться к сети Internet. Например, если сеть имеет подключение к сети Internet в Европе и Соединенных Штатах, то выберите авторитетный источник времени на каждом континенте и позвольте протоколу NTP выбирать источник времени ни для синхронизации.

В маршрутизаторах серии Cisco 7000 с помощью протокола NTP можно периодически осуществлять синхронизацию системы календаря. Для выполнения этой задачи используется команда глобального конфигурирования ntp update-calendar.

В локальной сети отправка и прием NTP-сообщений осуществляется с использованием широковещательных сообщений, что исключает необходимость в конфигурировании и образовании ассоциации с каждым NTP-устройством, находящимся в локальной сети. Для прослушивания широковещательных NTP-сообщений на интерфейсе используется субкоманда конфигурирования интерфейса ntp broadcast client. Чтоб передавать широковещательные NTP-сообщения в заданный сегмент локальной сети, необходимо воспользоваться интерфейсной субкомандой ntp broadcast. В самой распространенной конфигурации устройства, работающие под управлением ОС IOS, на страиваются так, чтобы они могли образовывать серверную ассоциацию с находящимся в сети Internet авторитетным источником времени, а затем рассылают широковещательные NTP-сообщения на все интерфейсы, на которых размещаются другие NTP-устройства. В примере ниже на маршрутизаторе SF-1 конфигурируется NTP-процесс с использованием двух авторитетных источников времени из сети Internet, находящихся в Северной Калифорнии, с периодическим обновлением показаний системы календаря на основе даты и времени протокола NTP и с широковещательной отправкой NTP-сообщений на интерфейс (Ethernet 0):

```
SF-1#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands one per line. End with CNTL/Z.
SF-l(config)#ntp server 192.216.191.10
SF-1(config)#ntp server 129.189.134.11
SF-1(config)#ntp update-calendar
SF-l(config)#interface (Ethernet 0)
SF-1(config)#ntp broadcast
SF-1(config)#^Z
```

NTP-ассоциации, сконфигурированные в устройстве, работающем с ОС IOS, можно просмотреть с помощью команды режима EXEC show ntp associations. Первый символ в каждой строке выводимого результата говорит, является ли конкретная ассоциация синхронизированной (ключом к значению символов из первого столбца служит последняя строка результата). В результате также показывается адрес каждой сконфигурированной ассоциации, страта источника времени и ведущий сервер. Ниже показан соответствующий пример:

```
SF-l>show ntp assoc
address             ref       lock    st      when     poll    reach     delay     offset     disp
*~192.216.191.10     .GPS.           1         127         512      377    285.5     7.57       32.8
+~129.189.134.11     .PPS.           1         207         512      377    147.2   -22.19      18.4
* master   (synced)   # master   (unsynced)   + selected - candidate       ~    configured
```

Используя команду режима EXEC show ntp status, можно узнать статус протокола NTP. В примере результата исполнения этой команды ниже показано, что протокол NTP синхронизирован, относится к страте 2 и в качестве опорного авторитетного источника времени использует

источник с IP-адресом 192.216.191.10:

```
SF-l>show ntp status
Clock is  synchronized    stratum 2     reference is  192.216.191.10
nominal freq is 250.0000 Hz    actual freq is 250.0003 Hz    precision is 2**24
reference time is B853B821.9813EB8D   (06:58:10 PST Fri Mar 30 2001)
lock offset is -7.3067 msec    root delay is 285.46 msec
root dispersion is  41.95 msec    peer dispersion is 32.82 msec
```

Деактивировать работу протокола NTP можно на конкретном интерфейсе с помощью команды ntp disable. Команда глобального конфигурирования ntp access-group вводит ограничение на тип NTP-ассоциации, которую может иметь устройство, работающее под управлением ОС IOS. Эта команда требует задать тип ассоциации, разрешенной с другими устройствами из конкретного множества IP-адресов, указанного в IP-списке доступа. Можно разрешить устройству образовывать одноранговую или серверную ассоциацию. Также можно разрешить ему только системные запросы времени или только NTP-сообшения. В примере, приведенном ниже, на маршрутизаторе SF-1 разрешаются серверные ассоциации со всеми системами из сети 131.108.0.0:

```
SF-1#configure
Configuring from terminal memory or network [terminal]?
Enter configuration commands  one per line. End with CNTL/Z.
SF-1(config)#access-list 50 permit 131.108.0.0 0.0.255.255
SF-1(config)#ntp access-group serve 50
SF-1(config)#^Z
```

Простой протокол сетевого времени

Маршрутизаторы моделей Cisco 1003, 1004 и 1005 работают только с простым протоколом сетевого времени (Simple Network Time Protocol — SNTP), который зад задокументирован в Запросе на комментарий № 2030. SNTP — это упрощенная версия протокола NTP, которая может получать время только от NTP-серверов. Протокол SNTPЫ не может быть авторитетным источником времени для других устройств. Такие ограниченные функциональные возможности, по мнению компании Cisco, были приемлемы в данном случае, так как эти маршрутизаторы серии Cisco 1000 являются малыми устройствами с фиксированным количеством интерфейсов и относительно низкой производительностью. Протокол SNTP обеспечивает получение информации о времени с точностью приблизительно в 100 миллисекунд. Эта информация предназначается для использования в устройствах ОС IOS.

Протокол SNTP можно сконфигурировать на запрос и прием пакетов от сконфигурированных серверов с помощью команды глобального конфигурирования sntp server. Организовать SNTP-процесс в маршрутизаторе, который бы слушал широковещательные пакеты протокола NTP, можно путем применения команды глобального конфигурирования sntp braodcast client. Если ввести в конфигурацию как конкретный сервер, так и возможность маршрутизатора принимать широковещательную информацию, то устройство предпочтет сервер более низкой страты или сконфигурированный сервер, если страты нескольких источников равны. Статистические данные о работе протокола SNTP можно просмотреть, воспользовавшись командой режима EXEC show sntp.

Резюме

Обсужденные в данной главе вопросы, связанные с администрированием и управлением и сведенные в представленный ниже перечень, являются последними элементами основ конфигурирования ОС IOS, которые необходимы для конфигурирования сети компании ZIP. В следующей главе приводятся полные конфигурации всех рабоающих под управлением ОС IOS устройств этой сети.

- В общем случае рекомендуется осуществлять управление доступом к сетевым устройствам через процедуры аутентификации, авторизации и учета (AAA-процедуры). Наиболее используемыми с ОС IOS протоколами управления являются RADIUS и TACACS+.
- Устройства, работающие под управлением ОС IOS, имеют возможность протоколировать сообщения о деятельности системы. Эти сообщения разбиваются на восемь классов по уровню серьезности. Возможно задание минимального уровня сообщений, подлежащих занесению в журнал, а также места, *в* которое эти Протокольные сообщения следует отсылать.
- Приложения управления сетью могут собирать информацию о поведении сетевых устройств и изменять его. Стандартным протоколом для управления сетью является протокол SNMP.
- Для оптимальной работы протокола SNMP следует использовать различные цепочки сообщества для доступа только на чтение и доступа на чтение-запись. Следует также использовать список доступа, который бы ограничивал количество хост-машин, имеющих право опрашивать устройства с ОС 1OS по протоколу SNMP. Кроме того, следует конфигурировать агента SNMP так, чтобы он отсылал Trap-сообщения обо всех технологиях, активных на устройстве.
- Системные часы устройства с ОС IOS могут устанавливаться вручную, протоколом NTP и протоколом SNTP.

В табл. 7.2 приведены основные команды администрирования.

Таблица 7.2. Сводная таблица команд конфигурирования для задач администрирования и управления

Команда	**Описание**
aaa accounting	Активирует службу учета на конкретном клиенте
aaa authentication	Активирует службу аутентификации на конкретном клиенте
aaa authorization	Активирует службу авторизации на конкретном клиенте
aaa new-model	Активирует все AAA-службы в ОС IOS
aaa server-group	Задает группу AAA-серверов
access-class *список доступа* in	Канальная субкоманда Задает список для доступа по терминальному каналу на вход
access-class *список доступа* out	Канальная субкоманда Задает список для доступа по терминальному каналу на выход
calendar set	Установка даты в системных часах вручную
clock calendar-valid	Делает дату и время, поставляемые календарем, достоверным источником для других функций ОС IOS
clock set	Установка времени на системных часах вручную
clock summer-time recurring	Устанавливает часовой пояс с летним временем
clock timezone	Установка часового пояса для устройства с ОС IOS вручную
crypto key generate rsa	Генерирует пару RSA-ключей для шифрования сеанса между SSH-сервером и клиентом. Активирует SSH-сервер на всех каналах виртуального терминала
crypto key zeroize rsa	Удаляет пару RSA-ключей для шифрования сеанса между
SSH-сервером и клиентом	Деактивирует SSH-сервер на всех каналах виртуального терминала
lp ssh	Активирует SSH-сервер
ip top intercept list *список доступа*	Задает расширенный IP-список доступа, который определяет TCP-соединения для функции TCP-перехвата
ip top intercept mode {intercept I watch)	Устанавливает режим работы функции TCP-перехвата на перехват или наблюдение
ip top intercept watch-timeout *секунды*	Задает количество секунд до сброса TCP-соединения, которое наблюдалось, но не было установлено
ip verify unicast reverse-path	Интерфейсная субкоманда активации функции одноадресной переадресации по обратному пути
line console 0	Главная команда для конфигурирования параметров канала консоли

line vty *начало конец*	Главная команда для конфигурирования каналов виртуального терминала, пронумерованных от начала до конца
logging buffered размер	Задает размер внутреннего буфера устройства в байтах
logging *место нахождения уровень*	Определяет протоколирование сообщений с указанным и более высоким уровнем серьезности в заданном месте
ntp access-group	Ограничивает типы NTP-ассоциаций устройства с ОС IOS, которое то может иметь, до определенных в IP-списке доступа
ntp broadcast	Конфигурирует интерфейс на широковещательную передачу NTP-сообщений в данный сегмент локальной сети
ntp broadcast client	Конфигурирует интерфейс на прослушивание широковещательных NTP-пакетов
ntp peer	Конфигурирует одноранговую ассоциацию между двумя сконфигурированными NTP-устройствами
ntp server	Конфигурирует серверную ассоциацию между устройством ОС IOS и сконфигурированным NTP-устройством
ntp update-calendar	Периодически синхронизирует календарь маршрутизаторе серии 7000 с календарем протокола NTP
password *пароль*	Задает пароль канальной субкоманды
radius-server host	Задает RADI US-сервер, с которым обменивается данным! клиент ОС IOS
radius-server key	Конфигурирует секретную цепочку для шифрования связи между RADIUS-сервером и ОС IOS
server	Субкоманда конфигурирования AAA-серверов Задает IP-адреса серверов в группе AAA-серверов
service password-encryption	Конфигурирует устройство с ОС IOS на шифрование всех паролей, показываемых командами режима EXEC
service timedtamps *тип*	Конфигурирует устройство с ОС IOS на введение временных меток в отладочные сообщения и сообщения для журнала
snmp-server community	Конфигурирует цепочку сообщества для защиты SNMP-агента
snmp-server contact	Конфигурирует текстовую строку, указывающую контактную персону по устройству, работающему под управлением ОС IOS
snmp-server host	Задает IP-адрес и цепочку сообщества менеджера, которому должны будут отсылаться Trap-сообщения
snmp-server location	Конфигурирует текстовую строку, указывающую место расположения устройства с ОС IOS
sntp broadcast client	Конфигурирует SNTP-процесс маршрутизатора на прослушивание широковещательных NTP-пакетов
sntp server	Конфигурирует протокол SNTP запрашивать и принимать пакеты от указываемых в конфигурации серверов
tacacs-server host	Конфигурирует TACACS+-сервер, с которым будет общаться клиент ОС IOS
tacacs-server key	Конфигурирует секретную цепочку для шифрования связи между TACACS+-сервером и ОС IOS

В табл. 7.3 приводятся основные команды режима EXEC для администрирования.

Таблица 7.3. Сводная таблица команд режима EXEC для задач администрирования и управления

Команда	Описание
show clock	Показывает текущие значения даты и времени системных часов
show calendar	Выводит на экран текущие значения даты и времени системы календаря в маршрутизаторах серии Cisco 7000
show crypto key mypubkey rsa	Показывает открытый RSA-ключ, используемый в протоколе SSH для шифрования
show ip ssh	Показывает текущие сеансы протокола SSH на устройстве
show logging	Описывает текущий статус протоколирования устройства
show ntp associations	Показывает текущие NTP-ассоциации и их статус
show ntp status	Показывает текущий статус протокола NTP в устройстве
show snmp	Показывает статистику протокола SNMP для SNMP-агента на устройстве с ОС IOS
show sntp	Показывает статус протокола SNTP в устройстве с ОС IOS
show tcp intercept connections	Показывает текущие незаконченные и установленные TCP-соединения
show tcp intercept statistics	Показывает статистические данные функции TCP-перехвата

Дополнительная литература

В приведенных ниже монографиях вопросы этой главы исследуются более подробно

1. Carasik, Anne. *UNIX SSH: Using Secure Shell.* New York: McGraw-Hill Companies Inc., 1999.
2. Case, J.D., M. Fedor, M.L. Scoffstall, and C. Davin. RFC 1157, "Simple Netwoi Management Protocol (SNMP)". May 1990.
3. Ferguson, P., and D. Senie. RFC 2827, "Network Ingress Filtering: Defeating Dena of Service Attacks Which Employ IP Sourse Address Spoofing". May 2000.
4. Finseth, C. RFC 1492, "An Access Control Protocol, Sometimes Called TACACS July 1993.
5. Leinwand, Allan, and Karen Fang-Conroy. *Network Management: A Practical Perspecth* Second Edition. Reading, Massachusetts: Addison-Wesley Publishing, 1996.
6. McCloghrie, K., and M. Rose. RFC 1213, "Management Information Base f Management Information of TCP/IP-based Internets: MIBII". March 1991.
7. Mills, D. RFC 1305, "Network Time Protocol (Version 3) Specification, Implementatio and Analysis". March 1992.
8. Mills, D. RFC 2030, "Simple Network Time Protocol (SNTP) Version 4 for IPv IPv6, and OSI". October 1996.
9. Postel, J., and J. Reynolds. RFC 854, "The Telnet Specification". May 1983.
10. Rigney, C. RFC 2866, "RADIUS Accounting". June 2000.
11. Rigney, C., S.Willens, A. Rubens, and W. Simpson. RFC 2865, "Remote Authenticati Dail-In User Service (RADIUS)". June 2000.

Глава 8

Ключевые темы этой главы

- Маршрутизатор в Куала-Лумпуре
- Маршрутизатор SF-1
- Маршрутизатор SF-2
- Маршрутизатор SF-Core-1
- Маршрутизатор SF-Core-2
- Маршрутизатор в Сан-Хосе
- Маршрутизатор Seoul -1
- Маршрутизатор Seoul-2
- Маршрутизатор в Сингапуре
- Сервер доступа SingISDN
- Сервер доступа Sing2511

Полная конфигурация ОС IOS для сети компании ZIP

В данной главе показаны полные наборы команд конфигурирования ОС IOS для всех маршрутизаторов и серверов доступа, стоящих в сети компании ZIP. Для демонстрации действующих на текущий момент команд конфигурирования устройства, работающего под управлением ОС IOS, используется команда режима EXEC show running-config.

При просмотре этих команд конфигурирования следует помнить следующее.

- В сети компании ZIP используется ОС IOS версии 12 1 Некоторые из команд конфигурирования ОС IOS могут работать не так, как описано в ранних версиях.
- Некоторые из устройств имеют неиспользованные интерфейсы, так как количество интерфейсов, необходимых для организации взаимодействия в сети компании ZIP, меньше, чем может быть сконфигурировано согласно спецификациям компании Cisco.
- Команды конфигурирования ОС IOS стоят не в том порядке, в котором они вводились в устройство
- Устройство разделяет некоторые основные сегменты команд конфигурирования символом восклицательного знака (') Все символы в конфигурационном файле, стоящие в данной строке после восклицательного знака, устройством ОС IOS не интерпретируются.
- В качестве протокола динамической маршрутизации для протоколов IP, Apple Talk и IPX все маршрутизаторы в сети компании ZIP используют протокол EIGRP.
- Для выполнения аутентификации, авторизации и учета устройства сети компании ZIP используют комбинацию протоколов RADIUS и TACACS+.
-

Маршрутизатор в Куала-Лумпуре

Устройство сети компании ZIP в Куала-Лумпуре представляет собой маршрутизатор модели Cisco 2501. Конфигурацию этого маршрутизатора отличает следующее.

- Сегмент локальной сети в Куала-Лумпуре подключен к интерфейсу Ethernet.
- Куала-Лумпур с маршрутизатором Seoul-1 соединяются по двухточечному интерфейсу Frame Relay.
- Для DHCP-клиентов в сегменте локальной сети в Куала-Лумпуре назначение IP-адресов выполняет DHCP-сервер ОС IOS.

Полная конфигурация маршрутизатора в Куала-Лумпуре выглядит следующим образом:

```
version  12.1
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
!
hostname Kuala-Lumpur
!
aaa new-model
aaa authentication login default group tacacs+ enable
aaa authorization exec group tacacs+ if-authenticated
aaa authorization network group radius if-authenticated
aaa acounting exe stop-only group tacacs+
enable secret 5 $2$5toY$IJQPTVD4.aEDLwZSnPrvX.
!
ip domain-list zipnet.com
ip domain-list zipnet.net
ip domain-name zipnet.com
ip name-server 131.108.110.34
ip name-server 131.108.110.35
ip dhcp database tftp://131.108.2.77/kl-dhcp-info
ip dhcp excluded-address 131.108.2.1 131.108.2.10
ip dhcp excluded-address 131.108.2.57
```

```
ip dhcp excluded-address 131.108.2.129 131.108.2.135
!
ip dhcp pool kl-common
    network 131.108.2.0/24
    dns-server 131.108.101.34 131.108.101.35
    domain-name zipnet. om
    netbios-name-server 131.108.21.70
    netbios-node-type h
    lease 0 1
!
ip dhcp pool kl-users
    network 131.108.2.0/25
   default-router 131.108.2.1
!
ip dhcp pool kl-users-2
   network 131.108.2.128/25
!
appletalk routing eigrp 25000
appletalk route-redistribution
routing 0000.Ob1.2 3e
clock timezone MST +8
!
interface Loopbackl
  description Kuala-Lumpur router loopback
  ip address 131.108.254.9 255.255.255.255
!
interface EthernetO
  description Kuala-Lumpur LAN Segment
  ip address 131.108.2.1 255.255.255.128

ntp broadcast
appletalk cable-range 3001-3010
appletalk zone Asia Manufacturing
ipx network 3010
!
interface SerialO
 description IETF frame relay PVCs on circuit M234563KL
 no ip address
encapsulation frame-relay ietf
 bandwidth 128
 frame-relay Imi-type ansi
!
interface SerialO.100 point-to-point
 description FR PVC 100 to Seoul-1
 ip address 131.108.242.2 255.255.255.252
 bandwidth 128
 frame-relay interface-dlci 100
 appletalk cable-range 2901-2901
 appletalk zone WAN Zone
 appletalk protocol eigrp
 no appletalk protocol rtmp
 ipx network 2901
!
interface Seriall
no ip address
 shutdown
 !
router eigrp 25000
 network 131.108.0.0
 no auto-summary
!
ip classless
logging trap debugging
logging console emergencies
logging 131.108.110.33
acess-list 1 permit 131.108.0.0 0.0.255.255
```

```
acess-list 2 permit host 131.108.20.45
!
ipx router eigrp 25000
 network 2901
 network 3010
tacacs-server host 131.108.110.33
tacacs-server key ZIPSecure
radius-server host 131.108.110.33
radius-server key Radius4Me
snmp-server community Zipnet RO 2
snmp-server community ZIPprivate RW 2
snmp-server host 131.108.20.45 Zipnet snmp frame-relay config
snmp-server location 1 KLCC Towers Kuala Lumpur Malaysia
snmp-server  contact Allan Leinwand allan@telegis.net
!
line con 0
 password 7 095B59
line aux 0
line vty 0 4
 password 7 095B59
 acess-class 1 in
 !
ntp update-calendar
ntp server 192.216.191.10
ntp server 129.189.134.11
!
end
```

Маршрутизатор SF-1

Устройство SF-1 сети компании ZIP представляет собой маршрутизатор модели Cisco 4700. Конфигурацию этого маршрутизатора отличает следующее.

- Магистральный сегмент локальной сети в Сан-Франциско подключается к интерфейсу Fast Ethernet.
- Сегмент локальной сети в Сан-Франциско подключается к интерфейсу Ethernet.
- На выходе в сегмент локальной сети в Сан-Франциско стоит IPX-фильтр GNS-сообщений.

Полная конфигурация маршрутизатора SF-1 выглядит следующим образом:

```
 version 12.1
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
!
hostname SF-1
!
aaa new-model
aaa authentication login default group tacacs+ enable
aaa authorization exe group tacacs+ if-authenticated
aaa authorization network group radius if-authenticated
aaa acounting exe stop-only group tacacs+
enable secret 5 $2$5toY$IJQPTVD4.aEDLwZSnPrvX.
!
ip domain-list zipnet.com
ip domain-list zipnet.net
ip domain-name zipnet. om
ip name-server 131.108.110.34
ip name-server 131.108.110.35
appletalk routing eigrp 25000
appletalk route-redistribution
ipx routing 0000.1 2 .23bb
!
clock timezone PST -8
```

```
clock sunnier-time PDT recurring
!
interface Loopbackl
 description SF-1 router loopback
 ip address 131.108.254.1 255.255.255.255
!
interface FastEthernetO
 description San Francisco FastEthernet backbone LAN
 ip address 131.108.20.1 255.255.252.0
 appletalk cable-range 1-10
 appletalk zone SF Zone
 ipx network 1010
 full-duplex
!
interface EthernetO
 description SF-1 LAN Segment
 ip address 131.108.101.1 255.255.255.0
 ip helper-address 131.108.21.70
 media-type lOBaseT
 ntp broadcast
 appletalk cable-range 11-100
 appletalk zone Operations
 ipx network 100
 ipx output-gns-filter 1010
!
interface Ethernetl
 no ip address
 shutdown
!
router eigrp 25000
 network 131.108.0.0
 no auto-summary
!
ip classless
logging 131.108.110.33
logging trap debugging
logging console emergencies
access-list 1 permit 131.108.0.0 0.0.255.255
access-list 2 permit host 131.108.20.45
access-list 1010 permit aa.0207.0104.0874
access-list 1010 deny -1
!
ipx router eigrp 25000
 network 100
 network 1010 i
tacacs-server host 131.108.110.33
tacacs-server key ZIPSecure
radius-server host 131.108.110.33
radius-server key Radius4Me
snmp-server community Zipnet RO 2
snmp-server community ZIPprivate RW 2
snmp-server host 131.108.20.45 Zipnet snmp frame-relay config
snmp-server location 22 Cable Car Drive, San Francisco, CA, USA
snmp-server contact Allan Leinwand allan@telegis.net
!
line  con 0
 password 7 095B59
line aux 0
line vty 0 4
 password 7 095B59
 access-class 1 in I
ntp update-calendar
ntp server 192.216.191.10
ntp server 129.189.134.11
!
```

```
end
```

Маршрутизатор SF-2

Устройство SF-2 сети компании ZIP представляет собой маршрутизатор модели Cisco 4700. Конфигурация этого маршрутизатора имеет такие характеристики.

- Магистральный сегмент локальной сети в Сан-Франциско подключается к интерфейсу Fast Ethernet.
- Два сегмента локальной сети в Сан-Франциско подключаются к двум интерфейсам Ethernet.
- На выходах в сегмент локальной сети в Сан-Франциско стоят IPX-фильтры GNS-сообщений.

Полная конфигурация маршрутизатора SF-2 выглядит следующим образом:

```
version 12.1
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
!
hostname SF-2
!
aaa new-model
aaa authentication login default group tacacs+ enable
aaa authorization exec group tacacs+ if-authenticated
aaa authorization network group radius if-authenticated
aaa accounting exec stop-only group tacacs+
enable secret 5 $2$5toY$IJQPTVD4.aEDLwZ8nPrvX.
i
ip domain-list zipnet.com
ip domain-list zipnet.net
ip domain-name zipnet.com
ip name-server 131.108.110.34
ip name-server 131.108.110.35
appletalk routing eigrp 25000
appletalk route-redistribution
ipx routing 0000.OcOc.llbb
!
clock timezone PST -8
clock summer-time PDT recurring
!
interface Loopbackl
 description SF-2 router loopback
 ip address 131.108.254.2 255.255.255.255
!
interface FastEthernetO
 description San Francisco FastEthernet backbone LAN
 ip address 131.Ю8.20.2 255.255.252.0
 appletalk cable-range 1-10
 appletalk zone SF Zone
 ipx network 10
!
interface EthernetO
 description SF-2 LAN Segment 1
 ip address 131.108.110.1 255.255.255.0
 ip helper-address 131.108.21.70
 media-type lOBaseT
 ntp broadcast
 appletalk cable-range 151-200
 appletalk zone Marketing
 ipx network 200
 ipx output-gns-filter 1010
!
interface Ethernetl
 description SF-2 LAN Segment 2
 ip address 131.108.120.1 255.255.255.0
 ip helper-address 131.108.21.70
```

```
 media-type lOBaseT
 ntp broadcast
 appletalk cable-range 101-150
 appletalk zone Sales
 ipx network 150
!
router eigrp 25000
 network 131.108.0.0
 no auto-summary
!
ip classless
logging 131.108.110.33
logging trap debugging
logging console emergencies
access-list 1 permit 131.108.0.0 0.0.255.255
access-list 2 permit host 131.108.20.45
access-list 1010 permit aa.0207.0104.0874
access-list 1010 deny -1
!
ipx router eigrp 25000
 network 10
 network 150
 network 200
!
tacacs-server host 131.108.110.33
tacacs-server key ZIPSecure
radius-server host 131.108.110.33
radius-server key Radius4Me
snmp-server community Zipnet RO 2
snmp-server  community ZIPprivate RW 2
snmp-server host 131.108.20.45 Zipnet snmp frame-relay config
snmp-server location 22 Cable Car Drive, San Francisco, CA, USA
snmp-server  contact Allan Leinwand, allan@telegis.net
!
line  con 0
 password 7 095B59
line aux 0
line vty 0 4
 password 7 095B59
 access-class 1 in
!
ntp update-calendar n
tp server 192.216.191.10
ntp server 129.189.134.11
!
end
```

Маршрутизатор SF-Core-1

Устройство SF-Core-1 сети компании ZIP представляет собой маршрутизатор модели Cisco 7505. Конфигурацию этого маршрутизатора отличает следующее.

- Магистральный сегмент локальной сети в Сан-Франциско подключается к интерфейсу Fast Ethernet.
- Используется последовательный HDLC-канал к маршрутизатору в Сан-Хосе.
- Существует последовательный HDLC-канал к провайдеру Internet для сети компании ZIP.
- Маршрутизаторы SF-Core-1 и SF-Core-2 образуют HSRP-группу.
- Между сетью компании ZIP и локальным провайдером Internet-сервиса-В используется маршрутизация с применением протокола EBGP. Маршруты, которые объявляются и принимаются в рамках протокола BOP, контролируются с помощью списков рассылки.
- В процессе EIGRP-маршрутизации для формирования маршрутов по умолчанию используется редистрибуция статических маршрутов.
- Для фильтрации трафика между сетью компании ZIP и общедоступной сетью Internet используется расширенный IP-список доступа.

• FastEthernet-сегмент по протоколу IPX подключается через SAP-фильтр.
Полная конфигурация маршрутизатора SF-Core-1 выглядит следующим образом:

```
Version 12.1
Service timestamps debug datetime localtime
Service timestamps log datetime localtime
Service password-encryption
!
hostname SF-Core-1
!
aaa   new-model
aaa   authentication login default group tacacs+ enable
aaa   authorization exec group tacacs+ if-authenticated
aaa   authorization network group radius if-authenticated
aaa   accounting exec stop-only group tacacs+
enable secret 5 $2$5toY$IJQPTVD4.aEDLwZSnPrvX.
!
ip tcp intercept mode watch
ip top intercept list 120
ip tcp intercept watch-timeout 15
ip domain-list zipnet.com
ip domain-list zipnet.net
ip domain-name zipnet. om
ip name-server 131.108.110.34
ip name-server 131.108.110.35
appletalk routing eigrp 25000
appletalk route-redistribution
ipx routing 0000.OeOd.lebO
!
lock timezone PST -8
lock summer-time PDT recurring!
interface Loopbackl
   description SF-Core-1 router loopback
   ip address 131.108.254.3 255.255.255.255
!
interface FastEthernetO/0
   description San Francisco FastEthernet backbone LAN
   ip address 131.108.20.3 255.255.252.0
   appletalk cable-range 1-10
   appletalk zone SF Zone
   ipx network 10
   standby ip 131.108.20.5
   standby preempt
   ipx router-sap-filter 1001
!
interface Seriall/0
   description HDLC leased line on circuit 456WS34209 to San-Jose
   ip address 131.108.240.1 255.255.255.252
   appletalk cable-range 901-901 appletalk zone WAN Zone
   appletalk protocol eigrp no
   appletalk protocol rtmp
   ipx network 901
!
   interface Seriall/1
   description HDLC leased line on circuit 789WS34256 to IS2-B
   ip address 192.7.2.2 255.255.255.252
   ip access-group 101 in
!
interface Seriall/2
   no ip address
   shutdown
!
interface Seriall/3 no ip address
   shutdown
!
```

```
router eigrp 25000
redistribute static
   redistribute bgp 25000 network 131.108.0.0
   distribute-list 1300 out
   no auto-summary
!
router bgp 25000
   no synchronization
   network 131.108.0.0
   neighbor 192.7.2.1 remote-as 1
   neighbor 192.7.2.1 description Internet Connection to ISP-B
   neighbor 192.7.2.1 distribute-list ISP-routes in
   neighbor 192.7.2.1 distribute-list ZIP-routes out
   remote-as 25000 description IBGP to Seoul-1 update-source Loopback 0
!
ip classless
ip default-network 131.119.0.0
ip default-network 140.222.0.0
ip route 131.108.232.0 255.255.255.0 FastEthernetO/0
ip route 131.108.0.0 255.255.0.0 NullO
logging 131.108.110.33
logging trap debugging
logging console emergencies
ip access-list standard ZIP-routes
permit 131.108.0.0
ip access-list standard ISP-routes
deny host 0.0.0.0 deny 127.0.0.0 0.255.255.255
deny 10.0.0.0 0.255.255.255
deny 172.16.0.0 0.15.255.255
deny 192.168.0.0 0.0.255.255
deny 192.0.2.0 0.0.0.255
deny 128.0.0.0 0.0.255.255
deny 191.255.0.0 0.0.255.
deny 192.0.0.0 0.0.0.255
deny 223.255.255.0 0.0.0.255
deny 224.0.0.0 31.255.255.255
permit any
access-list 1 permit 131.108.0.0 0.0.255.255
access-list 2 permit host 131.108.20.45
access-list 101 remark Permits NTP DNS WWW and SMTP
access-list 101 deny tcp host 192.7.2.2 host 192.7.2.2 log
access-list 101 deny ip 131.108.0.0 0.0.255.255 any log
access-list 101 deny ip 10.0.0.0 0.255.255.255 any
access-list 101 deny ip 172.16.0.0 0.15.255.255 any
access-list 101 deny ip 192.168.0.0 0.0.255.255 any
access-list 101 deny ip 127.0.0.0 0.255.255.255 any
access-list 101 permit ip host 192.7.2.1 host 192.7.2.2
access-list 101 deny ip any host 192.7.2.2
access-list 101 permit udp any 131.108.101.99 eq domain
access-list 101 permit udp host 15.255.160.64 host 131.108.254.3 eq ntp
access-list 101 permit udp host 128.4.1.1 host 131.108.254.3 eq ntp
access-list 101 permit udp host 16.1.0.4 host 131.108.254.3 eq ntp
access-list 101 permit udp host 204.123.2.5 host 131.108.254.3 eq ntp
access-list 101 permit tcp host 192.52.71.4 host 131.108.101.34 eq domain
access-list 101 permit tcp host 192.52.71.4 host 131.108.101.35 eq domain
access-list 101 permit tcp any host 131.108.101.34 eq smtp
access-list 101 permit tcp any host 131.108.101.35 eq smtp
access-list 101 permit tcp any host 131.108.101.100 eq www
access-list 101 permit tcp any host 131.108.101.100 eq ftp
access-list 101 permit tcp any host 131.108.101.100 eq ftp-data
access-list 101 permit tcp any gt 1023 host 131.108.101.100 gt 1023
access-list 101 permit icmp any any echo-reply
access-list 101 permit icmp any any time-exceeded
access-list 101 permit icmp any any port-unreachable
access-list 101 permit tcp any any established
access-list 101 permit tcp any any eq 22
```

```
access-list 101 deny  tcp any any eq ident
access-list 101 deny  ip any any log
access-list 120 permit ip any 131.108.0.0 0.0.255.255
access-list 1001 permit aa.0005.0112.0474
access-list 1001 deny -1
access-list 1300 permit 131.108.0.0 0.0.255.255
access-list 1300 permit 131.119.0.0
access-list 1300 pemit!40.222. 0. 0
!
ipx router eigrp 25000
  network 10
  network 901
!
tacacs-server host 131.108.110.33
tacacs-server key ZIPSecure
radius-server host 131.108.110.33
radius-server key Radius4Me
snmp-server community Zipnet RO 2
snmp-server community ZIPprivate RW 2
snmp-server host 131.108.20.45 Zipnet snmp frame-relay config
snmp-server location 22 Cable Car Drive, San Francisco,  CA, USA
snmp-server contact Allan Leinwand, allan@telegis.net
!
line con 0
 password 7 095B59
line aux 0
line vty 0 4
  password 7 095B59
  access-class 1 in
!
ntp update-calendar
ntp server 192.216.191.10
ntp server 129.189.134.11
!
end
```

Маршрутизатор SF-Core-2

Устройство SF-Core-2 сети компании ZIP представляет собой маршрутизатор модели Cisco 7505. Конфигурацию этого маршрутизатора отличает следующее.

- Магистральный сегмент локальной сети в Сан-Франциско подключается к интерфейсу Fast Ethernet.
- Используется последовательный HDLC-канал к маршрутизатору в Сан-Хосе.
- Маршрутизаторы SF-Core-1 и SF-Core-2 образуют HSRP-rpynny.
- В процессе EIGRP-маршрутизации для формирования маршрутов по умолчанию используется редистрибуция статических маршрутов.
- FastEthernet-сегмент по протоколу IPX подключается через SAP-фильтр.

Полная конфигурация маршрутизатора SF-Core-2 выглядит следующим образом:

```
version 12.1
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
!
hostname SF-Core-2
!
aaa new-model
aaa authentication login default group tacacs+ enable
aaa authorization exec  group tacacs+ if-authenticated
aaa authorization network group radius if-authenticated
```

```
aaa accounting exec  stop-only group tacacs+
enable secret 5 $2$5toY$IJQPTVD4.aEDLwZSnPrvX.
!
ip domain-list zipnet.com
ip domain-list zipnet.net
ip domain-name zipnet.com
ip name-server 131.108.110.34
ip name-server 131.108.110.35
appletalk routing eigrp 25000
appletalk route-redistribution
ipx routing OOOO.cOc.OlOb
!
clock timezone PST -8
clock summer-time PDT recurring
!
interface Loopbackl
 description SF-Core-2 router loopback
 ip address 131.108.254.4 255.255.255.255
!
interface FastEthernetO/0
description San Francisco FastEthernet backbone LAN
 ip address 131.108.20.4 255.255.252.0
 appletalk  able-range 1-10
 appletalk zone SF Zone
 ipx network 10
 standby ip 131.108.20.5
 standby preempt
 ipx router-sap-filter 1001
!
interface Seriall/0
 description HDLC leased line on circuit WSZ02980189 to Seoul-2
 ip address 131.108.240.5 255.255.255.252
 appletalk cable-range 902-902
 appletalk zone WAN Zone
 appletalk protocol eigrp
 no appletalk protocol rtmp
 ipx network 902
!
interface Seriall/1
 no ip address
 shutdown
!
interface Seriall/2
 no ip address
 shutdown
!
interface Seriall/3
 no ip address
 shutdown
router eigrp  25000
 redistribute static
 network 131.108.0.0
 no auto-summary
!
ip  classless
ip route 131.108.0.0 255.255.0.0 NullO
logging 131.108.110.33
logging trap  debugging
logging  console emergencies
access-list 1 permit 131.108.0.0 0.0.255.255
access-list 2 permit host 131.108.20.45
access-list 1001 permit aa.0005.0112.0474
access-list 1001 deny -1
!
ipx router eigrp 25000
 network 10
```

```
 network 902
!
tacacs-server host 131.108.110.33
tacacs-server key ZIPSecure
radius-server host 131.108.110.33
radius-server key Radius4Me
snmp-server   community Zipnet RO 2
snmp-server   community ZIPprivate RW 2
snmp-server host 131.108.20.45 Zipnet snmp frame-relay  config
snmp-server location 22 Cable Car Drive, San Francisco,  CA, USA
snmp-server   contact Allan Leinwand, allan@telegis.net
!
line  con 0
 password 7 095B59
line aux 0
line vty 0 4
 password 7 095B59
 access-class 1 in
!
ntp update-calendar
ntp server 192.216.191.10
ntp server 129.189.134.11
!
end
```

Маршрутизатор в Сан-Хосе

Устройство сети компании ZIP в Сан-Хосе представляет собой маршрутизатор модели Cisco 3640. Конфигурацию этого маршрутизатора отличает следующее.

- Сегмент локальной сети в Сан-Хосе подключается к интерфейсу Token Ring с полосой 16 Мбайт.
- Используется последовательный HDLC-канал к маршрутизатору SF-Core-1.
- Применяется последовательный HDLC-канал к маршрутизатору Seoul-1.
- Чтобы разрешить трафик к общедоступной части зоны технического дивизиона по протоколу AppleTalk, применяется список доступа.
- В целях объявления доступа к IPX-серверу общего пользования технического дивизиона последовательные каналы подключаются через выходной SAP-фильтр протокола IPX.

Полная конфигурация маршрутизатора в Сан-Хосе выглядит следующим образом:

```
version 12.1
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
!
hostname San-Jose
!
aaa new-model
aaa authentication login default group tacacs+ enable
aaa authorization exec group tacacs+ if-authenticated
aaa authorization network group radius if-authenticated
aaa accounting exec stop-only group tacacs+
enable secret 5 $2$5toY$IJQPTVD4.aEDLwZSnPrvX.
!
ip domain-list zipnet.com
ip domain-list zipnet.net
ip domain-name zipnet.com
ip name-server 131.108.110.34
ip name-server 131.108.110.35
appletalk routing eigrp 25000
appletalk route-redistribution
```

```
ipx routing OOOO.clOe.lOOd
!
clock timezone PST -8
clock summer-time PDT recurring
i
interface Loopbackl
 description San-Jose router loopback
 ip address 131.108.254.4 255.255.255.255
!
interface TokenRingO/0
 no ip address
 shutdown
!
interface SerialO/0
 description HDLC leased line on circuit BCS20198ASL to SF-Core-1
 ip address 131.108.240.2 255.255.255.252
 appletalk  cable-range 901-901
 appletalk zone WAN Zone
 appletalk protocol eigrp
 no appletalk protocol rtmp
 ipx network 901
 ipx output-sap-filter 1000
 appletalk access-group 601
!
interface SerialO/1
 no ip address
 shutdown
!
interface TokenRingl/0
 description San Jose LAN Segment
 ip address 131.108.100.1 255.255.255.128
 ip helper-address 131.108.21.70
 ring-speed 16
 early-token-release
 ntp broadcast
 appletalk cable-range 1001-1010
 appletalk zone Engineering
 ipx network 1010
!
interface Seriall/0
description HDLC leased line on circuit BCS1014343-9901 to Seoul-1
 ip address 131.108.241.2 255.255.255.252
 appletalk cable-range 1901-1901
 appletalk zone WAN Zone
 appletalk protocol eigrp
 no appletalk protocol rtmp
 ipx network 1901
 ipx output-sap-filter 1000
 appletalk access-group 601
!
interface Seriall/1
no ip address
 shutdown
!
router eigrp 25000
network 131.108.0.0
no auto-suiranary
!
ip  classless
logging 131.108.110.33
logging trap debugging
logging  console emergencies
access-list 1 permit 131.108.0.0  0.0.255.255
access-list 2 permit host 131.108.20.45
access-list 601 permit nbp 1 object Engineering Public
access-list 601 permit nbp 1 type AFPServer
```

```
access-list 601 permit nbp 1 zone San Jose Zone
access-list 601 deny other-nbps
access-list 1000 permit 10.0000.0000.aObO
access-list 1000 deny -1
!
ipx router eigrp 25000
 network 901
 network 1010
 network 1901
!
tacacs-server host 131.108.110.33
tacacs-server key ZIPSecure
radius-server host 131.108.110.33
radius-server key Radius4Me
snmp-server community Zipnet RO 2
snmp-server community ZIPprivate RW 2
snmp-server host 131.108.20.45 Zipnet snmp frame-relay  config
snmp-server location 20 Market Street, San Jose, CA, USA
snmp-server contact Allan Leinwand, allan@telegis.net
!
line con 0
 password 7 095B59
line aux 0
line vty 0  4
 password 7 095B59
 access-class  1 in
!
ntp update- clendar
ntp server  192.216.191.10
ntp server  129.189.134.11
!
end
```

Маршрутизатор Seoul-1

Устройство Seoul-1 сети компании ZIP представляет собой маршрутизатор модели Cisco 4700. Конфигурацию этого маршрутизатора отличает следующее.

- Сегмент локальной сети в Сеуле подключается к интерфейсу Ethernet. В этом сегменте за счет HSRP-групп организовано резервное дублирование.
- Маршрутизатор в Сингапуре и маршрутизатор в Куала-Лумпуре подключаются через двухточечный интерфейс Frame Relay.

Полная конфигурация маршрутизатора Seoul-1 выглядит так:

```
version 12.1
service timestamps debug datetime  localtime
service timestamps log datetime localtime
service password-encryption

hostname Seoul-1
!
aaa new-model
aaa authentication login default group tacacs+  enable
aaa authorization exec group tacacs+ if-authenticated
aaa authorization network group radius if-authenticated
aaa accounting exec   stop-only group tacacs+
enable secret  5 $2$5toY$IJQPTVD4.aEDLwZSnPrvX.
!
ip tcp intercept mode watch
ip tcp intercept list 120
ip tcp intercept watch-timeout 15
ip domain-list zipnet.com
```

```
ip domain-list zipnet.net
ip domain-name zipnet.com
ip name-server 131.108.110.34
ip name-server 131.108.110.35
appletalk routing eigrp 25000
appletalk route-redistribution
ipx routing 0000.0011.bceb
!
clock timezone KST ₸9
!
interface Loopbackl
 description Seoul-1  router loopback
 ip address 131.108.254.6 255.255.255.255
!
interface EthernetO
 description Seoul LAN Segment
 ip address 131.108.3.1 255.255.255.128
 ip helper-address 131.108.21.70
 no ip redirects
 media-type lOBaseT
 ntp broadcast
 appletalk  cable-range 2001-2010
 appletalk zone Asia Distribution
 ipx network 2010
 standby 1 ip 131.108.3.3
 standby 1 priority 100
 standby 1 track Seriall
 standby 1 preempt
 standby 2 ip 131.108.3.4
 standby 2 priority 95
 standby 2 preempt
!
interface SerialO
 description IETF frame relay PVCs on  circuit S123789y
 no ip address
 encapsulation frame-relay ietf
 bandwidth 256
 frame-relay Imi-type ansi
!
interface SerialO.16 point-to-point
 description FR PVC 16 to Kuala-Lumpur
 ip address 131.108.242.1 255.255.255.252
 bandwidth 128
 frame-relay interface-dlci 16
 appletalk  cable-range 2901-2901
 appletalk zone WAN Zone
 appletalk protocol eigrp
 no appletalk protocol rtmp
ipx network 2901
!
interface SerialO.17 point-to-point
 description FR PVC 17 to Singapore
 ip address 131.108.242.5 255.255.255.252
 bandwidth 128
 frame-relay interface-dlci 17
 appletalk  cable-range 2902-2902
 appletalk zone WAN Zone
 appletalk protocol eigrp
 no appletalk protocol rtmp
 ipx network 2902
!
interface Seriall
 description HDLC leased line on  circuit MC23-01-KL889 to San Jose
 ip address 131.108.241.2 255.255.255.252
 appletalk  cable-range 1901-1901
 appletalk zone WAN Zone
```

```
 appletalk protocol eigrp
 no appletalk protocol rtmp
 ipx network 1901
!
interface Serial2
 description HDLC leased line on  circuit ZW2390-1-H to ISP-A
 ip address 211.21.2.2 255.255.255.252
 ip access-group 101 in
interface Serial3
 no ip address
 shutdown
!
router eigrp 25000
 redistribute bgp 25000 network 131.108.0.0
 distribute-list 1300 out
 no auto-summary
!
router bgp 25000
 no synchronization
 network 131.108.0.0
 neighbor 211.21.2.1 remote-as 701
 neighbor 211.21.2.1 description Internet Connection to ISP-A
 neighbor 211.21.2.1 distribute-list ISP-routes  in
 neighbor 211.21.2.1 distribute-list ZIP-routes  out
 neighbor 131.108.254.3 remote-as 25000
 neighbor 131.108.254.3 des ription IBGP to SF-Core-1
 neighbor 131.108.254.3 update-source Loopback 0
!
ip  classless
logging 131.108.110.33
logging trap debugging
logging console emergencies
ip access-list standard ZIP-routes
 permit 131.108.0.0
ip access-list standard ISP-routes
 deny host 0.0.0.0
 deny 127.0.0.0 0.255.255.255
 deny 10.0.0.0 0.255.255.255
 deny 172.16.0.0 0.15.255.255
 deny 192.168.0.0 0.0.255.255
 deny 192.0.2.0 0.0.0.255
 deny 128.0.0.0 0.0.255.255
 deny 191.255.0.0 0.0.255.
 deny 192.0.0.0 0.0.0.255
 deny 223.255.255.0 0.0.0.255
deny 224.0.0.0 31.255.255.255
 permit any
access-list 1 permit 131.108.0.0  0.0.255.255
access-list 2 permit host 131.108.20.45
access-list 101 remark Permits NTP  DNS  WWW  and SMTP
access-list 101 deny tcp host 192.7.2.2 host 192.7.2.2 log
access-list 101 deny ip 131.108.0.0 0.0.255.255  any  log
access-list 101 deny ip 10.0.0.0 0.255.255.255 any
access-list 101 deny ip 172.16.0.0 0.15.255.255  any
access-list 101 deny ip 192.168.0.0 0.0.255.255  any
access-list 101 deny ip 127.0.0.0 0.255.255.255  any
access-list 101 permit ip host 192.7.2.1 host 192.7.2.2
access-list 101 deny ip any host 192.7.2.2
access-list 101 permit udp any 131.108.101.99 eq domain
access-list 101 permit udp host 15.255.160.64 host 131.108.254.3  eq ntp
access-list 101 permit udp host 128.4.1.1 host 131.108.254.3 eq ntp
access-list 101 permit udp host 16.1.0.4 host 131.108.254.3 eq ntp
access-list 101 permit udp host 204.123.2.5 host 131.108.254.3 eq ntp
access-list 101 permit tcp host 192.52.71.4 host 131.108.101.34 eq domain
access-list 101 permit tcp host 192.52.71.4 host 131.108.101.35 eq domain
access-list  101 permit tcp any host 131.108.101.34 eq smtp
```

```
access-list  101 permit tcp any host 131.108.101.35 eq smtp
access-list  101 permit tcp any host 131.108.101.100 eq www
access-list  101 permit tcp any host 131.108.101.100 eq ftp
access-list  101 permit tcp any host 131.108.101.100 eq ftp-data
access-list  101 permit tcp any gt 1023 host  131.108.101.100 gt 1023
access-list  101 permit icmp any any echo-reply
access-list  101 permit icmp any any time-exceeded
access-list  101 permit icmp any any port-unreachable
access-list  101 permit tcp any any established
access-list  101 permit tcp any any eq 22
access-list  101 deny tcp any any eq ident
access-list  101 deny ip any any log access-list  120 permit ip any
131.108.0.0
0.0.255.255
access-list  1300 permit 131.108.0.0 0.0.255.255
access-list  1300 permit 131.119.0.0
access-list  1300 permit 140.222.0.0
!
ipx router eigrp 25000
 network 1901
 network 2010
 network 2901
!
tacacs-server  host  131.108.110.33
tacacs-server  key ZIPSecure
radius-server  host  131.108.110.33
radius-server  key Radius4Me
snmp-server  community Zipnet RO 2
snmp-server  community ZIPprivate RW 2
snmp-server  host 131.108.20.45 Zipnet snmp frame-relay  config
snmp-server  location 251 Second Street, Seoul, Korea
snmp-server  contact Allan Leinwand, allan@telegis.net
!
line  con 0
 password 7  095B59
line aux 0
line vty 0 4
 password 7  095B59
 access-class  1 in
!
ntp update-  calendar
ntp server 192.216.191.10
ntp server 129.189.134.11
!
end
```

Маршрутизатор Seoul-2

Устройство Seoul-2 сети компании ZIP представляет собой маршрутизатор модели Cisco 4700. Конфигурация этого маршрутизатора имеет такие характеристики.

- Сегмент локальной сети в Сеуле подключается к интерфейсу Ethernet. В этом сегменте за счет HSRP-rpyrm организовано резервное дублирование.
- Используется последовательный HDLC-канал к маршрутизатору SF-Core-2.

Полная конфигурация маршрутизатора Seoul-2 выглядит следующим образом:

```
version 12.1
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
!
hostname Seoul-2
```

```
!
aaa new-model
aaa authentication login default group tacacs+ enable
aaa authorization exec group tacacs+ if-authenticated
aaa authorization network group radius if-authenticated
aaa accounting exec stop-only group tacacs+
enable secret 5 $2$5toY$IJQPTVD4.aEDLwZSnPrvX.
!
ip domain-list zipnet.com
ip domain-list zipnet.net
ip domain-name zipnet.com
ip name-server 131.108.110.34
ip name-server 131.108.110.35
appletalk routing eigrp 25000
appletalk route-redistribution
ipx routing 0000.d ec.elbO
!
clock timezone KST +9
!
interface Loopbackl
 description Seoul-2 router loopback
 ip address 131.108.254.7 255.255.255.255
!
interface EthernetO
 description Seoul LAN Segment
 ip address 131.108.3.2 255.255.255.128
 ip helper-address 131.108.21.70
 no ip redirects
 media-type lOBaseT
 ntp broadcast
 appletalk  cable-range 2001-2010
 appletalk zone Asia Distribution
 ipx network 2010
 standby 1 priority 95
 standby 1 preempt
 standby 1 ip 131.108.3.3
 standby 2 priority 100
 standby 2 track SerialO
 standby 2 preempt
 standby 2 ip 131.108.3.4
!
interface SerialO
 description HDLC leased line on  circuit ZW983800-03  to SF-Core-2
 ip address 131.108.240.6 255.255.255.252
 appletalk cable-range 902-902
 appletalk zone WAN Zone
appletalk protocol eigrp
 no appletalk protocol rtmp
 ipx network 902
!
interface  Seriall
 no ip address
 shutdown
!
router eigrp 25000
 network 131.108.0.0
 no auto-summary
!
ip  classless
logging 131.108.110.33
logging trap debugging
logging console emergencies
access-list I permit 131.108.0.0  0.0.255.255
access-list 2 permit host 131.108.20.45
!
ipx router eigrp 25000
```

```
 network 902
 network 2010
!
ta cacs-server host 131.108.110.33
tacacs-server key ZIPSecure
radius-server host 131.108.110.33
radius-server key Radius4Me
snmp-server  community Zipnet  RO 2
snmp-server  community ZIPprivate RW 2
snmp-server host 131.108.20.45 Zipnet snmp frame-relay  config
snmp-server location 251 Second Street, Seoul,  Korea
snmp-server  contact Allan Leinwand, allan@telegis.net
!
line  con  0
 password  7 095B59
line aux 0
line vty 0 4
 password  7 095B59
 access-class 1 in
!
ntp update-calendar
ntp server 192.216.191.10
ntp server 129.189.134.11
!
end
```

Маршрутизатор в Сингапуре

Устройство сети компании ZIP в Сингапуре представляет собой маршрутизатор модели Cisco 2501. Конфигурацию этого маршрутизатора отличает следующее.

- Сегмент локальной сети в Сингапуре подключается к интерфейсу Ethernet.
- Для связи с маршрутизатором Seoul-1 используется двухточечный интерфейс Frame Relay.
- В сегменте локальной сети в Сингапуре используется протокол RIP и выполняется редистрибуция маршрутной информации протокола EIGRP в протокол RIP.

Полная конфигурация маршрутизатора в Сингапуре выглядит следующим образом

```
version 12.1
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
!
hostname Singapore
!
aaa new-model
aaa authentication login default group tacacs+ enable
aaa authorization exec group tacacs+ if-authenticated
aaa authorization network group radius if-authenticated
aaa accounting exec  stop-only group tacacs+
enable secret 5 $2$5toY$IJQPTVD4.aEDLwZSnPrvX.
!
ip domain-list zipnet.com
ip domain-list zipnet.net
ip domain-name zipnet.com
ip name-server 131.108.110.34
ip name-server 131.108.110.35
appletalk routing eigrp 25000
appletalk route-redistribution
ipx routing 0000.ceec.eebb
!
clock timezone SST +8
```

```
i
interface Loopbackl
 description Singapore router loopback
 ip address 131.108.254.8 255.255.255.255
!
interface EthernetO
 description Singapore LAN Segment
 ip address 131.108.1.1 255.255.255.128
 ip helper-address 131.108.21.70
 ntp broadcast
 appletalk  able-range 4001-4010
 appletalk zone Asia Manufacturing
 ipx network 4010
!
interface SerialO
 description IETF frame relay PVCs on Circuit Z-234987-12-MS-01
 no ip address
 encapsulation frame-relay ietf
 bandwidth 128
 frame-relay Imi-type ansi
!
interface SerialO.100 point-to-point
 description FR PVC 100 to Seoul-1
 ip address 131.108.242.6 255.255.255.252
 bandwidth 128
 frame-relay interface-dl i 100
 appletalk  able-range 2902-2902
 appletalk zone WAN Zone
 appletalk protocol eigrp
 no appletalk protocol rtmp
 ipx network 2902
!
interface Seriall
 no ip address
 shutdown
!
router eigrp  25000
 network 131.108.0.0
no auto-summary
!
router rip
 redistribute eigrp 25000
 passive-interface SerialO.100
 network 131.108.0.0
 default-metric 3
!
ip  classless
logging trap  debugging
logging  console emergencies
logging 131.108.110.33
access-list 1 permit 131.108.0.0 0.0.255.255
access-list 2 permit host 131.108.20.45
!
ipx router eigrp 25000
 network 4010
 network 2902
!
tacacs-server host 131.108.110.33
tacacs-server key ZIPSecure
radius-server host 131.108.110.33
radius-server key Radius4Me
snmp-server   community Zipnet RO 2
snmp-server   community ZIPprivate RW 2
snmp-server location 1 Raffles Place, Singapore
snmp-server   contact Allan Leinwand,  allan@telegis.net
!
```

```
line  on 0
 password 7 095B59
line aux 0
line vty 0 4
 password 1 095B59
 access-class 1 in
!
ntp update-calendar
ntp server 192.216.191.10
ntp server 129.189.134.11
!
end
```

Сервер доступа SinglSDN

Устройство SinglSDN в сети компании ZIP представляет собой сервер доступа модели Cisco 4500. Этот сервер доступа имеет установки, позволяющие обслуживать удаленных клиентов ISDN по коммутируемым каналам связи с использованием протокола IP, и может быть взят в качестве эталона для любого подобного сервера доступа в сети компании ZIP. Необходимо, чтобы каждый сервер доступа имел уникальный адрес сетевого уровня, так как в сетевом комплексе не допускается наличие дублирующих адресов сетевого уровня. Этот сервер имеет следующую конфигурацию.

- Обеспечен удаленный доступ для IP-клиентов по коммутируемым каналам связи в ISDN-сети.
- Наличие локальной базы данных имен пользователей для их аутентификации.
- Наличие ISDN-группы из четырех ISDN BRI-интерфейсов.
- Интерфейс вызова по номеру выполняет аутентификацию PPP-сеансов с использованием протоколов PAP, CHAP и MS-CHAP.
- Список доступа разрешает только конкретный IP-трафик, поддерживающий текущий телефонный звонок.

Полная конфигурация сервера доступа SinglSDN выглядит так:

```
version 12.1
service timestamps debug datetime  localtime
service timestamps log datetime localtime
service password-encryption
!
hostname SinglSDN
!
aaa new-model
aaa authentication login default group tacacs+  local
aaa authentication ppp default group  tacacs+ local
aaa authentication arap default local
aaa authorization exec local group tacacs+ if-authenticated
aaa authorization network local group radius if-authenticated
aaa accounting exec stop-only group  tacacs+
aaa accounting network stop-only group tacacs+
enable secret 5 $2$5toY$IJQPTVD4.aEDLwZ8nPrvX.
!
username jim password 7 53633635
username Janet password 7 878743465
ip domain-list zipnet.com
ip domain-list zipnet.net
ip domain-name zipnet.com
ip name-server 131.108.110.34
ip name-server 131.108.110.35
ip address-pool local
async-bootp dns-server 131.108.101.34 131.108.101.35
async-bootp nbns-server 131.108.21.70
```

```
isdn switch-type basic-dmslOO
!
clock timezone SST +8
!
interface LoopbackO
 ip address  131.108.254.11 255.255.255.255
!
interface EthernetO
 description Singapore User LAN
 ip address  131.108.1.81 255.255.255.128
 media-type  lOBaseT
!
interface BRIO
 no ip address
 encapsulation ppp
 isdn spidl  98050101
 isdn spid2  98060101
 isdn answerl 50101
 isdn answer2 60101
 dialer rotary-group 1
!
interface BRI1
 no ip address
 encapsulation ppp
 isdn spidl 98070101
 isdn spid2 98080101
 isdn answerl 70101
 isdn answer2 80101
 dialer rotary-group 1
!
interface BRI2
 no ip address
 encapsulation ppp
 isdn spidl 91470102
 isdn spid2 91490102
 isdn answerl 70102
 isdn answer2 90102
 dialer rotary-group 1
!
interface BRI3
 no ip address
 encapsulation ppp
 isdn spidl 91350102
 isdn spid2 91390102
 isdn answerl 50102
 isdn answer2 90102
 dialer rotary-group 1
!
interface Ethernetl
 no ip address
 shutdown
i
interface Dialerl
 description Singapore ISDN Dialup Pool
 ip unnumbered EthernetO
 encapsulation ppp
 peer default ip address pool isdn-users
 dialer in-band
 dialer idle-timeout 300
 dialer-group 1
 ppp authentication chap ms-chap pap  call-in
 ppp multiljnk
  compress mpcc
!
router eigrp 25000
 network 131.108.0.0
```

```
 no auto-summary
!
ip local pool isdn-users 131.108.1.91 131.108.1.106
ip  classless
logging trap debugging
logging 131.108.110.33
access-list 1 permit 131.108.0.0 0.0.255.255
access-list 2 permit host 131.108.20.45
access-list 102 permit tcp any any eq telnet
access-list 102 permit tcp any any eq www
access-list 102 permit udp any any eq domain
access-list 102 permit tcp any any eq ftp
tacacs-server host 131.108.110.33
tacacs-server key ZIPSecure
radius-server host 131.108.110.33
radius-server key Radius4Me
snmp-server community Zipnet RO 2
snmp-server community ZIPprivate RW 2
snmp-server host 131.108.20.45 Zipnet snmp  frame-relay isdn  config
snmp-server location 1 Raffles Place,  Singapore
snmp-server  contact Allan Leinwand,  allan@telegis.net
dialer-list 1 protocol ip list 102
!
line  con 0
 password 7 052C092B284B47
line aux 0
password 7 095B59
line vty 0 4
password 7 095B59
 access-class 1 in
!
ntp  clock-period 17179886
ntp server 192.216.191.10
ntp server 129.189.134.11
end
```

Сервер доступа Sing2511

Устройство Sing2511 в сети компании ZIP представляет собой сервер доступа модели Cisco 2511. Этот сервер имеет установки, позволяющие обслуживать удаленных пользователей IP, AppleTalk и IPX по коммутируемым каналам связи с аналоговыми сигналами. Эта конфигурация может быть взята в качестве шаблона для любого подобного сервера доступа в сети компании ZIP. Необходимо, чтобы каждый сервер доступа имел уникальный адрес сетевого уровня, так как в сетевом комплексе не допускается наличие дублирующих адресов сетевого уровня. Конфигурацию этого сервера отличает следующее.

- Для использующих коммутируемые каналы связи удаленных пользователей доступ по протоколам IP, AppleTalk и IPX.
- Резервная локальная база данных имен пользователей для их аутентификации.
- Пул локальных IP-адресов для удаленных IP-пользователей, работающих с коммутируемыми каналами связи.
- Для конфигурирования 16 асинхронных терминальных каналов используется групповой асинхронный интерфейс.
- Конфигурация терминальных каналов учитывает специфические для модемов параметры и временные пределы ожидания в случае отсутствия действий во время пользовательских сеансов.

Полная конфигурация сервера доступа Sing2511 выглядит следующим образом:

```
version 12.1
service timestamps debug datetime  localtime
service timestamps log datetime localtime
service password-encryption
```

```
!
hostname Sing2511
!
aaa new-model
aaa authentication login default group tacacs+ local
aaa authentication ppp default group tacacs+ local
aaa authentication arap default auth-guest local
aaa authorization exec local group tacacs+ if-authenticated
aaa authorization network local group radius if-authenticated
aaa accounting exec stop-only group tacacs+
aaa accounting network stop-only group tacacs+
enable secret 5 $2$5toY$IJQPTVD4.aEDLwZSnPrvX.
!
username John password 7 15140403446A
username jane password 7 121B0405
ip domain-list zipnet.com
ip domain-list zipnet.net
ip domain-name zipnet.com
ip name-server 131.108.110.34
ip name-server 131.108.110.35
ip address-pool local
appletalk routing eigrp 25000
appletalk route-redistribution
appletalk virtual-net 3000 Ma -dialup
arap network 2500 Mac-dialup
!
clock timezone SST +8
!
interface LoopbackO
 ip address 131.108.254.10 255.255.255.255
 ipx netwrok 2500
!
interface EthernetO
 description Singapore User LAN
 ip address 131.108.1.80 255.255.255.128
appletalk cable-range 4001-4010
 appletalk zone Asia Manufacturing
 ipx network 4010
!
interface SerialO
 no ip address
 shutdown
!
interface Seriall
 no ip address
 shutdown
!
interface Group-Async 1
 description dialup pool on Singapore  2511
 ip unnumbered EthernetO
 encapsulation ppp
 async mode  interactive
 appletalk client-mode
 ipx ppp-client  Loopback 0
 ipx update  interval  rip 36000
 ipx update  interval  sap 36000
 peer default  ip address pool modem-users
 ppp authentication pap ms-chap chap call-in
 ppp ipcp dns 131.108.101.34 131.108.101.35
 ppp ipcp wins 131.108.21.70
 compress mpcc
 group-range 1 16
!
router eigrp 25000
 network 131.108.0.0
 no auto-summary
```

```
!
ip local pool modem-users 131.108.1.111 131.108.1.126
ip classless
logging trap debugging
logging 131.108.110.33
access-list 1 permit 131.108.0.0 0.0.255.255
access-list 2 permit host 131.108.20.45
!
ipx router eigrp 25000
 network 25000
 network 4010
!
tacacs-server host 131.108.110.33
tacacs-server key ZIPSecure
radius-server host 131.108.110.33
radius-server key Radius4Me
snmp-server community Zipnet RO 2
snmp-server community ZIPprivate RW 2
snmp-server host 131.108.20.45 Zipnet snmp frame-relay  config
snmp-server location 1 Raffles Place, Singapore
snmp-server  ontact Allan Leinwand, allan@telegis.net
!
line  con 0
 password 7 052C092B284B47
line 1 16
 session-timeout 30
 autoselect arap
 autoselect during-login
 autoselect ppp
 arap enable
 arap authentication default
 session-dis onne t-warning 60
 login authentication default
modem Dialin
 modem auto configure type usr_courier
 stopbits 1
 rxspeed 115200
 txspeed 115200
 flow control hardware
line aux 0
password 7 095B59
line vty 0 4
password 7 095B59
 access-class 1 in
!
ntp  clock-period 17179886
ntp server 192.216.191.10
ntp server 129.189.134.11
end
```

Резюме

В этой главе представлены полные наборы команд конфигурирования ОС IOS маршрутизаторов и серверов доступа сети компании ZIP. Эти команды отражают практическое приложение концепций, изложенных в настоящей книге.