Администрирование сетей Cisco: освоение за месяц

Сверхнадежные маршрутизаторы и коммутаторы компании Cisco верой и правдой служат в миллионах сетей, но принцип «установил и забыл» с ними не сработает. К счастью, вам не нужно быть сертифицированным администратором с многолетним опытом, чтобы развернуть и поддерживать сеть Cisco. С набором удобных технологий, небольшой практикой и этой книгой вы сможете содержать систему в полном порядке.

Научитесь администрировать сети Cisco за месяц!

Эта книга написана для фрилансеров и штатных сетевых администраторов, пользующихся оборудованием Cisco. За 22 подробные главы вы получите практические знания по настройке сети Cisco и обеспечению ее бесперебойной работы. Реальная практика начинается с настройки коммутатора и ведет вас через основные команды, протоколы, аспекты динамической маршрутизации и многое, многое другое.

Вы изучите:

- технологии сетей Cisco, включая разницу между коммутаторами и маршрутизаторами;
- настройку виртуальных локальных сетей (VLAN) и транков;
- обеспечение безопасности сети;
- подключение и настройку маршрутизаторов и коммутаторов;
- поддержку сети в рабочем состоянии.

Бен Пайпер (benpiper.com) — практикующий ИТ-консультант, имеющий многочисленные сертификаты Cisco, Citrix и Microsoft, включая Cisco CCNA и CCNP.
Он записал множество видеоуроков по работе с сетями, подготовке к сертификации Cisco CCNP, системам Puppet и управлению Windows Server.

Эта книга написана для читателей, не имеющих опыта работы с сетями Cisco.

Интернет-магазин: www.dmkpress.com
Книга – почтой: orders@alians-kniga.ru
Оптовая продажа: “Альянс-книга”
тел.(499)782-3889. books@alians-kniga.ru

ДМК издательство www.дмк.рф

ISBN 978-5-97060-519-6

«Это одна из лучших книг по теме, которые я когда-либо читал. Бен знает, о чем говорит и, что более важно, может этому научить.»
— Кент Р. Спиллнер, компания DRW

«Это та книга по сетевому окружению Cisco, которую мы так долго ждали. Настоятельно рекомендую прочитать, если вы хотите администрировать сети Cisco.»
— Марк Фурман, компания Info-Link Technologies

«Прекрасная книга, чтобы начать и продолжить без остановки.»
— Сау Фаи Фонг, компания Panda Tech Hub

«Если вы только хотите сдать экзамен – выберите другую книгу. Если вы хотите настроить свою рабочую сеть Cisco – это то, что надо.»
— Дэвид Кернс, компания Rincon Research

Администрирование сетей Cisco: освоение за месяц

Администрирование сетей Cisco: освоение за месяц

Бен Пайпер

ДМК издательство
MANNING

Бен Пайпер

Администрирование сетей Cisco: освоение за месяц

Ben Piper

Learn Cisco Network Administration in a Month of Lunches

MANNING
SHELTER ISLAND

Бен Пайпер

Администрирование сетей Cisco: освоение за месяц

ДМК Пресс
издательство
Москва, 2018

УДК 004.71
ББК 32.972.5
П12

Пайпер Б.
П12 Администрирование сетей Cisco: освоение за месяц / пер. с анг. М. А. Райтмана. – М.: ДМК Пресс, 2018. – 316 с.: ил.

ISBN 978-5-94074-519-6

Эта книга в доступной форме рассказывает об администрировании сетей с применением оборудования Cisco. С помощью практических заданий вы сможете за месяц получить полное представление о том, как работают сети, и получите знания, которые сможете использовать уже сегодня. Вы сможете не только усовершенствовать свои навыки, но так же будете в состоянии объяснить, почему сети работают так, а не иначе.

Издание будет полезно начинающим администраторам сетей.

УДК 004.71
ББК 32.972.5

ISBN 978-1-617-29363-4 (анг.)
ISBN 978-5-94074-519-6 (рус.)

Содержание

Предисловие

Для ИТ-специалистов одна из самых сложных концепций для полноценного понимания – это компьютерные сети. Сеть – это не отдельная вещь, как, например, программа, принтер или материнская плата. Это широкий, иногда неясный, набор оборудования, который работает как единое целое, передавая данные из одного места в другое. Уровень сложности сети может быть обескураживающе высок, именно поэтому некоторые ИТ-специалисты прилагают множество усилий, чтобы оставаться в стороне. Сети часто выглядят неприступно.

Если вы пытались читать другие книги, посвященные компьютерным сетям, то уже могли отметить, что все они чрезмерно академичны и переполнены теорией. Они запутывают вас непонятной терминологией, не связанной с практикой, и не дают практических навыков. Моя цель при написании этой книги состояла в том, чтобы сделать сети доступными для понимания ИТ-специалистам, которые любят технологии и которым нравится учиться, но находят сети слишком запутанными и отнимающими много времени. Вам не придется тратить несколько лет на изучение теоретических концепций, достаточно будет и месяца. В течение этого срока вы получите представление о том, как работают сети, и выполните практические задания, которые сможете использовать уже сегодня. Для вас станет очевиден смысл концептуальных понятий, потому что вы сможете связать их с повседневными задачами администрирования. Вы сможете не только усовершенствовать свои навыки, но также будете в состоянии объяснить, почему сети работают так, а не иначе.

Уделяйте обучению часть вашего обеденного перерыва каждый будний день в течение месяца, и к тому времени, когда вы закончите, у вас будет блестящий набор навыков работы с сетями, которые вы сможете продемонстрировать своему начальнику, друзьям или будущему работодателю.

Давайте приступим!

Благодарности

Спасибо Эрику за рецензию первых глав. Спасибо Билу, Брэду, Кевину, Майлсу и Миранде за всестороннюю поддержку в процессе написания этой книги.

Спасибо всем читателям, оставившим свои комментарии, которые помогли сделать книгу лучше. Бенуа Бенедетти, Чаду Маколи, Дэвиду Кернсу, Кенту Р. Спиллнеру, Луису Му, Марку Фурману, Микаэлю Доутри, Рою Легаарду-младшему, Сау Фэй Фонг и Шону Болану. Особую благодарность выражаю Джеймсу Беркенбиле, техническому корректору, который тщательно проверил рукопись.

И последнее, но не менее важное: благодарю людей из издательства Manning, особенно Марьяну Бас, Хелен Стерджиус, Грега Уилд и Дона Джонс. Всех тех, кто работал со мной в редакционной, производственной и рекламной сферах как напрямую, так и за кадром. Вы помогли претворить эту книгу в жизнь!

Об этой книге

Для администраторов серверов и сотрудников технической поддержки сеть уже давно является таинственным лабиринтом коробочек и проводов. Из этой книги вы почерпнете знания о существующих сетях, приобретете или усовершенствуете навыки работы в сети, выполните ряд практических заданий, которые почти сразу же сумеете применить на практике.

Большинство из того, что вам нужно для начала работы с этой книгой, описано в главе 1, но есть некоторые вещи, о которых я хочу упомянуть заранее.

Во-первых, чтобы выполнить полностью все практические упражнения, вам понадобится доступ к физической или виртуальной сети Cisco. О практических заданиях более подробно я расскажу в главе 1. Просто будьте готовы потратить некоторое время и силы на создание своей тестовой сети, если у вас ее еще нет.

Во-вторых, я составил эту книгу таким образом, чтобы вы начали с наиболее общих задач сетевого администратора. Это фундаментальные задачи и понятия, которые являются необходимым фундаментом для успешного усвоения материала, изложенного в последующих главах. Читайте книгу по порядку и не пропускайте ни одной главы.

В-третьих, существует ряд соглашений, принятых в этой книге, для облегчения восприятия. `Моноширинным шрифтом` выделены команды, которые вы вводите или которые должны увидеть. *Курсивный шрифт* указывает на важные термины и понятия, касающиеся сетей, которые вам нужно запомнить. Не удивляйтесь, когда вы увидите команд намного больше, чем терминов!

Такая пиктограмма обозначает совет.

Такая пиктограмма обозначает примечание.

О программном коде

Эта книга содержит множество примеров команд Cisco, размещенных рядом с обычным текстом. Они отформатированы моноширинным шрифтом, вот так, чтобы отделить их от обычного текста. Решения практических заданий и рекомендации по настройке виртуальной лаборатории доступны для загрузки по ссылке **Source Code** на веб-сайте книги по адресу www.manning.com/books/learn-cisco-network-administration-in-a-month-of-lunches.

Общение с автором

Приобретя данную книгу, вы получаете доступ к закрытому форуму, где вы можете оставить свои замечания, задать технические вопросы и получить

помощь от автора и других пользователей. Чтобы получить доступ к форуму и подписаться на сообщения в нем, перейдите на сайт www.manning.com/books/learn-cisco-network-administration-in-a-month-of-lunches и щелкните мышью по ссылке **Book Forum**. На этой странице может состояться полноценный диалог между отдельными читателями и между читателями и автором. Каких-либо юридических обязательств, присутствия на форуме со стороны автора нет, участие в форуме книги остается добровольным (и бесплатным). Онлайн-форум автора и архивы предыдущих обсуждений будут доступны на сайте до тех пор, пока книга будет издаваться.

Об авторе

Бен Пайпер (benpiper.com) – практикующий ИТ-консультант, имеющий многочисленные сертификаты Cisco, Citrix, и Microsoft, включая Cisco CCNA и CCNP. Он изучил свыше 17 курсов Pluralsight, охватывающих сети, сертификацию Cisco CCNP, Puppet и администрирование серверов Windows.

Глава 1

Прежде чем начать

Большинство корпоративных сетей построено с использованием оборудования Cisco, преимущественно коммутаторов и маршрутизаторов. Оборудование такого рода служит долго, и даже если наступает время его заменить, большинство предпочитает сохранять верность надежному и проверенному бренду. Таким образом, и большие корпорации, и малый бизнес широко применяют оборудование Cisco и заинтересованы в его надлежащей эксплуатации. Сети, однако, требуют постоянного обслуживания. Здесь не сработает принцип «настроил и забыл». Например, когда организация нанимает или увольняет персонал, пересаживает за другие столы, переводит в другой отдел, необходимо произвести соответствующие изменения в сети. Если увеличивается число сотрудников (и число компьютеров), требуется расширение сети и увеличение числа устройств Cisco. Эта книга научит вас конфигурировать коммутаторы и маршрутизаторы Cisco, чтобы вы сумели самостоятельно справиться с подобными сетевыми изменениями и расширениями.

1.1. Для вас ли эта книга?

Прежде чем мы начнем, давайте убедимся, что эта книга именно для вас. Если вы заинтересованы в получении сертификата администратора сетей Cisco (CCNA) или инженера по вводу в эксплуатацию (CCENT), то эта книга станет необходимым начальным условием в достижении цели. Прочтение только одной этой книги не сделает вас сертифицированным специалистом, зато *даст* твердый фундамент знаний, который сэкономит много времени и сил, когда придет время получать сертификат. В дополнение к четкому и ясному пониманию концепции и принципов работы коммутаторов и маршрутизаторов эта книга научит вас строить сети Cisco, расширять их, приспосабливая к нуждам растущей организации, и решать небольшие проблемы.

Большие корпорации часто позволяют себе роскошь нанять одного или нескольких сетевых администраторов. Эти ребята, имеющие преимущество в виде сертификата Cisco, могут ничего не делать, работая с сетями. Но, что неожиданно, даже крупные предприятия редко имеют хотя бы пару сетевых администраторов с полной занятостью. Малые и средние предприятия часто

не содержат в штате даже одного администратора, и тогда задача по управлению сетью ложится на человека, ответственного за эксплуатацию рабочих станций, серверов и программного обеспечения. На первый взгляд кажется, что само небо благословляет такой выбор. Человек, который знает входящие и исходящие серверы и программные приложения компании, – первый, кто должен знать и понимать, как настроить сетевое оборудование. Такие люди имеют целостное представление о сетевом ландшафте и наилучшим образом подходят для решения подобных задач.

Но все большее число организаций приходит к выводу, что подобные построения проблематичны. Что делать, если ответственный за «сетевые операции» в отпуске, а новому сотруднику в удаленном офисе требуется доступ к сети? Что, если он на больничном, а какого-либо пользователя переводят в другой отдел? Что, если бизнесу требуется расширение в новых условиях, но необходимо ждать, пока такой человек в одиночку выполнит все работы по расширению сети? Содержание администратора сети на полной ставке – откровенное излишество. Проблема не в том, что они ленивы или безответственны. Проблема не в том, что они не хотят никого в помощники (вероятно, хотят!). Проблема в том, что они единственные, кто *знает*, как администрировать сеть.

В отсутствие сетевого администратора у вас фактически есть выбор: подождать его возвращения или попытаться все сделать самому. *Эта книга для тех, кто не может ждать и берет бразды правления в свои руки.* Я покажу вам, как решать наиболее общие задачи по администрированию сети. Я покажу, как подключить нового пользователя, как произвести перемещения и изменения, как обезопасить сеть, используя списки доступа по протоколу IP, и даже как увеличить ее емкость, обеспечить рост, используя порты виртуальной сети (транки VLAN) и маршрутизацию по протоколу IP. Я поделюсь теорией сетей для «чайников», чтобы разобраться, почему сеть работает именно так, а не иначе, и дам практическую основу, необходимую, если вы решите в дальнейшем изучить сети более подробно.

Я встречал немало ИТ-специалистов, для которых сеть представлялась мистической паутиной кабелей и коробок, которая каким-то образом соединяла вместе компьютеры, серверы и приложения, все это было им знакомо. Но сеть для них сохраняла свою таинственность. Они *хотели* ее изучить, но не знали, с чего начать. У них были какие-то познания о сети, но они не знали, что с этим делать. Эта книга для ИТ-специалистов, для тех, кто хочет (или должен) стать профессионалом по работе с сетями Cisco за месяц.

1.2. Как пользоваться этой книгой

Постарайтесь сосредоточиться на одной главе в течение дня. Каждая глава предположительно потребует 30 минут на чтение и 30 минут на практику. Читайте книгу последовательно. Хотя впоследствии вы сможете использовать эту книгу как настольный справочник, важно, чтобы сначала вы уделили время на усвоение каждой главы.

1.2.1. Основные главы

Главы со 2 по 22 представляют основное содержание книги, которое вы можете освоить примерно за месяц. Не поддавайтесь соблазну перейти непосредственно к главе, которая освещает то, с чем вы недавно столкнулись или что вас особенно интересует. Я расположил главы с наиболее общими и основополагающими задачами настройки в самом начале, и с них и следует начинать, чтобы оставалось достаточно времени для их повторения и оттачивания профессионализма.

1.2.2. Практические занятия

Большинство глав содержит указания для практических занятий на закрепление материала. Каждое практическое задание состоит из нескольких задач и, иногда, из набора вопросов для самопроверки понимания изученного материала. Ответов на тесты в книге нет, но вы можете их найти, перейдя по ссылке: www.manning.com/books/learn-cisco-network-administration-in-a-month-of-lunches. Просто учитывайте, что обучение будет более эффективным, если вы будете самостоятельно обдумывать ответы на вопросы.

1.2.3. Углубленное изучение

Сети Cisco используют множество технологий, многие из которых довольно сложны и запутанны. Эта книга предлагает введение в большинство наиболее часто используемых технологий, вполне достаточное для профессионального конфигурирования реально работающих сетей. Если у вас возникнет потребность в бóльших объемах информации, чем вы получили, я укажу ссылки на дополнительные ресурсы, которые помогут вам расширить набор профессиональных навыков.

1.2.4. Дополнительно

Иногда для понимания некоторых специфичных аспектов сетей необходима развернутая информация. Врезки «Дополнительно» содержат дополнительную информацию, которая может оказаться полезной при изучении наиболее трудных аспектов. Их можно пропустить и вернуться к ним позже.

1.3. Практические соображения

Единственно возможный путь освоить администрирование сетей Cisco – это потренироваться в выполнении тех же задач, которые возникают при администрировании реальных сетей. Именно потому данная книга и предлагает описание практических заданий. Для их выполнения необходим соответствующий тестовый стенд. Начните со стенда с минимальными возможностями.

Для начала потребуется лишь компьютер с сетевой картой. Операционная система может быть Windows или macOS, главное, чтобы была возможность

административного или root-доступа. Можно, конечно, попрактиковаться и на существующей сети или собрать свою собственную. Следующие несколько разделов предлагают руководство по выбору тестового оборудования.

1.3.1. Выбор тестового оборудования

Лучший путь для изучения администрирования реальных сетей Cisco – попрактиковаться на одной из них. Идеальный вариант – это построить или позаимствовать сеть, использующую оригинальные коммутаторы и маршрутизаторы Cisco. Ваша организация, может быть, уже имеет нечто подобное, но мой личный опыт говорит, что в большинстве случаев отдельной учебно-тренировочной сети нет. Но тем не менее существуют какие-то запасные, неиспользуемые устройства. Приступая к построению сети, можно использовать как реальное оборудование Cisco, так и виртуальную лабораторию, имитирующую работу оборудования. Давайте рассмотрим преимущества и недостатки обоих вариантов.

Построение сети на реальном оборудовании Cisco даст вам более наглядное представление о том, как функционирует сеть. Когда вы видите Ethernet-кабель, идущий от одного коммутатора к другому, вы понимаете, где и как коммутаторы соединяются. Связь между различными устройствами прослеживается визуально и, как следствие, лучше запоминается. Может быть, даже щелчок штекера Ethernet-кабеля доставит вам удовольствие, когда вы будете подключать его к задней панели коммутатора, чтобы сбросить сетевой пароль. Такие, по-настоящему ценные навыки сетевого администрирования может предоставить только реальное, физически существующее оборудование.

Если у вас есть приятель или сотрудник, который может одолжить устройства Cisco, то это будет наименее затратный вариант. Если нет возможности попросить или арендовать эти устройства, их можно купить. Оборудование Cisco, бывшее в употреблении, продается совсем не дорого, хоть и не бесплатно.

В табл. 1.1 приведен список оборудования, которое я рекомендую для вашей тестовой сети, с указанием его ориентировочной стоимости. Вам потребуется два трехуровневых коммутатора типа Catalyst и один маршрутизатор. Для администрирования сети необходимо, чтобы на вашем компьютере был один свободный последовательный порт USB или RS-232. Также вам потребуется голубой витой кабель Cisco, иногда его называют консольным кабелем. Если на компьютере нет порта RS-232, потребуется адаптер RS-232/USB.

Таблица 1.1. Необходимое оборудование для тестовой сети

Устройство	Количество	Модель	Ориентировочная стоимость (б/у, в рублях)
Catalyst 3560 (трехуровневый коммутатор)	2	WS-C3560-24TS-S	6000 (каждый)
1841 Маршрутизатор с интегрированным сервисом	1	CISCO1841-SEC/K9	7500
Консольный/Витой кабель Cisco	1	72-3383-01	300

Когда вы будете приобретать коммутаторы или маршрутизатор, продавец, скорее всего, приложит витой кабель бесплатно. Ну или, как минимум, вам дадут на него скидку.

1.3.2. Рассмотрим виртуальную лабораторию

Преимущества и недостатки виртуальной лаборатории совершенно обратны тем же характеристикам реальной сети. Виртуальная лаборатория не требует приобретения или аренды физического оборудования. Но следует понимать, что виртуальная лаборатория не предоставит того же опыта администрирования, что и физическая сеть, не поможет развить навыки, необходимые при работе с устройствами Cisco, как при непосредственном, ручном доступе. Соединение компьютера с виртуальной сетью – это совсем не то же самое, что подключение компьютера к реальной, физической сети. Если вы решили использовать виртуальный маршрутизатор, скачайте пошаговое руководство по его настройке по адресу www.manning.com/books/learn-cisco-network-administration-in-a-month-of-lunches (ссылка **Source Code**).

Моя задача – научить вас не разворачивать с нуля виртуальную среду, а как построить реальную, полнофункциональную сеть Cisco, которую вы можете встретить в структуре той или иной организации. Существуют значительные различия между виртуальной и реальной сетями, и я не собираюсь акцентировать на них внимание, потому что вы никогда не столкнетесь с ними в реальной сети.

Одной из наиболее популярных виртуальных сред является GNS3. Это мощная среда виртуализации сетей, но ее мощь – результат некоторого компромисса. Для начинающих ее настройка – задача несколько более сложная, чем настойка физической сети. Также необходимо приобрести копию операционной системы Cisco IOS (Internetwork Operating System) – лицензионное программное обеспечение Cisco, которое доступно только тем пользователям и организациям, которые заключили с компанией договор об обеспечении технической поддержки. Это значит, что если вы решили использовать GNS3, вам необходимо отыскать и установить образ IOS.

Другая возможность состоит в использовании платформы виртуализации сетевых устройств Cisco (Cisco Virtual Internet Routing Lab, VIRL). Платформа VIRL не бесплатна, но и не ужасающе дорога. Персональная лицензия сроком на год обойдется примерно в 200 долларов. Преимущество VIRL – в том, что Cisco осуществляет техническую поддержку, производит регулярные обновления с исправлением недоработок и внедрением новых возможностей. Недостаток тот же, что у GNS3, – она слишком сложна в настройке.

1.3.3. Практика в условиях реальной сети

Если использование собственной физической или виртуальной сети связано с проблемами, в качестве альтернативы можно потренироваться на рабочей сети. Это даст вам даже больше, чем физическая тестовая сеть, но вы не сможете выполнить все практические задания. Кроме того, практика на реальной сети сохраняет определенные риски. Если вы все же решились попробовать

это, следует заручиться поддержкой сетевого администратора или команды, ответственной за эксплуатацию сети. Вам необходимо получить так называемый привилегированный аккаунт администратора для доступа к коммутатору или маршрутизатору, ну, и физический доступ к этим устройствам.

1.3.4. Мои рекомендации для вашей тестовой среды

Хотя некоторые из практических заданий вы сможете выполнить и в рабочей сети, предпочтительнее было бы, чтобы вы приложили максимум усилий для создания собственной тестовой сети. Если нет возможности получить доступ к физической тестовой сети, я бы рекомендовал создать виртуальную. Как VIRL, так и GNS3 предъявляют определенные требования к вашему компьютеру. Вам необходим компьютер с установленной операционной системой Windows версии 7 или выше, с оперативной памятью (RAM) не менее 8 Гб и 60 Гб свободного пространства на жестком диске. Для VIRL, помимо прочего, необходим процессор Intel с поддержкой аппаратной виртуализации VT-x и технологии страничных таблиц EPT. Подведем итоги.

- *Хорошо: привилегированный доступ к устройствам Cisco в рабочей сети* – вы не сможете выполнить все практические задания, и для их выполнения необходимо получить разрешение ответственного лица, убедив его в том, что вы не нарушите работу сети. Необходимо иметь компьютер с операционной системой Windows 7 или macOS, а также с последовательным портом USB и адаптером RS-232/USB. Нужен витой кабель Cisco.
- *Лучше: виртуальная лаборатория на GNS3, VIRL или любой другой на виртуальной платформе* – вам придется потратить немного больше времени и денег, но вы сможете выполнить бóльшую часть практических заданий этой книги. На странице www.manning.com/books/learn-cisco-network-administration-in-a-month-of-lunches вы найдете пошаговые инструкции по настройке виртуальной среды.
- *Лучшее: простенький тестовый стенд с двумя трехуровневыми коммутаторами и одним маршрутизатором* – это даст вам представление о том, как выглядит реальная сеть, и предоставит полную свободу в экспериментах без опасения ее нарушить. Нет необходимости выпрашивать разрешения у начальства, и если что-то пойдет не так, можно перезагрузиться и начать все заново. Вам понадобится компьютер с операционной системой Windows 7 или выше, с последовательным портом RS-232 или портом USB и адаптер RS-232/USB. Также необходим витой кабель Cisco.

1.3.5. Версии операционной системы Cisco IOS

Cisco Internetwork Operating System (IOS) – это программное обеспечение для управления маршрутизаторами и коммутаторами Cisco. Это то, что обеспечивает интерактивность при конфигурировании устройств Cisco. Во время написания этой книги я использовал систему IOS версии 15, и все написанное верно именно для этой версии. Если вы создаете свою сеть на оборудовании, бывшем

в употреблении, целесообразно использовать версию программного обеспечения постарше. Это не должно стать проблемой, так как вы будете выполнять фундаментальные задачи конфигурирования, которые не претерпевают изменений в течение многих лет. Оборудование Cisco служит долго, и многие сети работают на этом оборудовании долгие годы. Даже предпочтительнее начать с более или менее старой версии, так что не расстраивайтесь, если у вас не самая последняя и полная версия IOS.

Множество организаций работает с использованием как старого, так и нового оборудования Cisco. Новейшее оборудование требует версии IOS не ниже 15, а оборудование постарше может работать на версии 12 или даже более ранней. В основном различные версии IOS совместимы, так, например, коммутатор, работающий под управлением версии IOS 12.4, может без проблем взаимодействовать с коммутатором, управляемым версией IOS 15. Опять же, так как вы будете выполнять фундаментальные задачи конфигурирования, то можете обнаружить, что все, что работает для новейшего оборудования Cisco с IOS версии 15, также работает и на старом, потрепанном коммутаторе под управлением операционной системы IOS 12.4. Но я сконцентрируюсь на IOS 15. Если вы используете другую версию, просто учитывайте, что некоторые команды могут иметь незначительные отличия, но я не буду акцентировать на этом внимание.

1.4. Онлайн-ресурсы

Для получения полных инструкций по настройке вашей системы щелкните мышью по ссылке **Source Code** на сайте www.manning.com/books/learn-cisco-network-administration-in-a-month-of-lunches. Если у вас возникают вопросы или проблемы, посетите форум, посвященный этой книге, ссылка на который опубликована на том же сайте.

1.5. Замечание по моим рекомендациям

Некоторые компании бдительно следят за обновлениями Cisco и меняют оборудование для поддержки последних версий IOS. У других тот же самый маршрутизатор работает 10 лет, и когда он выходит из строя, сетевой администратор ищет в Интернете объявления о продаже подержанного оборудования в поисках идентичной модели. Вы никогда не знаете, с чем придется столкнуться, поэтому я и попытался свести на нет значимость версии программного обеспечения. Почти все, чему вы научитесь, применимо к *любому* маршрутизатору или коммутатору Cisco, вне зависимости от того, работают они в вашей тестовой среде или в структуре предприятия.

Если вы решите самостоятельно поискать оборудование и программное обеспечение, то попросту захлебнетесь в рыночном жаргоне и технических подробностях; настолько широк диапазон предложений Cisco. Компания Cisco выпускает устройства для обслуживания организаций всех типов и размеров, от небольших магазинчиков до интернациональных корпораций. Следует по-

нимать, что данная книга охватывает лишь малую часть всей экосистемы Cisco. Хотя эта книга и *сделает* вас эффективным администратором сетей Cisco, она не превратит вас в мастера по любому устройству или программе со словом *Cisco* в названии. В организационной структуре компании коммутаторы Cisco могут работать под управлением не только системы IOS, но и системы Nexus (Nexus Operating System, NX-OS). Хотя система NX-OS и имеет значительные отличия от архитектуры IOS, консольные конфигурации для задач, описанных в этой книге, в большинстве те же самые. Навыки, приобретенные по прочтении последующих глав, легко транслируются в NX-OS, так что не поддавайтесь на провокации тех, кто говорит, что вы упускаете нечто важное, придерживаясь IOS. Нет. Правда как раз в обратном. Вы получите фундаментальные знания и навыки, которые будете применять ежедневно, вне зависимости от того, на какой платформе и с какой программной версией придется работать.

1.6. Немедленно стать эффективным администратором сетей

Вероятно, уже сейчас вы готовы нырнуть непосредственно в главы, посвященные практическим вопросам. Но для начала обдумайте два вопроса, которые ставят в тупик большинство новичков:

- для чего на самом деле нужны маршрутизаторы и коммутаторы?
- почему устройства имеют и MAC-адрес, и IP-адрес?

В следующей главе я дам ответ на оба вопроса при описании того, как работает сеть Cisco. Если вы уже пытались понять концепцию сети и нашли, что это сложно и обескураживающе, следующая глава станет для вас приятным сюрпризом.

Я составил эту книгу таким образом, чтобы каждая глава учила вас чему-нибудь такому, что вы немедленно могли бы применить в реальной работающей сети. Это значит, что я часто пропускаю или лишь слегка касаюсь некоторых теоретических основ. Чтобы немедленно стать эффективным сетевым администратором, вы вовсе не должны обладать глубоким пониманием теоретической концепции сетей. Когда необходимо, я привожу теорию, после того как вы попрактикуетесь в достаточном объеме, чтобы вы видели, как соотносится теория со спецификой задач администрирования. Когда есть выбор между тем, чтобы показать что-нибудь или рассказать что-нибудь, я всегда сначала выбираю показать. Это вовсе не означает, что я не касаюсь теории совсем. Касаюсь, но лишь в том объеме, чтобы вы могли применить теорию непосредственно на практике. Учитывайте, эта книга – лишь стартовая точка, и вы можете потратить годы (что многие и делают), изучая в деталях, как и почему сети работают именно так. Но, прежде чем бежать, надо научиться ползать. Опять же, цель моей книги – в том, чтобы немедленно сделать вас эффективным сетевым администратором, а не совершенным специалистом в этой области. Итак, без дальнейших задержек давайте приступим к первому уроку.

Глава 2

Что такое сети Cisco?

Любая организация проводит основные объемы трафика через устройства двух типов: коммутаторы и маршрутизаторы. Cisco – наиболее популярный бренд, производящий надежные коммутаторы и маршрутизаторы, поэтому многие компании приняли его как стандарт для подобного рода устройств. Для прочего сетевого оборудования, например брандмауэра или точки беспроводного доступа, кто-то предпочитает Cisco, кто-то выбирает что-нибудь другое или использует бренды совместно. Но если сеть построена с использованием маршрутизаторов и коммутаторов Cisco, то это сеть Cisco.

Нет никаких обязательных требований к тому, чтобы использовать исключительно этот бренд. Вы можете использовать коммутаторы Cisco с маршрутизаторами Juniper, и они будут прекрасно работать вместе. Можно использовать маршрутизатор Cisco с коммутатором Juniper, и они тоже прекрасно уживутся. Но есть парочка возражений против подобных тандемов.

Во-первых, последовательность конфигурирования устройств Cisco в корне отлична от настройки оборудования Juniper. Синтаксис команд и терминология совершенно различны. Администрирование смешанных сетей требует знаний обеих платформ и принципов их взаимодействия, а эта книга посвящена только оборудованию компании Cisco.

Во-вторых, если у вас возникают проблемы и вы не уверены, связаны они с маршрутизатором или коммутатором, вам придется обращаться за техподдержкой сразу к обеим компаниям. В худшем случае каждая компания начнет тыкать пальцем в конкурента. В лучшем случае это чревато задержками, пока они придут к соглашению.

Использование в одной сети коммутаторов и маршрутизаторов разных брендов – это плохая идея. Вот почему большинство компаний использует и маршрутизаторы, и коммутаторы только компании Cisco. Так проще. И даже если у вас смешанная сетевая среда, эта книга все равно будет вам полезна, чтобы научиться администрировать коммутаторы и маршрутизаторы Cisco. Просто напомню, что в этой книге описывается сеть Cisco, и это *всегда* маршрутизаторы и коммутаторы компании Cisco.

На рис. 2.1 показано, как мой компьютер пересылает «конверт», содержащий некоторые данные, на сервер базы данных. В этой главе вы узнаете, как коммутаторы и маршрутизаторы определяют наилучший путь для передачи данных.

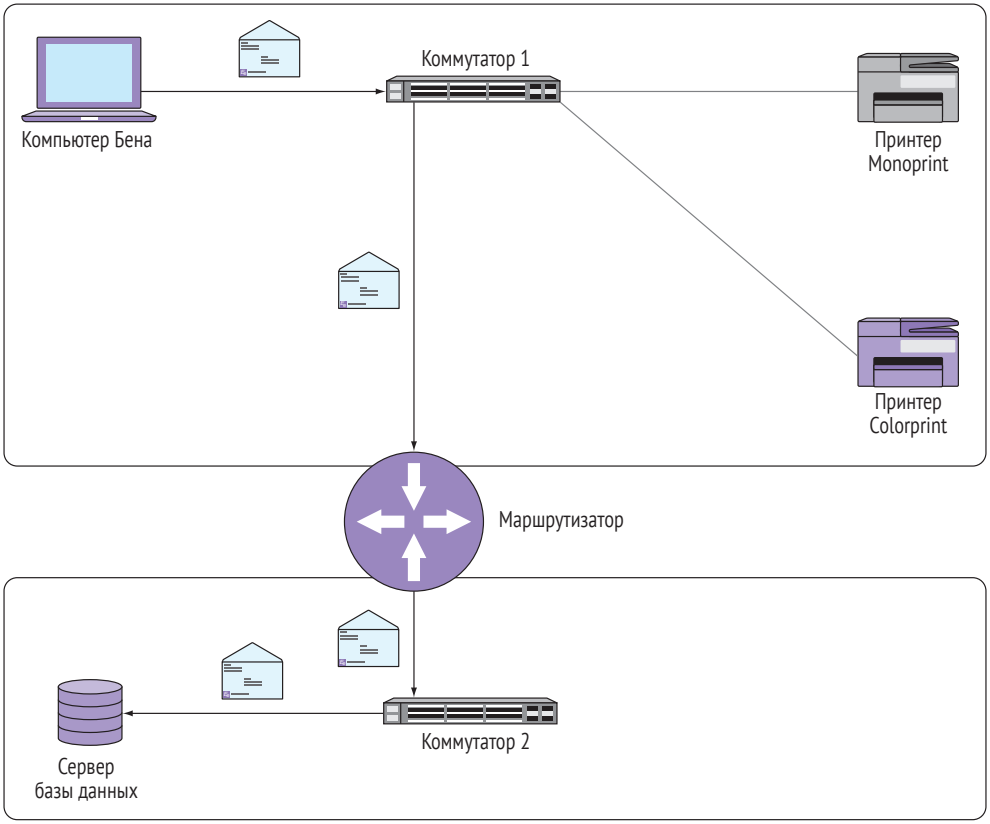

Рис. 2.1 ❖ Коммутаторы и маршрутизатор в сети

2.1. Правда о коммутаторах и маршрутизаторах

Новички часто задают два вопроса:

- что на самом деле делают коммутаторы и маршрутизаторы?
- почему эти устройства имеют и MAC- и IP-адреса?

Эти, казалось бы, простые вопросы не имеют простых ответов. Я неоднократно наблюдал попытки дать ответ на эти вопросы в нескольких предложениях, но все эти попытки вносили лишь больше сумятицы и еще больше запутывали.

Истина в том, что и коммутаторы, и маршрутизаторы – это порождение конкретной технологической необходимости, а не каких-либо общих практических надобностей. В принципе, ни один из этих приборов не наделен какой-то особенной интеллектуальностью, хотя Cisco и снабжает их некоторым количеством «мозгов», чтобы улучшить их функциональность. Как и большинство

технологий, коммутаторы и маршрутизаторы появились как результат сомнительных решений, принятых десятилетия назад.

Новые технологии обычно строятся на более ранних. Например, электронные книги позаимствовали концепции *страниц* и *закладок* у традиционных печатных книг. Попробуйте объяснить, что такое страница, кому-нибудь, кто знаком с прокруткой, но никогда не видел традиционных печатных книг. Как вы это сделаете? Прежде чем объяснять, что такое страница, надо объяснить, зачем они существуют.

Поэтому, прежде чем объяснять, что такое маршрутизатор или коммутатор, я должен коротко пояснить, для решения каких проблем они служат. После того как вы это поймете, все встанет на свои места, и вы сразу же сможете администрировать собственную сеть Cisco.

2.2. MAC-адрес

Много лет назад кто-то решил, что все сетевые приборы должны иметь определенный идентификатор, чтобы идентифицировать друг друга в сетевом пространстве, и назвал этот идентификатор MAC-адресом (от англ. Media Access Control – управление доступом к среде). MAC-адрес – это строка длиной 48 бит, содержащая шестнадцатеричное число, примерно вот так: `0800.2700.EC26`. Вероятно, вы уже встречались с чем-то подобным.

Что интересно: производители сетевых устройств присваивают им MAC-адреса еще на стадии изготовления. Целесообразность этого состоит в том, что можно просто включить устройства в сеть и коммутировать их между собой, не имея никакого руководства по конфигурации. Звучит достойно, но есть одна проблема: производитель присваивает MAC-адрес в отсутствие связи с тем, куда именно будет помещено устройство в конечном итоге. То есть это не совсем адрес, поскольку он совершенно не помогает в определении месторасположения устройства.

Практикум

Запустите оболочку командной строки Windows и введите команду `ipconfig /all`. В появившемся списке MAC-адрес сетевой карты вашего компьютера будет указан в строке **Физический адрес** (Physical Address). Если установлено несколько сетевых карт, вы увидите несколько MAC-адресов.

MAC-адрес сродни полному имени человека. Его присваивают при рождении для простой идентификации, чтобы выделить человека из толпы или послать сообщение на его имя. Если мы с вами находимся в толпе людей и вы хотите послать мне сообщение, но понятия не имеете, где я, вы можете, набрав побольше воздуха, крикнуть: «Бен Пайпер, где ты?» И если я в той толпе, то получу ваше сообщение.

Сетевые устройства общаются друг с другом таким же образом, но вместо полного имени используют MAC-адреса. Предположим, мой компьютер имеет MAC-адрес `0800.2700.EC26`, и его надо напечатать на сетевом принтере с именем Monoprint и MAC-адресом `0020.3500.CE26`. Мой компьютер физически соединен с принтером через устройство, называемое коммутатором, как показано на рис. 2.2. Точнее, мой компьютер и принтер *физически* присоединены к отдельным Ethernet-портам коммутатора. Отметим, что, в отличие от беспроводной точки доступа, подключение к коммутатору *всегда* производится с помощью кабеля. Таким образом, коммутатор – это место сбора всех сетевых устройств. Подобно тому, как я с вами и с другими могу собраться на переполненном рынке, сетевые устройства собираются вместе в коммутаторе. Такой набор соединенных между собой устройств называется локальной вычислительной сетью (ЛВС, от англ. Local area network, LAN).

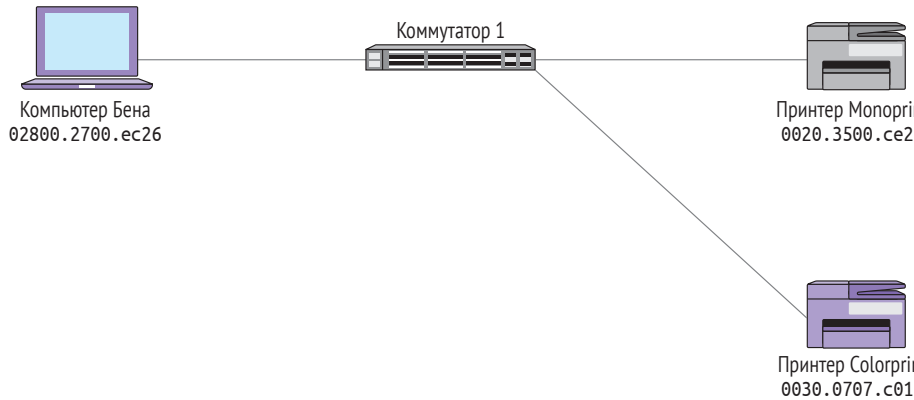

Рис. 2.2 ❖ Два принтера соединены с компьютером через коммутатор

Но здесь возникает проблема: мой компьютер не знает, где расположен принтер Monoprint, не знает даже, является ли он частью локальной сети – частью «толпы», подключенной к коммутатору. *MAC-адрес, подобно полному имени, может служить хорошим идентификатором, но он не может указать точного месторасположения устройства.* Именно поэтому мой компьютер вынужден просто «кричать в рупор», вызывая Monoprint по его MAC-адресу.

Дополнительно

Каждое устройство в процессе изготовления получает заводской уникальный идентификатор (organizationally unique identifier, OUI) в виде строки, содержащей шестнадцатеричное число. Идентификатор OUI образует левую часть MAC-адреса,

присваиваемого при изготовлении. Его можно рассматривать как «фамилию» прибора. Хотя они и присваиваются при «рождении», устройства одной серии имеют идентичный номер QUI. Остальная часть MAC-адреса – это просто следующий член возрастающей последовательности. Таким образом, производитель достигает уникальности MAC-адреса каждого устройства.

2.3. Ethernet-кадр: большой конверт

Мой компьютер создает *Ethernet-кадр*, содержащий указания на источник – его собственный MAC-адрес – и конечный адресат – MAC-адрес принтера. Рисунок 2.3 демонстрирует Ethernet-кадр в виде большого конверта с адресами отправителя и получателя.

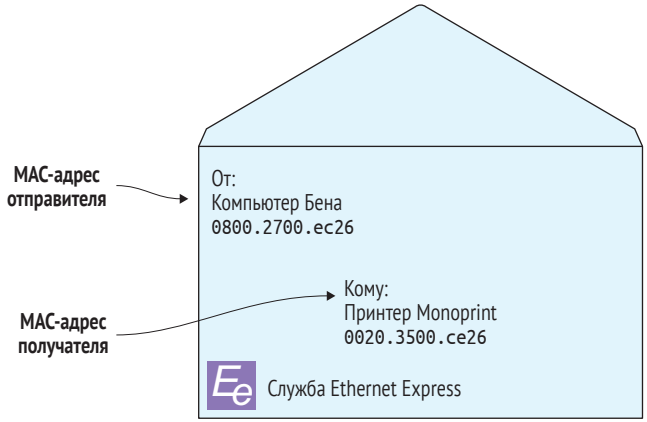

Рис. 2.3 ❖ Ethernet-кадр содержит MAC-адреса отправителя и получателя

Мой компьютер собирает данные, которые хочет обработать на принтере, помещает их в «большой конверт» и отправляет на коммутатор. Коммутатор получает кадр и обращается к MAC-адресу удаленного принтера. Изначально коммутатор не знает, подключен к нему принтер или нет, поэтому он рассылает кадр всем остальным подключенным сетевым устройствам для определения, есть ли среди них принтер. Это называется *лавинной передачей*.

На шаге 1, на рис. 2.4, мой компьютер отправляет Ethernet-кадр, адресованный принтеру Monoprint, со своим MAC-адресом (`0020.3500.ce26`). На шаге 2 коммутатор рассылает этот кадр всем подключенным устройствам.

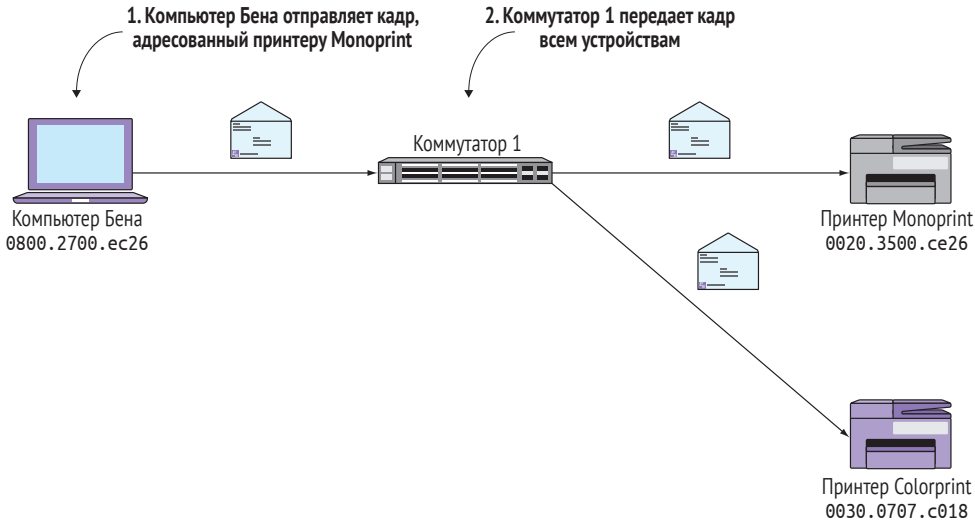

Рис. 2.4 ❖ Лавинная передача Ethernet-кадра

2.3.1. Когда все говорят, никто не слушает

Лавинная передача имеет тот же эффект, что и крик в рупор в большой толпе. Все слышат вас, но в то же время в толпе люди не могут расслышать друг друга. Для увеличения эффективности вы временно прекращаете их общение. Но даже после того, как вы прокричали в рупор, пройдет какое-то время, после того как люди получат ваше сообщение и поймут, что оно адресовано не им. То же самое происходит, когда коммутатор рассылает сообщение всем устройствам. Все они не в состоянии слышать друг друга, пока идет лавинная передача. А затем они должны обработать сообщение, чтобы понять – должны ли они что-то сделать в соответствии с ним. Это явление называется *прерыванием*.

Хотя несколько рассылок кадров и прерываний и не представляется чем-то значительным, представьте, что произойдет в толпе, скажем человек на 1000, в которой у каждого есть рупор. Как раз в тот момент, как вы собрались отправить мне сообщение через свой рупор, кто-то прямо рядом с вами кричит что-нибудь еще через свой. После того как у вас утихнет звон в ушах, вы поднимаете свой рупор только для того, чтобы опять быть прерванным кем-нибудь еще. Пока, наконец, не произойдет пауза, достаточная для пересылки сообщения. Да, это проблема. Вы действуете со всеми остальными в одной среде – в воздухе. При таком методе коммуникации «один – многим» трудно ожидать, что конкретная персона получит сообщение вовремя. И чем больше толпа, тем больше проблем.

В сети с несколькими устройствами лавинная передача не представляет проблем. А если в локальной сети сотни или тысячи устройств, то это проблематично. И это порождает другую проблему. Сеть, которая не может связать тысячи устройств, практически бесполезна.

2.4. Широковещательные домены

Предположим, что вы добавили в топологию сети еще один коммутатор, назвали его Коммутатор 2 и присоединили к нему сервер базы данных, как показано на рис. 2.5. Когда мой компьютер отправляет кадр на MAC-адрес сервера, Коммутатор 1 начинает лавинную передачу (и прерывание) на все устройства, присоединенные к его портам, включая и Коммутатор 2! Коммутатор 2, в свою очередь, тоже передает кадр всем устройствам. В этом случае сервер базы данных – всего лишь рядовое устройство, присоединенное к Коммутатору 2.

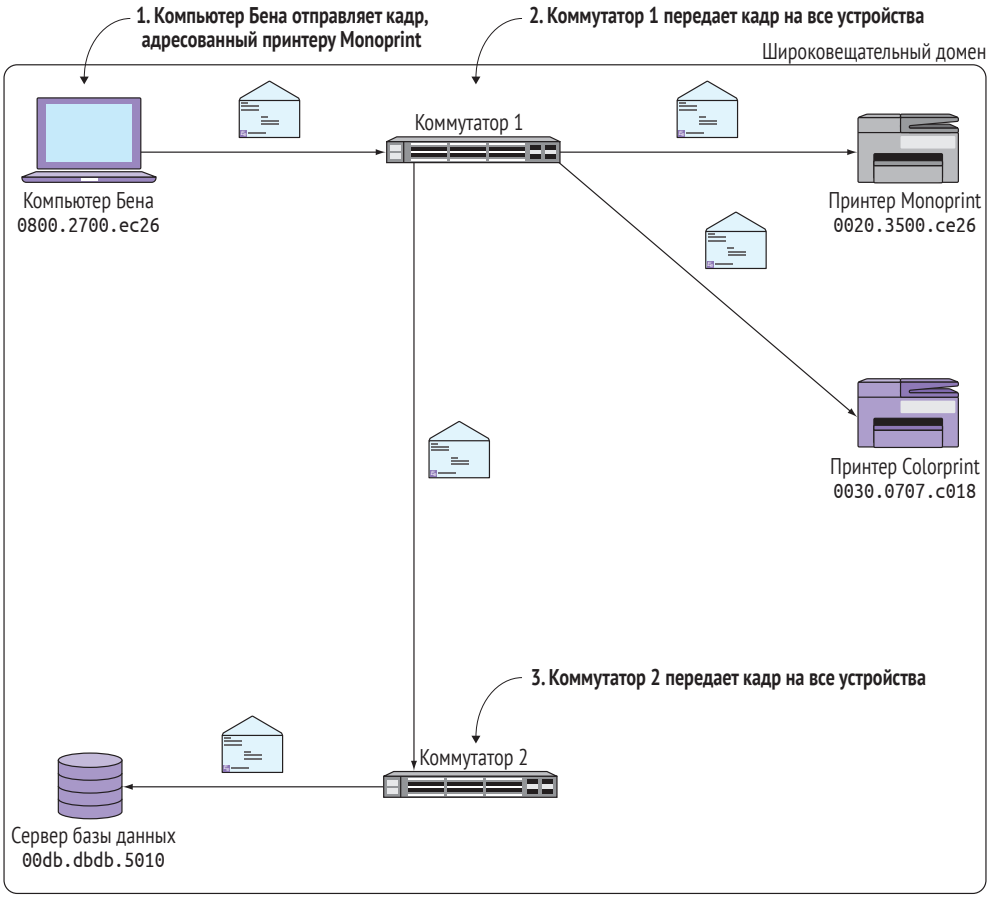

Рис. 2.5 ❖ Коммутатор 2 расширяет широковещательный домен

В шаге 1 мой компьютер пересылает кадр на MAC-адрес сервера базы данных (`00db.dbdb.5010`). В шаге 2 Коммутатор 1 рассылает кадр всем устройствам. И наконец, в шаге 3 Коммутатор 2 передает кадр на сервер базы данных.

Все эти устройства, которые получили кадр, – члены одного *широковещательного домена*. Широковещательный домен – это не устройство и даже не настраиваемый параметр, а скорее неотъемлемый атрибут сети. Для лучшего понимания представлю следующую аналогию.

Когда вы стоите один в центре улицы, вы – не толпа. Но если несколько человек собирается вокруг вас, вы становитесь частью толпы. И вы становитесь частью еще большей толпы, когда вокруг вас собирается больше людей. Вы не меняетесь, но меняется ваше виртуальное свойство – часть толпы, – в зависимости от того, сколько людей собралось вокруг вас. Точно так же и устройство становится частью широковещательного домена тех устройств, которые получили кадр при лавинной передаче.

2.4.1. Избавление от лавинной передачи: таблица MAC-адресов

Лавинная передача – неизбежная операция при использовании MAC-адресов. К счастью, коммутаторы используют ловкий трюк, чтобы уменьшить необходимость лавинной передачи. Каждый раз, когда коммутатор получает кадр, он изучает MAC-адрес источника и порт, к которому присоединен источник кадра. Эта информация используется для построения *таблицы MAC-адресов*.

Дополнительно

В документации Cisco таблица MAC-адресов иногда называется ассоциативной памятью (content addressable memory, CAM), но это одно и то же.

Когда Коммутатор 1 получает кадр от моего компьютера, он записывает его MAC-адрес `0800.2700.ac26`, а также порт, к которому компьютер подключен, – FastEthernet0/1. Эта информация добавляется в таблицу MAC-адресов, как показано в табл. 2.1.

Таблица 2.1. Таблица MAC-адресов Коммутатора 1

Устройство	MAC-адрес	Порт коммутатора
Компьютер Бена	`0800.2700.ec26`	FastEthernet0/1

Теперь предположим, сервер базы данных отправляет кадр с MAC-адресом моего компьютера. Кадр попадает на Коммутатор 2, который отправляет его прямиком на Коммутатор 1. Но вместо слепого забрасывания кадром всех устройств Коммутатор 1 проверяет таблицу MAC-адресов.

Он видит, что MAC-адрес `0800.2700.ec26` соответствует устройству, подключенному к порту FastEthernet0/1, и отправляет кадр *только* на этот порт, как показано на рис. 2.6. Это работает по принципу старого телефонного коммутатора, откуда и происходит термин *коммутатор*.

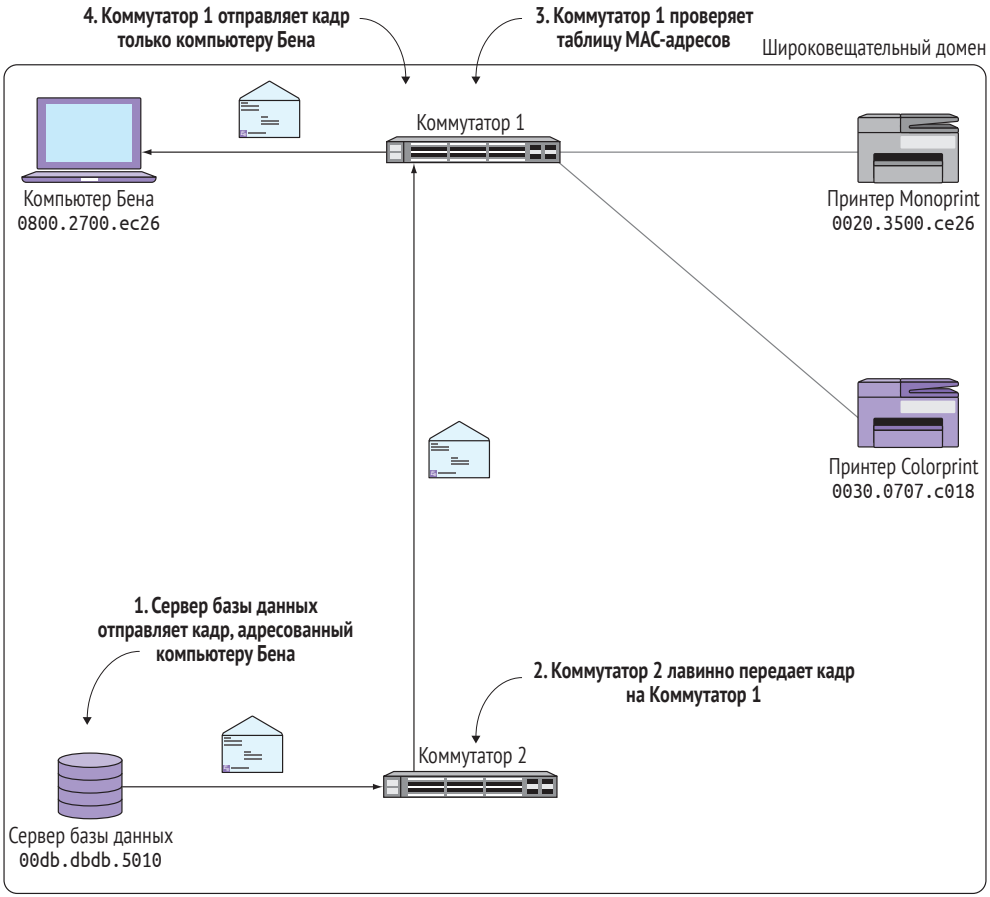

Рис. 2.6 ❖ Как таблица MAC-адресов позволяет избавиться от лавинной передачи

На шаге 1 сервер базы данных отправляет кадр на MAC-адрес моего компьютера (`0800.2700.ec26`). На шаге 2 Коммутатор 2 (лавинно) отправляет кадр на Коммутатор 1. На шаге 3 Коммутатор 1 сверяется с таблицей MAC-адресов и находит порт запрашиваемого адреса. На шаге 4 Коммутатор 1 отправляет кадр только на порт моего компьютера, а не лавинно передает кадр на все остальные устройства.

2.4.2. Разделение широковещательного домена

С ростом размера широковещательного домена коммуникации становятся все более затруднительными. И как следствие, широковещательный домен, состоящий из сотен устройств, начинает работать неудовлетворительно. Но современной компании требуется сеть, соединяющая тысячи устройств. И просто наличия связи недостаточно. Сеть должна быть быстрой и надежной.

Решение заключается в ограничении размера широковещательного домена. Это значит, что его нужно разбить на части таким образом, чтобы отдельные части имели связь друг с другом.

Возвращаясь к нашему примеру, мы видим, что простейший путь разбить широковещательный домен – это отключить Ethernet-кабель, соединяющий Коммутаторы 1 и 2, как показано на рис. 2.7. Отмечу, что коммутаторы не соединяются каким-либо иным способом. Это простая часть. А теперь сложная: мой компьютер и сервер базы данных размещены на разных широковещательных доменах. Не существует путей для их связи друг с другом. Что вы натворили? Вы не можете просто заново соединить коммутаторы, потому что воссоздадите то, что было, – единый широковещательный домен.

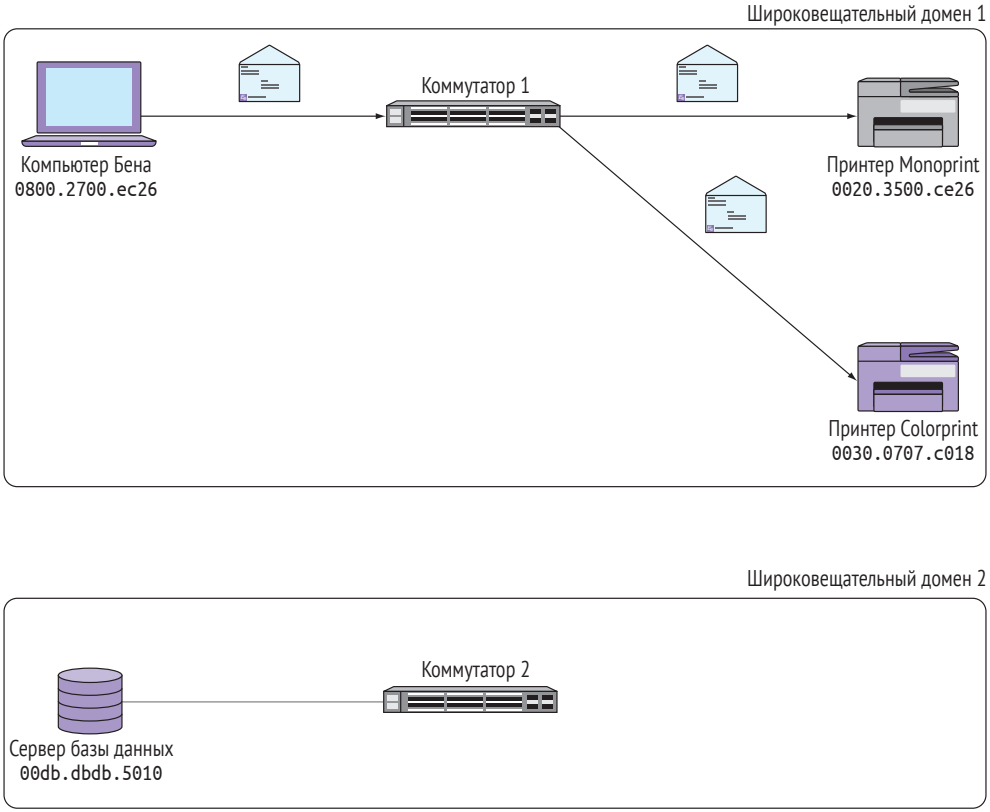

Рис. 2.7 ❖ Два широковещательных домена

2.4.3. Соединение широковещательных доменов

Для соединения двух широковещательных доменов без повторения этой ужасной проблемы лавинной передачи необходимо сделать две вещи.

Во-первых, так как два широковещательных домена не имеют связи, вам нужно специальное устройство, физически соединяющее их, но таким образом, чтобы рассылка кадров не выходила за границы широковещательного домена. Так как кадр содержит MAC-адреса и отправителя, и адресата, это устройство будет эффективно скрывать MAC-адреса одного широковещательного домена от другого.

Во-вторых, так как MAC-адреса одного широковещательного домена скрыты от другого, вам нужна другая схема адресации устройств для обращения к оборудованию в разделенных доменах. Новая адресная схема, в отличие от MAC-адресов, должна не только идентифицировать прибор, но и предоставлять какие-то указания на то, в каком домене прибор размещен. Давайте начнем с последнего.

2.4.4. Адресация устройств из разных широковещательных доменов

Схема адресации должна удовлетворять следующим требованиям:

- во-первых, адрес должен быть уникальным для всех широковещательных доменов. Два устройства из одного домена не могут иметь одинаковый адрес;
- во-вторых, адрес должен сообщать, какому домену он принадлежит. *Адрес должен быть не только уникальным идентификатором прибора, но также и сообщать другим устройствам, к какому домену он принадлежит.* Все это для того, чтобы избежать этих ужасных проблем лавинной передачи;
- в-третьих, адреса не могут присваиваться «при рождении», подобно MAC-адресу. Они должны конфигурироваться вами как сетевым администратором.

К счастью, вам нет необходимости ломать над этим голову. Такая адресная схема существует, и вы уже пользовались ею.

2.5. Адреса протокола Интернета

Вы уже знаете, как выглядят IP-адреса. Один из самых распространенных IP-адресов – `192.168.1.1`. Это последовательность четырех восьмеричных чисел (*октетов*), разделенных точкой, каждое число может располагаться в диапазоне от `0` до `255`.

Вы, вероятно, видели адреса типа `192.168.`*`x`*`.`*`x`*, всплывающие в различных местах. Это связано с тем, что адреса `192.168.`*`x`*`.`*`x`* зарезервированы для использования в частных сетях, используемых у вас дома или на работе. Они глобально не уникальны, так как не доступны в общем пространстве Интернета. Но вы можете их использовать для адресации устройств в своей собственной внутренней сети.

В отличие от MAC-адресов, вы можете присваивать IP-адрес любому устройству, какому захотите. Вы можете создать собственную схему адресации, основанную на месторасположении прибора, а не просто на том, что они есть. Давайте рассмотрим пример.

2.5.1. Где ты?

Устройства, присоединенные к Коммутатору 1, образуют домен 1, а устройства, присоединенные к Коммутатору 2, входят в состав домена 2. Вы можете присваивать адреса `192.168.1.`*x* устройствам в домене 1, а адреса `192.168.2.`*x* – членам домена 2. Даже не глядя на рис. 2.8, просто зная IP-адреса, можно со всей очевидностью определить, какому домену принадлежит устройство.

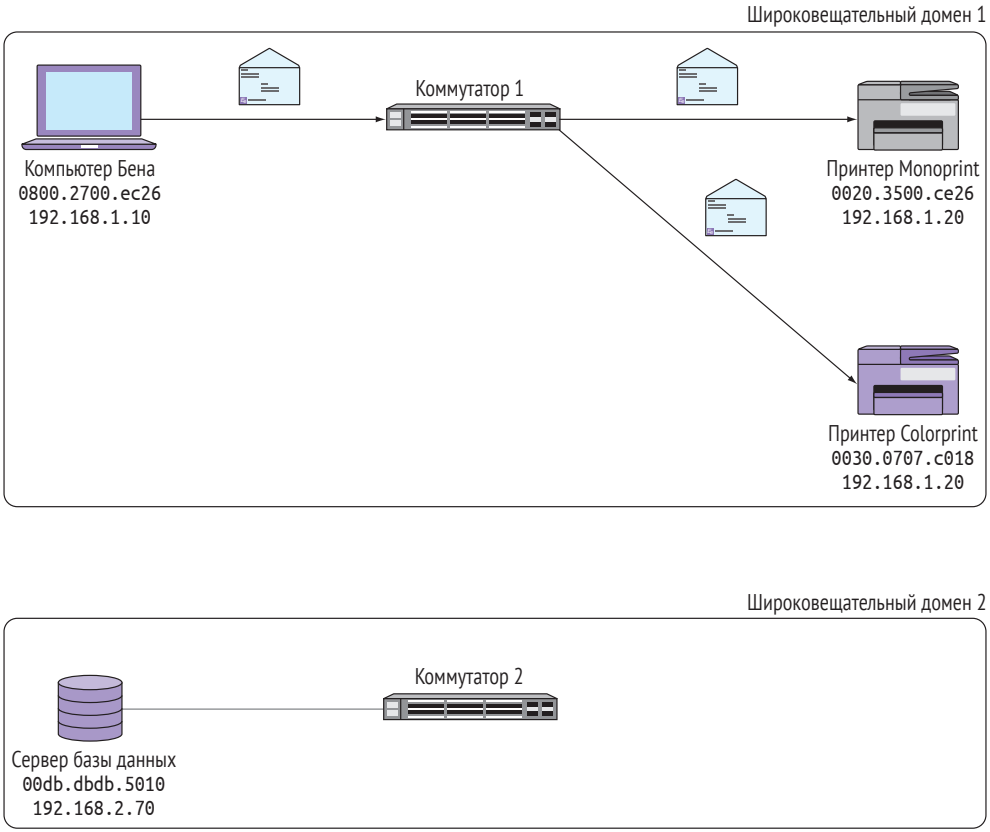

Рис. 2.8 ❖ Каждое устройство имеет IP-адрес, который соответствует его домену

Дополнительно

Обратите внимание: если вы хотите добавить третий широковещательный домен, то можете назначить адреса `192.168.3.`*x* устройствам в этом домене. Удобство использования IP-адресов заключается в том, что нет никакого практического ограничения на количество отдельных широковещательных доменов, которыми вы можете управлять.

Но у нас все еще нет связи между широковещательными доменами, устройства могут связываться между собой только внутри домена. Но возникает вопрос: теперь каждый прибор имеет два адреса, MAC- и IP-, какой из них использовать для коммуникаций *внутри* домена?

2.5.2. Дилемма: IP- или MAC-адрес

«Почему мы просто не можем использовать IP-адреса вместо MAC-адресов?» – распространенный вопрос среди ИТ-специалистов, пытающихся изучать сети. Это хороший вопрос.

Кроме всего прочего, MAC-адреса не очень удобны. Они тяжелы для запоминания, бессмысленны, их трудно (или невозможно) изменить. IP-адрес, напротив, легко запоминается, легко изменяется и может содержать множество полезной информации относительно месторасположения и функциональности. Победитель очевиден.

Итак, почему мы не можем просто использовать IP-адреса и забыть все вместе взятые MAC-адреса? Ответ прост, но немного тревожен.

Сетевые устройства внутри широковещательного домена все еще должны взаимодействовать с помощью MAC-адресов. Это требование стандарта Ethernet, которое существует уже десятилетия. Присвоение IP-адресов этого не меняет. Разумеется, кто-то может создать новый стандарт, который сделает MAC-адреса совершенно не нужными, но это потребует замены *всех* устройств в вашей сети.

Короче говоря, MAC-адреса по-прежнему используются. Это плохая новость. А хорошая новость состоит в том, что вам не нужно о них беспокоиться, ну или как минимум не очень часто.

2.5.3. ARP: протокол определения адреса

Напомню, что совместное использование MAC- и IP-адресов неэффективно и расточительно. Вот почему почти все приложения используют IP-адреса и совершенно игнорируют MAC-адреса. *Протокол определения адреса* (*Address Resolution Protocol*, *ARP*) делает это возможным.

Протокол ARP предоставляет возможность простого замещения MAC-адресов IP-адресами. Преимущество протокола ARP – в том, что он допускает использование дружественных IP-адресов, совершенно не обращая внимания на MAC-адреса. Все сетевые устройства, произведенные с середины 1980-х годов, используют протокол ARP по умолчанию, поэтому нет нужды его настраивать.

Предположим, что мой компьютер отсылает на принтер некоторое задание на печать. Оба устройства в одном домене, следовательно, продолжают взаимодействовать, используя MAC-адреса. Но вы как сетевой администратор можете даже и не вспоминать о них. И мой компьютер обращается к принтеру Monoprint по IP-адресу: `192.168.1.20`.

Рисунок 2.9 иллюстрирует работу ARP. Мой компьютер посылает *ARP-запрос*, который преобразуется в MAC-адрес принтера Monoprint. Этот запрос

говорит: «Это 192.168.1.10, и мой MAC-адрес 0800.2700.EC26. У кого 192.168.1.20?» Мой компьютер помещает такой ARP-запрос в Ethernet-кадр и отправляет его по специальному *широковещательному MAC-адресу*, FFFF.FFFF.FFFF, как показано на рис. 2.9.

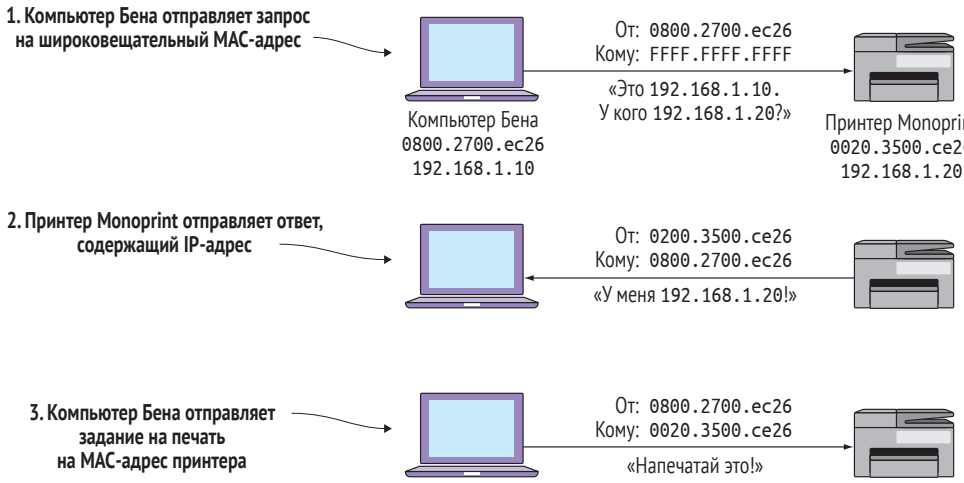

Рис. 2.9 ❖ Запросы и ответы в протоколе ARP

Напомню, что все сетевые устройства *должны* использовать MAC-адреса для коммуникации. Чтобы ARP-запрос моего компьютера получили все устройства в сети, он должен послать его по *некоторому* MAC-адресу. Он не может послать его на пустой адрес. Поэтому он посылает ARP-запрос на широковещательный MAC-адрес. Каждое устройство слышит широковещательный адрес в дополнение к своему собственному MAC-адресу. Это гарантирует, что каждый прибор в сети обратит внимание на любой ARP-запрос.

На шаге 1 мой компьютер отсылает ARP-запрос на широковещательный MAC-адрес (FFFF.FFFF.FFFF). На шаге 2 Monoprint возвращает ARP, заменяя содержащийся в нем IP-адрес на 192.168.1.20. Наконец, на шаге 3 мой компьютер отсылает задание на печать на MAC-адрес принтера Monoprint.

Коммутатор рассылает этот кадр по всем портам, включая и порт, к которому подключен принтер Monoprint. Принтер Monoprint получает кадр, рассматривает его и видит ARP-запрос. Принтер Monoprint видит вопрос: «Кто 192.168.1.20?» – и думает: «О, это мой IP-адрес!» Затем принтер Monoprint отсылает ARP-ответ на мой компьютер: «Это 192.168.20. Мой MAC-адрес – 0020.3500.CE26». Бинго. Теперь мой компьютер знает MAC-адрес и может использовать его для коммуникации.

Протокол ARP – это секретный «соус», который спасает от необходимости думать о MAC-адресах слишком часто. А ваша работа сводится к использованию дружественных, осмысленных IP-адресов большую часть времени.

2.6. Связь широковещательных доменов с помощью маршрутизатора

Теперь, когда вы можете использовать IP-адреса, пора изучить, как устройства могут использовать их для коммуникации между широковещательными доменами.

В данный момент у вас есть два отдельных, не связанных между собой домена. Чтобы их связать, не образуя единый широковещательный домен, вам необходимо специальное устройство, которое называется *маршрутизатором*. Маршрутизатор физически связывает домены таким образом, что кадры не могут покинуть их границ. Так как кадр содержит MAC-адреса и отправителя, и адресата, маршрутизатор эффективно скрывает MAC-адреса одного широковещательного домена от другого.

На рис. 2.10 маршрутизатор физически подключен к обоим доменам. Он имеет как минимум два порта или интерфейса, по одному на каждый связываемый домен. Каждый сетевой интерфейс маршрутизатора имеет уникальный MAC-адрес. Просто запомните, что каждый интерфейс маршрутизатора имеет уникальный MAC-адрес для совместимости с Ethernet-стандартами всех остальных устройств в сети. Подобно тому, как электронная книга продолжает использовать «страницы», маршрутизатор использует MAC-адреса для совместимости. Маршрутизатор имеет не только два MAC-адреса, но и два IP-адреса. Интерфейс маршрутизатора, подключенный к Коммутатору 1, имеет IP-адрес `192.168.1.254`. Интерфейс маршрутизатора, подключенный к Коммутатору 2, имеет IP-адрес `192.168.2.254`. Это уникальные IP-адреса, и третий октет в них указывает на домен.

2.6.1. Где ты? И где я?

Мой компьютер имеет IP-адрес `192.168.1.10` и принадлежит домену 1. IP-адрес сервера базы данных `192.168.2.70`, и размещен он в домене 2. Значение этих адресов очевидно каждому. Устройство с адресом `192.168.1.x` принадлежит домену 1, а адрес `192.168.2.x` принадлежит домену 2.

Но мой компьютер этого не знает. Компьютер – прежде всего это тупая машина, которая делает лишь то, что ему сказано. Поэтому компьютеру надо каким-то образом объяснить, где какой домен. Когда он это поймет, он сможет разобраться, принадлежит ли какой-либо прибор его собственному домену или какому-то другому.

2.6.2. Определение подсети

На самом деле широковещательным доменам номеров не присваивают, так как это не какая-то реальная, физически существующая вещь. Но ассоциация набора IP-адресов с абстрактным понятием широковещательного домена упрощает понимание принципов его работы. Набор адресов, привязанных к определенному домену, называется *подсетью*.

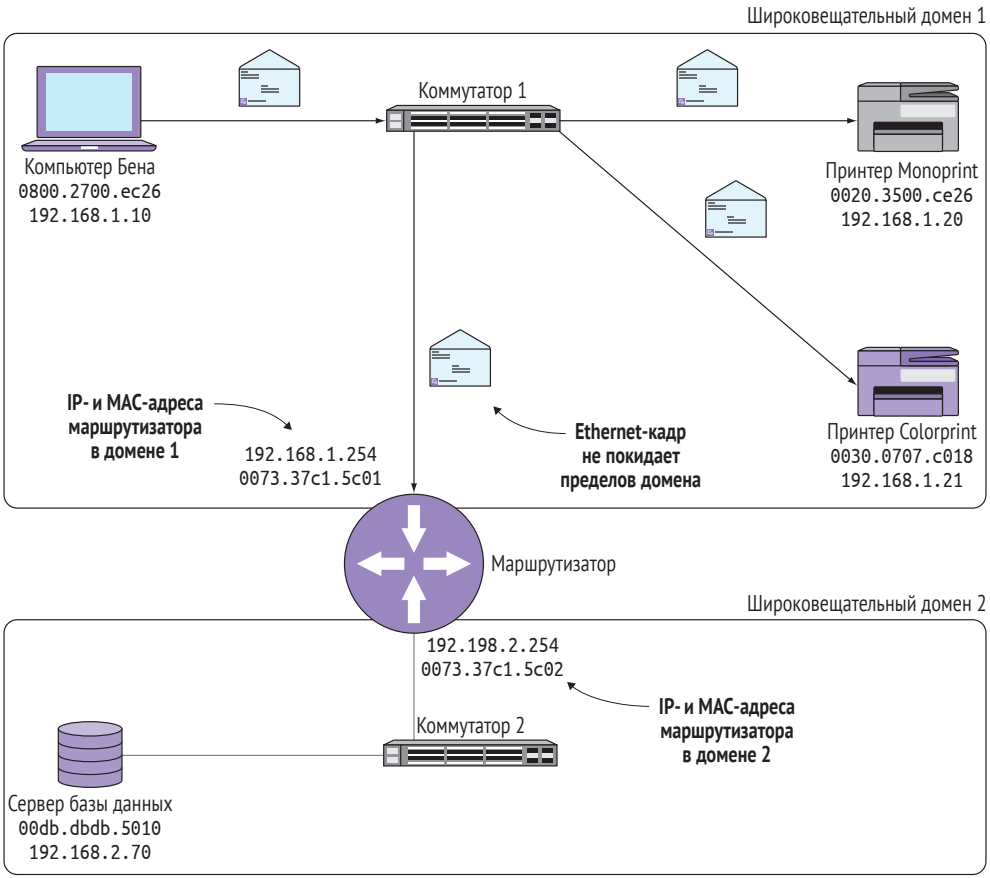

Рис. 2.10 ❖ Маршрутизатор подключен к двум широковещательным доменам. Для каждого домена маршрутизатору назначены уникальные IP- и MAC-адреса. Обратите внимание, что Ethernet-кадр не покидает пределов домена.

Для примера рассмотрим подсеть `192.168.1.`*x*. В этом наборе адресов нет ничего, что бы говорило: «Все адреса от `192.168.1.1` до `192.168.1.255` принадлежат одному домену!» Если вы об этом уже подумали, то, вероятно, эта идея пришла к вам на основании прочитанного в этой главе, а не от того, что вы рассмотрели сам адрес. Но мой компьютер не может читать и понимать как человек, поэтому ему нужны явные указания на принадлежность адреса определенному домену.

Для этого используется *маска подсети*. Маска подсети – это четыре восьмеричных числа, формируемых как IP-адрес, и именно она указывает на диапазон адресов, которые принадлежат одному домену.

На рис. 2.11 показано, что мой компьютер имеет IP-адрес `192.168.1.10`, а маска подсети – `255.255.255.0`. В первой и второй строках табл. 2.2 приведено сравнение каждого октета. Значение `255` в маске подсети означает, что IP-адрес, у которого значение соответствующего октета равно этому значению, принад-

лежит этой подсети. Значение `0` означает, что величина соответствующего октета в IP-адресе не имеет значения для данной подсети.

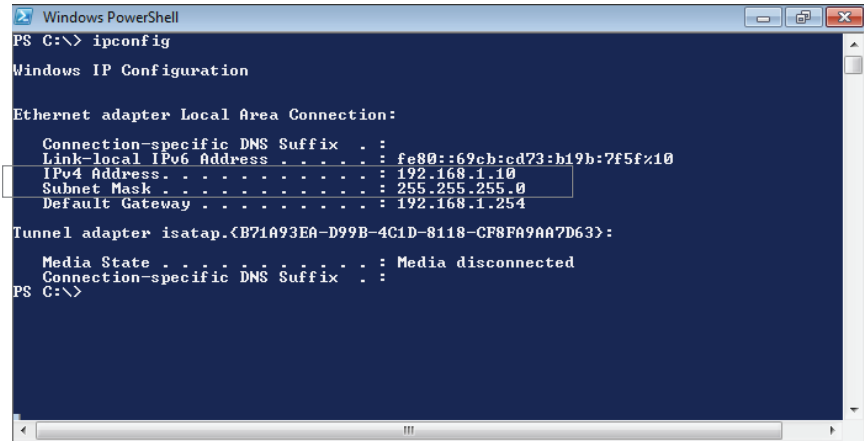

Рис. 2.11 ❖ Команда `ipconfig` выводит на экран информацию о настройках протокола IP на моем компьютере

Таблица 2.2. Определение домена по IP-адресу и маске подсети

IP-адрес моего компьютера	192	168	1	10
Маска подсети	255	255	255	0

IP-адрес моего компьютера и маска подсети сами по себе совершенно бесполезны. Вопрос на засыпку: принадлежит ли IP-адрес `192.168.2.70` тому же домену, что и мой компьютер? Давайте возьмем этот IP-адрес и проанализируем его, как показано в табл. 2.3.

Таблица 2.3. IP-адрес сервера базы данных отличается от IP-адреса моего компьютера

IP-адрес моего компьютера	192	168	1	10
Маска подсети	255	255	255	0
IP-адрес сервера базы данных	192	168	2	70

Первые два октета совпадают, но третьи октеты различны. И в силу того, что соответствующий октет маски подсети равен `255`, мой компьютер уже знает, что сервер базы данных находится в другом домене. Поэтому для обращения к серверу базы данных нужно использовать маршрутизатор. Но, прежде чем использовать маршрутизатор, он должен знать, что он существует и как к нему обратиться.

Практикум

Запустите оболочку командной строки Windows и введите команду `ipconfig`. Вы увидите IP-адрес и маску подсети. Проанализируйте известные вам IP-адреса, ко-

торые использует ваша компания. Определите, принадлежат ли они тому же домену, что и ваш компьютер.

2.7. Пересылка между доменами с использованием шлюза по умолчанию

Теперь, когда мой компьютер определил, что сервер базы данных принадлежит другому домену, ему нужно знать, какой маршрутизатор использовать для соединения с этим сервером. Для этого он проверяет адрес его шлюза по умолчанию.

Адрес шлюза по умолчанию моего компьютера `192.168.1.254`. Он соответствует IP-адресу интерфейса маршрутизатора, который подключен к его домену. Основываясь на адресе шлюза по умолчанию, мой компьютер знает, что когда ему нужно отправить что-либо на IP-адрес вне своего собственного широковещательного домена, он должен передать это сообщение через маршрутизатор.

Дополнительно

Отмечу, что IP-адрес моего компьютера – `192.168.1.10`, а IP-адрес шлюза по умолчанию принадлежат одной подсети. Это действительно важно. Если устройство не принадлежит той же подсети, что и маршрутизатор, оно не может обратиться ни к кому вне своего домена.

Мой компьютер посылает ARP-запрос по адресу `192.168.1.254`, и маршрутизатор в ответе сообщает, что его собственный MAC-адрес – `0073.37c1.5c01`. Мой компьютер собирает Ethernet-кадр и отправляет его на MAC-адрес маршрутизатора. Но в то же время он собирает «конверт» поменьше, называемый *IP-пакетом*. Если Ethernet-кадр – это большой конверт с MAC-адресами, то IP-пакет – конверт поменьше, содержащий лишь IP-адреса отправителя и получателя.

IP-пакет содержит IP-адрес моего компьютера `192.168.1.10` (отправителя) и IP-адрес `192.168.2.70` сервера базы данных (получателя). На рис. 2.12 показано, как мой компьютер размещает этот малый конверт – IP-пакет – внутри большого конверта – Ethernet-кадра, в котором MAC-адрес маршрутизатора указан как адрес получателя. Этот процесс «наполнения Ethernet-конверта» называется *инкапсуляцией*.

Мой компьютер отсылает маршрутизатору Ethernet-кадр, который содержит IP-пакет. Маршрутизатор, получив Ethernet-кадр, извлекает IP-пакет и видит IP-адрес получателя. Маршрутизатор определяет, что `192.168.2.70` – это адрес, принадлежащий домену `2`, который подключен к одному из его интерфейсов.

Тогда он отправляет ARP-запрос серверу, используя его IP-адрес `192.168.2.70`. `ARP`-запрос говорит: «Это `192.168.2.254`. У кого `192.168.2.70`?» Сервер отвечает, указывая свой MAC-адрес, а маршрутизатор берет IP-пакет и вкладывает его в новый Ethernet-кадр, в котором уже содержится MAC-адрес сервера как получателя. Все это отражено на рис. 2.13.

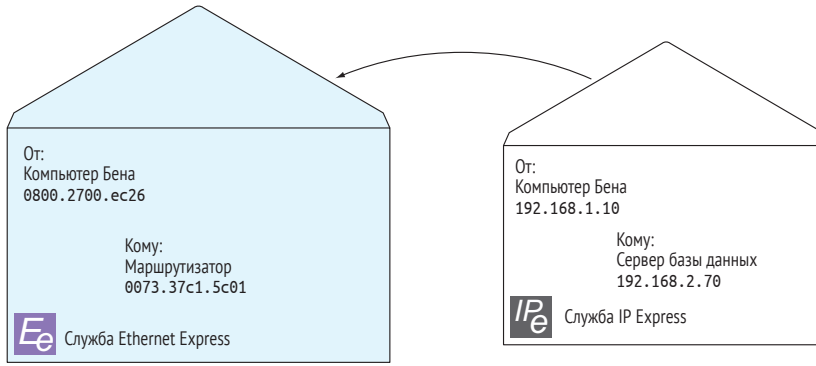

Рис. 2.12 ❖ IP-пакет инкапсулирован в Ethernet-кадр

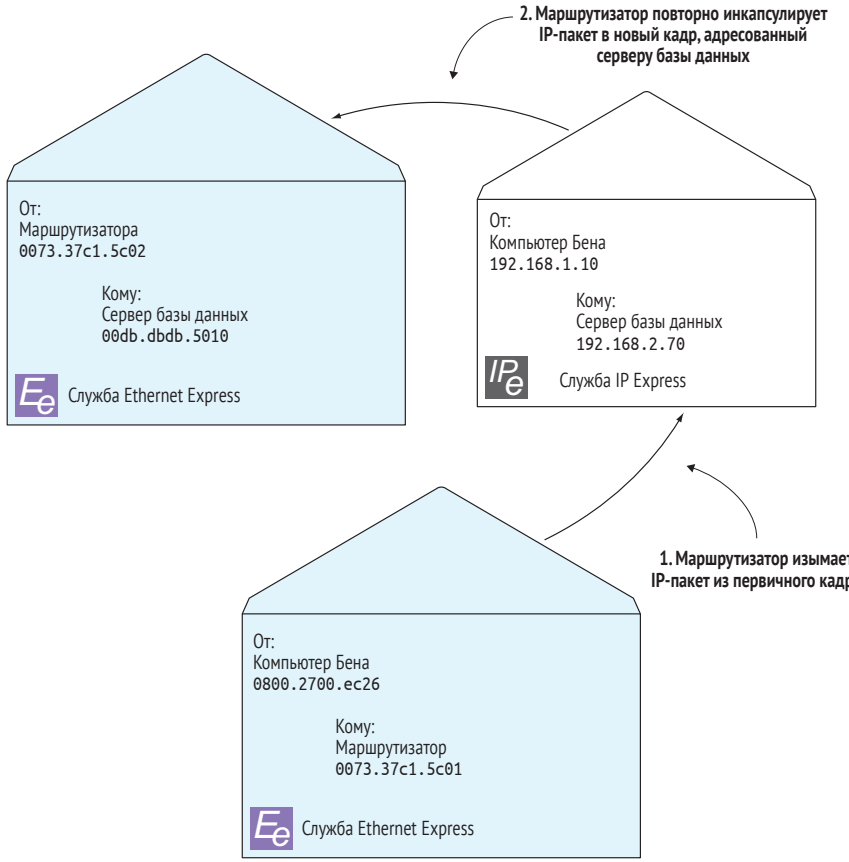

Рис. 2.13 ❖ Маршрутизатор повторно инкапсулирует IP-пакет, переданный моим компьютером

На шаге 1 маршрутизатор удаляет (декапсулирует) IP-пакет из первичного кадра. На шаге 2 маршрутизатор повторно инкапсулирует пакет в новый кадр, адресованный серверу базы данных.

Отмечу, что сам IP-пакет никогда не изменяется в процессе пересылки. Маршрутизатор сохраняет оба IP-адреса, как отправителя, так и получателя, и заменяет только MAC-адреса в Ethernet-кадре. Далее он отправляет новый Ethernet-кадр серверу. Сервер, получив его, извлекает IP-пакет и говорит: «Эй! Я и есть `192.168.2.70`! Этот пакет предназначен для меня».

Рисунок 2.14 иллюстрирует, как маршрутизатор пересылает пакет за пределы домена, скрывая MAC-адреса одного домена от устройств другого. Этот процесс называется *IP-маршрутизацией*.

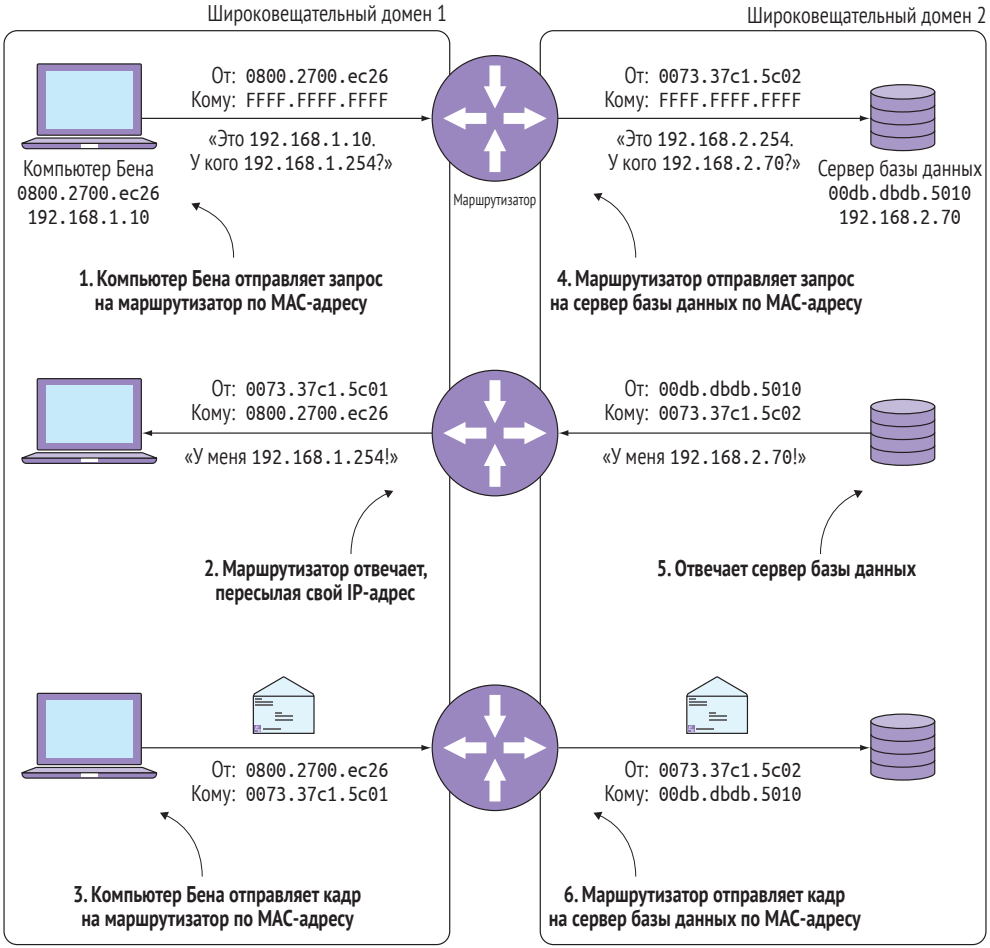

Рис. 2.14 ❖ Использование маршрутизатора для пересылки информации между доменами

На шаге 1 мой компьютер отправляет ARP-запрос, чтобы получить MAC-адрес. На шаге 2 маршрутизатор отправляет ARP-ответ, содержащий его IP-адрес. На шаге 3 мой компьютер пересылает кадр, адресованный маршрутизатору, по его MAC-адресу (`0073.37c1.5c01`). Кадр содержит IP-пакет, адресованный серверу базы данных (`192.168.2.70`). На шаге 4 маршрутизатор отправляет ARP-запрос для получения MAC-адреса сервера. На шаге 5 сервер отправляет ARP-ответ. Наконец, на шаге 6 маршрутизатор отправляет кадр, адресованный серверу базы данных, по его MAC-адресу (`00db.dbdb.5010`); пересылаемый кадр содержит оригинальный IP-пакет.

Пришла пора подвести итоги. На рис. 2.15 показано, как IP-пакет проходит весь путь от моего компьютера до сервера базы данных без лавинной передачи на все устройства.

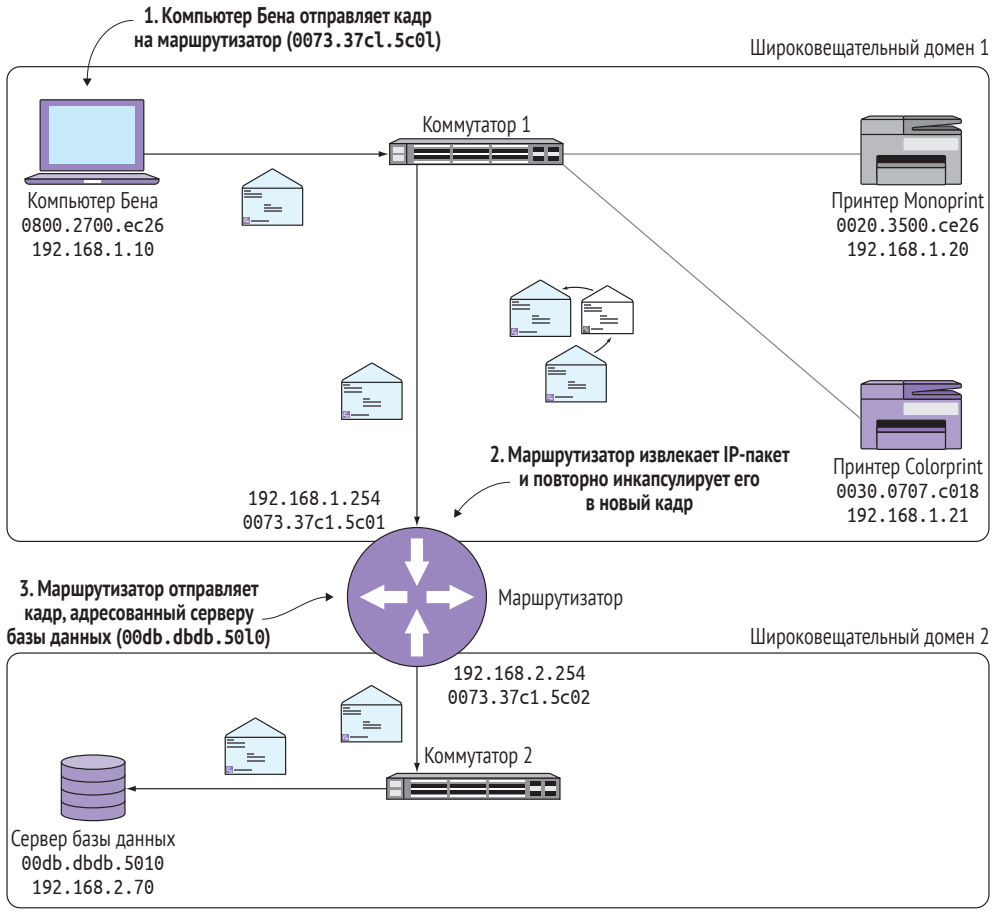

Рис. 2.15 ❖ Использование маршрутизации и коммуникации для пересылки IP-пакета между доменами без лавинной передачи

На шаге 1 мой компьютер инкапсулирует IP-пакет в кадр, адресованный маршрутизатору. Кадр пересылается на Коммутатор 1, который пересылает его на Коммутатор 2. На шаге 2 маршрутизатор удаляет IP-пакет, видит IP-адрес получателя и инкапсулирует его в новый кадр, адресованный серверу базы данных. На шаге 3 маршрутизатор пересылает новый кадр на Коммутатор 2, который и пересылает его на сервер.

2.8. Управление маршрутизаторами и коммутаторами

Теперь у вас есть базовое понимание роли маршрутизаторов и коммутаторов. Возможно, у вас уже зачесались руки потыкать в них пальцем и попытаться их конфигурировать. Но, прежде чем вы попробуете это, вам нужно получить к ним реальный доступ.

Маршрутизаторы и коммутаторы имеют собственные IP-адреса. Для назначения каждому маршрутизатору и коммутатору IP-адреса обычно используют специальный менеджер IP-адресов. Такой менеджер позволяет производить администрирование устройств дистанционно, не имея к ним физического доступа. Маршрутизаторы и коммутаторы вашей компании почти наверняка закрыты на ключ, где-нибудь в каморке дата-центра, и даже если вы получите к ним доступ, конфигурировать их непосредственно в ручном режиме довольно затруднительно. Вот зачем нужно иметь менеджер IP-адресов и пароль, предоставляющий полномочия администратора для конфигурирования всех устройств. Убедитесь, что вы сможете это сделать до завтрашнего урока.

2.9. Практическое задание

Загрузите файл *Inventory worksheet.xlsx* с сайта www.manning.com/books/learn-cisco-network-administration-in-a-month-of-lunches. Откройте менеджер IP-адресов и перепишите IP-адреса всех маршрутизаторов и коммутаторов вашей компании (или вашей тестовой сети). Получите также данные для авторизации (имя пользователя (логин) и пароль), предоставляющие полномочия администратора для каждого устройства сети.

На вашем компьютере запустите оболочку командной строки. Узнайте MAC-адрес, IP-адрес и адрес шлюза по умолчанию вашего компьютера, выполнив команду `ipconfig /all`. Введите команду `arp -a` и узнайте MAC-адрес шлюза по умолчанию. Внесите полученную информацию в файл *Inventory worksheet.xlsx*.

Глава 3

Краткий курс по операционной системе Cisco IOS

Если вы уже используете графический пользовательский интерфейс (GUI) для системного администрирования, то администрирование сети Cisco не вызовет затруднений. Но, несмотря на то что сотрудники компании Cisco приложили мало усилий к разработке конфигурационных утилит «в один щелчок», интерфейс командной строки (command-line interface, CLI) остается доступным и сохраняет свою значимость. Это мощный и эффективный инструмент, если вы хорошо понимаете, как работают команды, которые вводите. Интерфейс командной строки черно-белый, в отличие от графического интерфейса с его красочными кнопками и удобными для пользователя уведомлениями.

Если вы не привыкли использовать оболочку командной строки на других платформах, таких как Windows или Linux, не беспокойтесь. Во многих отношениях интерфейс командной строки в IOS прост, потому что оснащен встроенной справкой, пользоваться которой я вас научу.

3.1. Что такое IOS?

Подавляющее большинство маршрутизаторов и коммутаторов Cisco работает под управлением операционной системы Cisco Internetwork Operating System (IOS). Система IOS контролирует все аспекты работы устройства, например список тех, кому разрешен доступ, какой трафик допустим, а какой заблокирован, включен/отключен интерфейс и т. д.

Оболочка командной строки IOS – это интерфейс для настройки устройств Cisco. Вы будете выполнять все задачи по настройке, описанные в этой книге, именно в оболочке командной строки IOS. В этой главе вы узнаете, как получить доступ к оболочке командной строки IOS и как использовать ее для просмотра, изменения и сохранения конфигураций устройств.

Система IOS – это настолько мощный инструмент, что вы легко можете нарушить работу всей сети, просто выполнив неправильную команду. Команды, которые вы изучите в этой главе, довольно безобидны, поэтому вы можете вполне безопасно выполнять их все (с разрешения, конечно) в рабочей сети. Просто имейте в виду, что ввод случайных команд может быть катастрофическим, поэтому не увлекайтесь. Если у вас есть тестовая сеть, работу которой вы можете нарушить без последствий, не стесняйтесь экспериментировать, но не скачите хаотично по материалу этой книги. Даже в этом случае следует учиться методично, по главам.

Дополнительно

IOS-XE – еще одна операционная система Cisco. Системы IOS и IOS-XE основаны на разных программных архитектурах, но команды практически идентичны. Все команды, описанные в этой книге, должны отлично работать и в системе IOS-XE.

3.2. Авторизация на устройствах Cisco

Независимо от того, подключаетесь вы к маршрутизатору или коммутатору Cisco, процесс в целом одинаков. Во-первых, вам нужен клиентский терминал, который поддерживает как протокол Telnet, так и Secure Shell (SSH). Для пользователей операционной системы Windows я рекомендую программу PuTTY, которую вы можете скачать с сайта www.chiark.greenend.org.uk/~sgtatham/putty/download.html. Если вы поклонник операционной системы macOS или Linux, то можете использовать команды `telnet` или `ssh` в программе Terminal (Терминал). В примерах в этой книге я использую операционную систему Windows и программу PuTTY, но скажу, что как только вы подключитесь к устройству Cisco, ни ваша операционная система, ни клиентский терминал уже не будут иметь большого значения. Команды конфигурации, которые вы будете использовать для настройки устройств, будут одинаковыми для любой системы.

Откройте клиентский терминал и выберите один из коммутаторов вашей сети. Я собираюсь подключиться к учебному коммутатору, набрав его IP-адрес в поле **Host Name (or IP address)** (Имя хоста (или IP-адрес)), установив тип подключения **Telnet** и нажав кнопку **Open** (Открыть), как показано на рис. 3.1. Если вы не можете подключиться через протокол Telnet, попробуйте использовать протокол SSH. Интерфейс программы PuTTY не менялся годами, но если это когда-либо произойдет в вашей версии, базовые настройки все равно должны быть теми же.

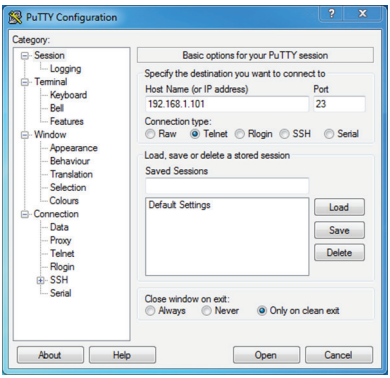

Рис. 3.1 ❖ Окно конфигурации программы PuTTY. В поле **Host Name (or IP address)** введите IP-адрес устройства, к которому вы хотите подключиться. Установите переключатель в положение **Telnet** или **SSH**, а затем нажмите кнопку **Open**

При появлении запроса введите имя пользователя (логин) и пароль привилегированного пользователя. Вы должны увидеть имя хоста коммутатора, за которым следует либо хеш (#), либо знак больше (>):

```
User Access Verification

Username: admin
Password:
Switch1>enable   ◄— Введите слово enable, чтобы войти в привилегированный режим EXEC.
Password:
Switch1#         ◄— Символ # указывает, что вы находитесь в привилегированном режиме EXEC.
```

Если вы не видите символ #, введите слово `enable` и нажмите клавишу **Enter**. Возможно, вам будет предложено ввести еще один пароль для режима enable. Если вход в систему выполнен успешно, вы должны увидеть приглашение с символом #. Сотрудники компании Cisco называют этот привилегированный режим как EXEC, но многие люди называют его режимом enable. Режим enable – это режим root-пользователя или администратора, который позволяет просматривать более подробную информацию о коммутаторе и вносить изменения в его конфигурацию.

Практикум

Авторизуйтесь на одном из ваших коммутаторов 3-го уровня. Убедитесь, что вы можете перейти в режим enable. В противном случае не читайте дальше. У вас должна быть возможность войти в режим enable на всех устройствах, иначе не сможете управлять своей сетью. Самое серьезное препятствие для перехода в режим enable – ввод неправильного пароля при авторизации. Убедитесь, что вы напечатали его правильно!

Имейте в виду, что в зависимости от индивидуальных настроек коммутатора вы можете выйти из режима администрирования автоматически, по истечении определенного времени бездействия. Это важный параметр безопасности, и подобное поведение не указывает на что-то неправильное в ваших настройках. Если это произойдет, просто заново авторизуйтесь в системе и вернитесь к моменту, где вы остановились.

3.3. Команда show

Команда show используется наиболее часто. Она позволяет вывести практически любую информацию об устройстве, на котором вы авторизовались. В оболочке командной строки введите show ?. Встроенная справочная система должна заполнить экран внушительным списком команд, которые могут рассказать вам о различных аспектах устройства. Этот список состоит из нескольких экранов, и на каждом экране отображается строка --More--. Нажмите клавишу **Пробел**, чтобы перейти к следующему экрану. Затем к следующему и так далее. Продолжайте нажимать клавишу **Пробел**, пока не вернетесь к приглашению командной строки. Кроме того, вы можете нажать любую клавишу (кроме **Enter** или **Пробел**), чтобы выйти из встроенной справочной системы и вернуться к приглашению:

```
Switch1#show ?
  aaa                 Show AAA values
  access-expression   List access expression
  access-lists        List access lists
  adjacency           Adjacent nodes
  aliases             Display alias commands
  …
  vtp                 VTP information
  wsma                Show Web Services Management Agents information
  xdr                 Show details about XDR
  xos                 Cross-OS Library Information and Traces
  xsd-format          Show the ODM XSD for the command

Switch1#show
```

Обратите внимание, что отображается две колонки информации. В левой колонке указаны имена команд, а в правой – краткие описания команд.

Для большинства команд show требуется одна или несколько подкоманд. Например, если вы набрали show ip, после нажатия клавиши **Enter** вы получите сообщение об ошибке:

```
Switch1#show ip
% Incomplete command.

Switch1#
```

Оно означает, что для просмотра информации об IP-адресе требуется подкоманда. Принимая во внимание большое количество команд show, необоснованно ожидать, что вы запомните их все или даже большинство из них. К счастью,

вам это не нужно. Если вы введете команду show ip ?, появится еще один не очень длинный список подкоманд. Обратите внимание, что этот список отличается от предыдущего. Он короче и содержит только подкоманды, связанные с IP-адресами:

```
Switch1#show ip ?
  access-lists         List IP access lists
  accounting           The active IP accounting database
  admission            Network Admission Control information
  aliases              IP alias table
  arp                  IP ARP table
  as-path-access-list  List AS path access lists
  auth-proxy           Authentication Proxy information
  bgp                  BGP information
  cache                IP fast-switching route cache
  cef                  Cisco Express Forwarding
  community-list       List community-list
  device               Show IP Tracking Hosts
  dhcp                 Show items in the DHCP database
  eigrp                Show IPv4 EIGRP
  extcommunity-list    List extended-community list
  flow                 NetFlow switching
  host                 IP host information
  http                 HTTP information
  igmp                 IGMP information
  interface            IP interface status and configuration
  irdp                 ICMP Router Discovery Protocol
  local                IP local options
--More--
```

Если вам показалось, что первый список команд show был слишком велик и непонятен, то *общее* количество возможных команд show намного ужаснее. К счастью, есть только несколько команд, которые вам действительно нужно запомнить. Давайте посмотрим на одну из них.

В первом списке подкоманд show ip, примерно наполовину страницы, вы видите слово interface, за которым следует текст IP interface status and configuration.

Выполните команду interface ? (полная команда show ip interface ?). Теперь вы получите гораздо более короткий список, состоящий в основном из таких типов интерфейсов, как FastEthernet и GigabitEthernet, с последующим их кратким описанием:

```
Switch1#show ip interface ?
  Async              Async interface
  Auto-Template      Auto-Template interface
  BVI Bridge-Group   Virtual Interface
  CTunnel            CTunnel interface
  Dialer             Dialer interface
  FastEthernet       FastEthernet IEEE 802.3
  Filter             Filter interface
```

```
  Filtergroup        Filter Group interface
  GigabitEthernet    GigabitEthernet IEEE 802.3z
  GroupVI            Group Virtual interface
  Lex                Lex interface
  Loopback           Loopback interface
  Null               Null interface
  Port-channel       Ethernet Channel of interfaces
  Portgroup          Portgroup interface
  Pos-channel        POS Channel of interfaces
  Tunnel             Tunnel interface
  Vif                PGM Multicast Host interface
  Virtual-Template   Virtual Template interface
  Virtual-TokenRing  Virtual TokenRing
  Vlan Catalyst      Vlans
  brief              Brief summary of IP status and configuration
--More-
```

Выполните команду `brief ?` – и вы увидите список, почти идентичный предыдущему:

```
Switch1#show ip interface brief ?
  Async              Async interface
  Auto-Template      Auto-Template interface
  BVI Bridge-Group   Virtual Interface
  CTunnel            CTunnel interface
  Dialer             Dialer interface
  FastEthernet       FastEthernet IEEE 802.3
  Filter             Filter interface
  Filtergroup        Filter Group interface
  GigabitEthernet    GigabitEthernet IEEE 802.3z
  GroupVI            Group Virtual interface
  Lex                Lex interface
  Loopback           Loopback interface
  Null               Null interface
  Port-channel       Ethernet Channel of interfaces
  Portgroup          Portgroup interface
  Pos-channel        POS Channel of interfaces
  Tunnel             Tunnel interface
  Vif                PGM Multicast Host interface
  Virtual-Template   Virtual Template interface
  Virtual-TokenRing  Virtual TokenRing
  Vlan Catalyst      Vlans
  fcpa               Fiber Channel
  |                  Output modifiers
  <cr>                    ◄— Указывает, что вы можете нажать клавишу Enter, чтобы выполнить команду.
```

Обратите внимание на оператор в конце последней строки, `<cr>`. Он обозначает «возврат каретки», который является причудливым термином для клавиши **Enter**. `<cr>` указывает, что вы можете нажать клавишу **Enter** без добавления дополнительных подкоманд. Это удобная подсказка, что команда `show`, которую вы ввели, вероятно, будет работать без ошибок. Если вы не видите строку

<cr> в нижней части списка, это означает, что перед нажатием клавиши **Enter** вам нужно указать дополнительные подкоманды. Нажмите клавишу **Enter** на команде `show ip interface brief`.

Вы должны увидеть список всех интерфейсов коммутатора вместе со всеми назначенными IP-адресами. Ваш интерфейс `Vlan1` имеет назначенный IP-адрес `192.168.1.101`. Не кажется ли он вам знакомым? Это IP-адрес, к которому вы подключены!

```
Switch1#show ip interface brief
Interface              IP-Address      OK? Method  Status    Protocol
Vlan1                  192.168.1.101   YES NVRAM   up        up      <-- Это интерфейс и IP-адрес,
FastEthernet0/1        unassigned      YES unset   up        up          к которому я подключен.
FastEthernet0/2        unassigned      YES unset   down      down
FastEthernet0/3        unassigned      YES unset   down      down
FastEthernet0/4        unassigned      YES unset   down      down
FastEthernet0/5        unassigned      YES unset   down      down
FastEthernet0/6        unassigned      YES unset   down      down
FastEthernet0/7        unassigned      YES unset   down      down
FastEthernet0/8        unassigned      YES unset   down      down
FastEthernet0/9        unassigned      YES unset   down      down
FastEthernet0/10       unassigned      YES unset   down      down
FastEthernet0/11       unassigned      YES unset   down      down
FastEthernet0/12       unassigned      YES unset   down      down
FastEthernet0/13       unassigned      YES unset   down      down
FastEthernet0/14       unassigned      YES unset   down      down
FastEthernet0/15       unassigned      YES unset   down      down
FastEthernet0/16       unassigned      YES unset   down      down
FastEthernet0/17       unassigned      YES unset   down      down
FastEthernet0/18       unassigned      YES unset   down      down
FastEthernet0/19       unassigned      YES unset   down      down
FastEthernet0/20       unassigned      YES unset   down      down
--More--
```

Практикум

Выполните команду `show ip interface brief`. Найдите интерфейс и IP-адрес, который вы использовали для подключения к коммутатору.

3.3.1. Фильтрация вывода

Команды `show` могут выдавать много результатов, и если вы ищете только одну или две нужные строки на экране, заполненном данными, это может стать довольно трудоемким делом. Команды `include` и `exclude` представляют собой две команды синтаксического анализа, которые позволяют фильтровать вывод команды `show`, чтобы отображать только строки, которые вам нужны.

Включение строк

Выполните команду `show ip interface brief ?`. Во второй с конца строке вы должны увидеть символ «пайп» `|`, описанный как модификатор вывода. Он обычно

используется в сценарных и командных файлах для передачи или перенаправления вывода из одной команды в другую. В IOS команда «пайп» имеет аналогичную функцию – для ввода-вывода в одну из встроенных функций синтаксического анализа IOS:

```
Switch1#show ip interface brief ?
  Async              Async interface
  Auto-Template      Auto-Template interface
  BVI                Bridge-Group Virtual Interface
  CTunnel            CTunnel interface
  Dialer             Dialer interface
  FastEthernet       FastEthernet IEEE 802.3
  Filter             Filter interface
  Filtergroup        Filter Group interface
  GigabitEthernet    GigabitEthernet IEEE 802.3z
  GroupVI            Group Virtual interface
  Lex                Lex interface
  Loopback           Loopback interface
  Null               Null interface
  Port-channel       Ethernet Channel of interfaces
  Portgroup          Portgroup interface
  Pos-channel        POS Channel of interfaces
  Tunnel             Tunnel interface
  Vif                PGM Multicast Host interface
  Virtual-Template   Virtual Template interface
  Virtual-TokenRing  Virtual TokenRing
  Vlan               Catalyst Vlans
  fcpa               Fiber Channel
  |                  Output modifiers      ← Перенаправляет вывод на другую команду.
  <cr>
```

Нажмите клавишу **Enter** после ввода команды `show ip interface brief`:

```
Switch1#show ip interface brief
Interface              IP-Address      OK? Method  Status  Protocol
Vlan1                  192.168.1.101   YES NVRAM   up      up
FastEthernet0/1        unassigned      YES unset   up      up
FastEthernet0/2        unassigned      YES unset   down    down
FastEthernet0/3        unassigned      YES unset   down    down
FastEthernet0/4        unassigned      YES unset   down    down
FastEthernet0/5        unassigned      YES unset   down    down
FastEthernet0/6        unassigned      YES unset   down    down
FastEthernet0/7        unassigned      YES unset   down    down
FastEthernet0/8        unassigned      YES unset   down    down
FastEthernet0/9        unassigned      YES unset   down    down
FastEthernet0/10       unassigned      YES unset   down    down
FastEthernet0/11       unassigned      YES unset   down    down
FastEthernet0/12       unassigned      YES unset   down    down
FastEthernet0/13       unassigned      YES unset   down    down
FastEthernet0/14       unassigned      YES unset   down    down
FastEthernet0/15       unassigned      YES unset   down    down
FastEthernet0/16       unassigned      YES unset   down    down
```

```
FastEthernet0/17     unassigned     YES unset   down    down
FastEthernet0/18     unassigned     YES unset   down    down
FastEthernet0/19     unassigned     YES unset   down    down
FastEthernet0/20     unassigned     YES unset   down    down
FastEthernet0/21     unassigned     YES unset   down    down
FastEthernet0/22     unassigned     YES unset   down    down
FastEthernet0/23     unassigned     YES unset   down    down
FastEthernet0/24     unassigned     YES unset   down    down
GigabitEthernet0/1   unassigned     YES unset   down    down
GigabitEthernet0/2   unassigned     YES unset   down    down
```

Вы должны увидеть много портов FastEthernet и пару портов GigabitEthernet, но вам они не нужны. Вам нужен интерфейс `Vlan1`. Если вы не хотите видеть все 28 строк, то можете включить только строки с поисковым запросом `Vlan`, используя команду `show ip interface brief | include Vlan`:

```
Switch1#show ip interface brief | include Vlan
Vlan1                  192.168.1.101   YES NVRAM  up    up
Switch1#
```

Обратите внимание, что вместо 28 строк вы получаете одну, содержащую точную информацию, которую хотите видеть.

Команда включает в себя слово `Vlan` с прописной буквы *V*. Когда дело доходит до фильтрации, система IOS учитывает регистр. Команда `show ip interface brief | include vlan` ничего не отобразит, потому что указан интерфейс `vlan` со строчной буквой *v*.

Исключение строк

Предположим, что вы хотите видеть информацию об IP-адресах всех ваших портов, кроме FastEthernet. Для этого вы можете использовать ключевое слово `exclude`, чтобы исключить любые строки, содержащие поисковый запрос `Fast`.

Введите команду `show ip interface brief | exclude Fast` и нажмите клавишу **Enter**. Теперь вы увидите только четыре строки вывода, показывающие все порты, кроме FastEthernet:

```
Switch1#show ip interface brief | exclude Fast
Interface              IP-Address      OK? Method Status    Protocol
Vlan1                  192.168.1.101   YES NVRAM  up        up
GigabitEthernet0/1     unassigned      YES unset  down      down
GigabitEthernet0/2     unassigned      YES unset  down      down
Switch1#
```

Дополнительно

Система IOS поддерживает в поисковых запросах регулярные выражения. Регулярные выражения предоставляют способ задания сложных поисковых запросов. Если вы хотите одновременно указать несколько параметров поиска, то можете указать между ними символ `|`. Например, если вы хотите просмотреть все строки с интерфейсами `Fast` и `Giga`, то можете ввести команду `show ip interface brief | include Fast|Gig`.

Практикум

Выполните пробную фильтрацию с помощью показанных ниже команд `show ip`. Эти команды безопасны и не нарушат работу коммутатора:

```
show ip interface | include up|Internet
show ip interface brief | exclude down
```

Кроме того, попробуйте найти команду, которая покажет вам подробную информацию обо всех интерфейсах. Отфильтруйте выводимый список, чтобы включить только строки с термином `address`.

3.4. Идентификация версии и пакета IOS

Хотя пользователи в разговорной речи относятся к Cisco IOS как к единой операционной системе, на самом деле существует много разных платформ IOS. Вам не нужно знать все различия между ними, но необходимо уметь определять, какую платформу использует ваш коммутатор или маршрутизатор, чтобы вы могли определить, какие функции доступны.

Платформа Cisco IOS зависит от модели устройства, версии и пакета. Вы можете просмотреть эту информацию, выполнив команду `show version | include IOS`:

```
Switch1#show version | include IOS
Cisco IOS Software, C3560 Software (C3560-IPSERVICESK9-M), Version 15.0(2)SE5, RELEASE SOFTWARE (fc1)
Switch1#
```

В моем примере используется платформа C3560 – система IOS для управления коммутатором Catalyst 3560. Информация о платформе не станет для вас неожиданностью, потому что вы уже знаете, подключены вы к маршрутизатору или коммутатору. Давайте рассмотрим более интересные моменты: версию и пакет.

3.4.1. Номера версий

Мой коммутатор работает под управлением версии 15.0(2)SE5. 15 – основной релиз версии, .0 – младший релиз, а (2) – номер релиза в рамках версии. Вам не нужно знать различия между ними, но если вы когда-нибудь будете иметь дело с технической поддержкой Cisco и специалист спросит: «Что у вас установлено?», – это полезно знать.

Буквы SE указывают на идентификатор платформы/семейства, а 5 – номер ребилда. Значение SE указывает, что данная версия IOS предназначена для коммутаторов Cisco Catalyst. Это может показаться излишним, потому что C3560 и так указывает непосредственно на модель коммутатора.

Релизы 12 и 15 наиболее распространены среди основных версий (на момент работы над этой книгой). Компания Cisco пропустила номера 13 и 14, потому что они считают эти числа невезучими. Что касается модификаций основных

версий, то вы можете встретить и старые версии. Я, к примеру, недавно настраивал коммутатор с системой версии 12.1, который отработал без перерыва семь лет!

Вероятно, вы не увидите версий более ранних, чем 12.0, но это возможно. Ранние версии устарели по технологическим соображениям, и если вы работаете в среде, более ранней, чем 12.0, я рекомендую поскорее выбираться оттуда!

Дополнительно

Обновление системы IOS выходит за рамки этой книги, поскольку это не рутинная административная операция. Если вам когда-либо понадобится обновление, вам лучше позвать на помощь квалифицированного сисадмина. Модернизация платформы IOS на корпоративном оборудовании может быть сложна даже для специалистов, сертифицированных Cisco.

3.4.2. Пакеты

Пакеты также известны как наборы функций, поскольку определяют, какие функции доступны. Существует три общих пакета, с которыми вы можете столкнуться:

- базовый IP-функционал (IPBASE);
- расширенный IP-функционал (IPSERVICES);
- расширенный корпоративный функционал (ADVENTERPRISE).

Давайте определим пакет, который использую я:

```
Cisco IOS Software, C3560 Software (C3560-IPSERVICESK9-M), Version 15.0(2)SE5, RELEASE SOFTWARE (fc1)
```

Строка, заключенная в круглые скобки, `C3560-IPSERVICESK9-M`, говорит о том, что у меня установлен пакет расширенного корпоративного функционала. Этот пакет содержит больше функциональных возможностей, чем IPBASE, и включает все возможные функции.

Практикум

Выполните команду `show version` для просмотра версии системы IOS на каждом из используемых в вашей сети коммутаторов. Убедитесь, что вам доступен хотя бы расширенный IP-функционал. Вам не обязательно нужна версия IOS 15, но чем выше версия, тем лучше.

3.5. Просмотр рабочей конфигурации

Во время нормальной работы устройства Cisco сохраняют большинство настроек конфигурации в оперативной памяти (ОЗУ). Это так называемая *рабочая конфигурация*. Рабочая конфигурация – это те настройки системы IOS, которые используются в режиме реального времени. Следовательно, любые изменения, внесенные вами в рабочую конфигурацию, вступают в силу почти

сразу. Например, если вы измените IP-адрес коммутатора, это изменение будет сохранено в рабочей конфигурации и вступит в силу немедленно.

Рабочая конфигурация представляет собой длинную строку текста – текстовый файл, разделенный на различные разделы, которые управляют различными аспектами устройства. Когда вы познакомитесь со своей сетью и начнете вносить в нее изменения, вам будет нужно знать, как найти и просмотреть каждый из этих разделов. Чтобы просмотреть всю рабочую конфигурацию, выполните команду `show running-config`:

```
Switch1#show running-config
Building configuration...
Current configuration : 3069 bytes
!
version 15.0
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Switch1
!
boot-start-marker
boot-end-marker
!
!
!
username admin privilege 15 secret 5 $1$r/gI$sNjAw2i0L1Syobws.5tzT1
no aaa new-model
system mtu routing 1500
vtp domain cisco
vtp mode transparent
!
!
--More--
```

Вы можете сокращать команды IOS, чтобы не печатать их полностью. Например, вы можете ввести сокращенную команду `sh run` вместо полной `show running-config` и получить тот же результат. Встроенная справочная система не содержит список сокращений, но если введенная команда не покажется системе двусмысленной, IOS разберется, что вы имеете в виду.

В вашем случае конфигурация может быть огромной или, наоборот, состоять всего из несколько экранов. Продолжайте нажимать клавишу **Пробел**, пока не доберетесь до раздела, в котором перечислены интерфейсы:

```
!
interface FastEthernet0/20
!
interface FastEthernet0/21
!
```

```
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
ip address 192.168.1.101 255.255.255.0
!
ip http server
ip http secure-server
!
!
!
--More--
```

В конце конфигурации вашего коммутатора должен отобразиться раздел с названием `interface Vlan1`. Обратите внимание, что следующая строка, содержащая IP-адрес, содержит отступ в один пробел, указывая, что это часть текущего раздела. Вы можете просмотреть только этот раздел, используя команду `show run | section Vlan1`:

```
Switch1#show run | section Vlan1
interface Vlan1
 ip address 192.168.1.101 255.255.255.0
Switch1#
```

Ключевое слово `section` – это еще один фильтр, который вы можете использовать для просмотра определенных разделов рабочей конфигурации. Вы также можете использовать ключевые слова `include` и `exclude`, если это необходимо.

Практикум

Просмотрите рабочую конфигурацию коммутатора. Выберите желаемый раздел конфигурации и отфильтруйте вывод, чтобы просмотреть только выбранный раздел.

3.6. Изменение рабочей конфигурации

Вы можете изменить рабочую конфигурацию «на лету». Нет необходимости перезагружать или иным образом фиксировать ваши изменения. Во многих случаях, как только вы вводите команду, она вступает в силу.

Вы будете настраивать свои устройства Cisco с помощью оболочки командной строки, которую в Cisco называют *терминалом*. Чтобы внести изменения в конфигурацию, вам необходимо войти в специальный режим, называемый

режимом глобальной конфигурации. Войдите в режим глобальной конфигурации, выполнив команду `configure terminal`:

```
Switch1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch1(config)#
```

Приглашение изменится на `Switch1(config)#`, указывая, что вы находитесь в режиме глобальной конфигурации. Здесь также поддерживается встроенная справочная система. Если вы наберете вопросительный знак (?), то увидите длинный список команд, аналогичный команде `show`:

```
Switch1(config)#?
Configure commands:
  aaa                 Authentication, Authorization and Accounting.
  access-list         Add an access list entry
  access-session      Access Session Global Configuration Commands
  alias               Create command alias
  archive             Archive the configuration
  arp                 Set a static ARP entry
  authentication      Auth Manager Global Configuration Commands
  auto                Configure Automation
  banner              Define a login banner
  beep                Configure BEEP (Blocks Extensible Exchange Protocol)
  boot                Modify system boot parameters
  bridge              Bridge Group.
  buffers             Adjust system buffer pool parameters
  call-home           Enter call-home configuration mode
  cdp Global          CDP configuration subcommands
  cef                 Cisco Express Forwarding
  cisp                Set CISP parameters
  class-map           Configure CPL Class Map
  clns                Global CLNS configuration subcommands
  clock               Configure time-of-day clock
--More--
```

Это описание не очень информативно, но встроенная справка может предоставить дополнительную информацию.

Чтобы продемонстрировать, каким образом изменения конфигурации вступают в силу, предположим, что вы хотите изменить баннер авторизации, который представляет собой строку текста, отображаемую IOS при входе в систему. Но вы еще не знаете, как работает команда `banner`. Если вы выполните команду `banner ?`, то увидите список подкоманд `banner`:

```
Switch1(config)#banner ?
  LINE                c banner-text c, where 'c' is a delimiting character
  config-save         Set message for saving configuration
  exec                Set EXEC process creation banner
  incoming            Set incoming terminal line banner
  login               Set login banner
  motd                Set Message of the Day banner
  prompt-timeout      Set Message for login authentication timeout
  slip-ppp            Set Message for SLIP/PPP
```

Это и есть та команда, которую вы ищете!

Вы все еще не уверены, что делать дальше, поэтому вводите команду `banner login ?`, чтобы получить немного больше информации:

```
Switch1(config)#banner login ?
  LINE c banner-text c, where 'c' is a delimiting character
```

Здесь вам нужно обратить пристальное внимание на отображаемую подсказку. Она указывает, что вам нужно ввести разделительный символ, который указывает на начало сообщения, за которым следует само сообщение, а затем снова разделительный символ. Введите символ # и нажмите клавишу **Enter**:

```
Switch1(config)#banner login #
Enter TEXT message. End with the character '#'.
Welcome to Switch1!#        ◄— Это сообщение для баннера авторизации, за которым следует символ #.
Switch1(config)#
```

Теперь, когда баннер авторизации настроен, выполните команду `exit`, чтобы выйти из режима глобальной конфигурации. Изменение, которое вы только что сделали, вступает в силу немедленно. Теперь выйдите из интерфейса коммутатора, снова введя команду `exit`. Если вы пользуетесь программой PuTTY, ваш сеанс должен завершиться.

Запустите программу PuTTY и вернитесь к интерфейсу коммутатора. На этот раз вы должны увидеть баннер авторизации в системе!

```
Welcome to Switch1!         ◄— Входной баннер
User Access Verification

Username:
```

Практикум

Перейдите в режим глобальной конфигурации и измените текст баннера авторизации. Завершите сеанс работы с коммутатором и снова подключитесь к нему. Отобразился ли баннер?

Хотя настройки сохраняются в рабочей конфигурации, она не является постоянной. Рабочая конфигурация хранится в оперативной памяти, содержимое которой очищается, когда коммутатор выключается или перезагружается. Чтобы внести настройки на постоянной основе, вам нужно сохранить рабочую конфигурацию в конфигурации запуска. *Это очень важный шаг, гарантирующий, что любые изменения конфигурации, которые вы внесли, не будут впоследствии неожиданно сброшены.*

3.7. Сохранение конфигурации запуска

При загрузке устройства Cisco система IOS считывает конфигурацию запуска, которая хранится в aptly-файле, называемом файлом конфигурации запуска. Этот файл хранится постоянно в энергонезависимой памяти (NVRAM), кото-

рая сохраняется даже тогда, когда коммутатор перезагружается или на него перестает подаваться питание. Можно представить NVRAM как жесткий диск. Данные в этой памяти остаются вне зависимости, включено или выключено устройство. Затем система IOS копирует содержимое файла конфигурации запуска в оперативную память, создавая рабочую конфигурацию.

Когда вы вносите изменения в рабочую конфигурацию, например меняете баннер авторизации, чаще всего вы хотите сохранить настройки на постоянной основе, записав рабочую конфигурацию в файл конфигурации запуска.

Верный способ сделать это – использовать команду `copy running-config startup-config`. Когда вы вводите эту команду и нажимаете клавишу **Enter**, система IOS уведомляет вас об имени файла, которое отображается в скобках, указывая, что вы можете нажать клавишу **Enter**, чтобы принять заданное имя файла:

```
Switch1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Switch1#
```

Примерно через секунду коммутатор сохранит файл конфигурации. Теперь, когда вы перезагрузите коммутатор, изменение, внесенное вами в баннер авторизации, сохранится.

Практикум

Сохраните рабочую конфигурацию на коммутаторе. Если у вас есть права доступа и операция не приведет к печальным последствиям, перезагрузите коммутатор, выполнив команду `reload`. Проверьте, отображается ли новый баннер авторизации.

Дополнительно

Мне нравится использовать более короткую команду для сохранения конфигурации запуска – `write memory` (которую вы можете ввести в сокращенном виде `wr me`). Она аналогична команде `copy run start`, за исключением того, что не запрашивает имя целевого файла. Имейте в виду, что сокращенная команда работает не на всех устройствах Cisco.

3.8. Команда no

Большинство конфигурационных команд можно отменить, выполнив команду `no`. Помещение слова `no` перед командой удаляет соответствующую настройку из рабочей конфигурации.

Вы можете применить команду `no`, чтобы удалить только что созданный баннер авторизации. Сначала проверьте рабочую конфигурацию для уточнения команды:

```
Switch1#show run | include banner
banner login ^C
```

Конфигурация выглядит несколько иначе, чем то, что вы набирали. Вы не набирали символы ^C в режиме глобальной конфигурации, но они есть. Не паникуйте, потому что система IOS иногда меняет или переупорядочивает введенные вами команды. Также обратите внимание, что сообщение Welcome to Switch1! не отображается. Этот текст сохранен в отдельной строке конфигурации, поэтому не отображается в выводе. Не обращайте на это внимания; вам просто нужно удалить команду banner login.

Для этого вернитесь в режим глобальной конфигурации и укажите слова no перед удаляемой командой следующим образом:

```
Switch1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch1(config)#no banner login ^C  ◄— Команда no удаляет данную настройку из рабочей конфигурации.
Switch1(config)#exit
```

Теперь вам нужно проверить, что система IOS фактически удалила команду конфигурации:

```
Switch1#show run | include banner
Switch1#
```

В выводе ничего не отображается, и теперь вы уверены, что система IOS удалила команду из рабочей конфигурации. Но команда все еще записана в конфигурации запуска:

```
Switch1#show startup-config | include banner
banner login ^C
Switch1#
```

Чтобы полностью удалить команду, вам нужно снова сохранить рабочую конфигурацию, перезаписав существующий файл конфигурации:

```
Switch1#write memory
Building configuration...
[OK]
Switch1#show startup-config | include banner
Switch1#  ◄— Команда banner login удалена из конфигурации запуска.
```

Система IOS перезаписала существующую конфигурацию запуска, а мы добрались до конца главы.

Практикум

Удалите баннер авторизации из конфигурации запуска вашего коммутатора. Сравните рабочую конфигурацию с конфигурацией запуска. Вы заметили какую-то разницу? Сохраните конфигурацию запуска, чтобы сохранить изменение на постоянной основе.

3.9. Команды, использованные в этой главе

Система IOS зависима от контекста, поэтому нельзя просто вводить какую-либо команду в любом месте и ожидать нужного результата.

Но существует несколько команд, которые вы будете использовать постоянно, поэтому важно понимать, для чего они предназначены. В табл. 3.1 перечислены некоторые распространенные команды и их описание.

Таблица 3.1. Команды, использованные в этой главе

Команда	Описание
`show ?`	Отображает подкоманды, выводящие на экран информацию об устройстве
`show version`	Отображает сведения об аппаратном и программном обеспечении (IOS), имя файла IOS
`show running-config`	Отображает рабочую конфигурацию устройства
`configure terminal`	Переводит в режим глобальной конфигурации
`no`	В режиме глобальной конфигурации удаляет указанную команду из рабочей конфигурации
`show startup-config`	Отображает конфигурацию загрузки, сохраненную в NVRAM
`copy running-config startup-config`	Перезаписывает конфигурацию загрузки рабочей конфигурацией
`reload`	Перезагружает устройство

3.10. Практическое задание

Экспериментируйте со всеми командами, о которых вы узнали в этой главе. Убедитесь, что вы можете авторизоваться на всех устройствах в своей сети и просмотреть рабочие конфигурации. Нет необходимости в дополнительных изменениях конфигурации. Важно то, что вы можете попасть в оболочку командной строки IOS, просмотреть рабочую конфигурацию и перейти в режим глобальной конфигурации.

Глава 4

Управление портами коммутатора

Сеть может состоять из множества различных устройств: компьютеров, IP-телефонов, принтеров, серверов, точек беспроводного доступа и самых разнообразных коммутирующих устройств. Единственное общее, что есть у всех этих устройств, – это то, что они физически подключаются к коммутатору, в частности к *Ethernet-порту* коммутатора.

Хотя в документации Cisco *иногда* ссылаются на порты как *интерфейсы*, я предпочитаю называть их портами, потому что так привычнее для большинства людей. Хотя подключение устройства к порту коммутатора является тривиальной задачей, процесс передачи данных между устройством и коммутатором отнюдь не тривиален, скорее это волшебство.

Знакома ли вам такая история или хотя бы ее часть: я работал в компании, офисы которой были разбросаны по всей стране. В каждом офисе было несколько коммутаторов, брандмауэр и маршрутизатор, но не было ИТ-специалистов для управления всем этим богатством. Каждый раз, когда сотрудник пересаживался за другой компьютер или новый сотрудник приходил в штат, человек, далекий от ИТ (обычно менеджер), подключал IP-телефон сотрудника к сетевому разъему в своем кабинете. Неудивительно, что телефон часто даже не включался. В других случаях он включался, но не работал. Или, может быть, телефон работал, но компьютер (который был подключен к телефону) не мог получить доступа к сети.

Даже в самых простых сетях новые устройства не определяются и не настраиваются автоматически. Когда устройство подключается к коммутатору и не работает должным образом, вам нужно разобраться, почему. Для этого надо войти в интерфейс коммутатора и выполнить поиск неполадки, начиная с порта. Возможно, порт отключен или неправильно настроен. Иногда проблема совсем не в конфигурации порта. Возможно, проблема с кабелями или компьютерами, и система IOS может помочь в определении проблем.

В этой главе вы узнаете о причинах наиболее распространенных сбоев и о том, как их устранять.

4.1. Просмотр состояния порта

Предположим, пользователь сообщает вам, что он просто сел за другой стол и теперь не может подключиться к сети. Вы идете к этому столу, проверяете сетевой разъем, через который подключен его компьютер, затем обращаетесь к аппаратному шкафу. Вы находите порт на коммутационной панели и отслеживаете подключенный к нему кабель до порта FastEthernet0/2 на коммутаторе. И тогда вы обнаруживаете, что индикатор соединения на коммутаторе не горит. Сетевой кабель подключен, но сетевого подключения нет. Почему? Вы отследили не тот кабель? Может быть, кабель как-то хитро проложен, может быть, он вообще в потолке зашит? А может быть, проблема с сетевой картой компьютера? Или проблемы с портом коммутатора?

Изучение каждой из этих проблем по отдельности займет много времени. А если проблема возникает в офисе на расстоянии в 1000 километров, обнаруживаются еще большие затруднения. В подобных случаях вам не обойтись без набора команд, предоставляющих информацию о состоянии каждого порта коммутатора. Команда `show interfaces status` предоставляет именно это. Интуитивно понятна не только сама команда, но и информация, которую она выводит. Давайте посмотрим, что я имею в виду.

Практикум

Авторизуйтесь на одном из ваших коммутаторов, войдите в режим enable и выполните команду `show interfaces status`.

Вы должны увидеть нечто подобное:

```
Switch1#show interfaces status

Port     Name   Status       Vlan     Duplex   Speed    Type
Fa0/2           disabled     1        auto     auto     10/100BaseTX  ◄── Порт отключен
Fa0/3           connected    1        a-full   a-100    10/100BaseTX
Fa0/4           connected    1        a-half   a-10     10/100BaseTX
Fa0/5           notconnect   1        auto     auto     10/100BaseTX
Fa0/6           disabled     1        auto     auto     10/100BaseTX
Fa0/7           connected    1        a-full   a-100    10/100BaseTX
```

Первое, что вы заметите, – это несколько колонок информации, но сейчас вас интересуют из них только колонки `Port` и `Status`.

Интересно отметить, что, хотя команда использует термин `interfaces`, в выводе употребляется слово `ports`. По мере того как вы будете набираться опыта в работе с системой IOS, вы начнете замечать, что сотрудники компании Cisco имеют тенденцию использовать разные термины для описания одного и того же.

Обратите внимание на то, что порты могут находиться в одном из трех состояний: `notconnect`, `disabled` и `connected`. Для понимания, что значит то или иное состояние, вам не понадобится сетевой администратор. Как я уже сказал, вы-

водимые командой сообщения понятны интуитивно. Обратите внимание, что порт FastEthernet0/2 отключен. Это объясняет, почему пользователь не может подключиться к сети! Но почему он отключен? Давайте разбираться.

Примечательно, что команда `show interfaces status` предоставляет в выводе краткую информацию. Для получения более подробной информации о конкретном интерфейсе, в нашем случае это FastEthernet0/2, следует воспользоваться командой `show interface FastEthernet0/2`. Из 2-й главы вы уже знаете, что команды в системе IOS можно сокращать. Точно так же можно сокращать и имена портов.

Практикум

Если команда `show interfaces status` сообщила, что некий порт отключен, выполните команду с синтаксисом `show interface` *порт*. В нашем случае выполните команду `show interface fa0/2`.

Вы должны увидеть следующий вывод (или нечто подобное):

```
Switch1#show interface fa0/2
FastEthernet0/2 is administratively down, line protocol is down (disabled)
 Hardware is Fast Ethernet, address is 0023.ab40.8e04 (bia 0023.ab40.8e04)
 MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
 reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation ARPA, loopback not set
 Keepalive set (10 sec)
 Auto-duplex, Auto-speed, media type is 10/100BaseTX
 input flow-control is off, output flow-control is unsupported
 ARP type: ARPA, ARP Timeout 04:00:00
 Last input never, output never, output hang never
 Last clearing of "show interface" counters never
 Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
 Queueing strategy: fifo
 Output queue: 0/40 (size/max)
 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 multicasts)
 0 runts, 0 giants, 0 throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
 0 watchdog, 0 multicast, 0 pause input
 0 input packets with dribble condition detected
 0 packets output, 0 bytes, 0 underruns
 0 output errors, 0 collisions, 1 interface resets
 0 unknown protocol drops
 0 babbles, 0 late collision, 0 deferred
 0 lost carrier, 0 no carrier, 0 pause output
 0 output buffer failures, 0 output buffers swapped out
```

Однозначно указывает на то, что кто-то принудительно отключил этот порт

Не стоит пытаться осмыслить всю выводимую информацию. То, что нужно знать, лежит прямо на поверхности. В нашем случае это самая первая строка. В ней говорится, что порт `administratively down` кто-то отключил.

Если порт отключен, это равносильно тому, что к нему ничего подключено. Коммутатор его полностью игнорирует. Отключенный порт не будет подавать даже питание на IP-телефон, если его подключить к этому порту. Он «мертв», и вам нужно его «оживить».

4.2. Включение портов

Для описания того, что порт находится в отключенном состоянии, в документации и командах Cisco используются три разных термина: *disabled*, *administratively down* и *shutdown*. Все эти термины означают одно и то же.

Теперь предположим, что ваш менеджер получает жалобы, что люди не могут подключиться к сети. Вас попросят проверить все порты коммутатора и убедиться, что ни один из них не отключен. Вам нужно использовать команду `show interfaces status`, как и ранее, но, чтобы вывод был информативнее, вы можете отфильтровать его, чтобы включить в сообщение информацию только об отключенных портах.

Практикум

Выполните команду `show interfaces status | include disabled`, чтобы отобразить только отключенные порты.

Все, что вы должны увидеть, – это отключенные порты, если таковые есть:

```
Fa0/2  disabled 1 auto auto 10/100BaseTX
Fa0/6  disabled 1 auto auto 10/100BaseTX
Fa0/13 disabled 1 auto auto 10/100BaseTX
Fa0/14 disabled 1 auto auto 10/100BaseTX
Fa0/18 disabled 1 auto auto 10/100BaseTX
Fa0/23 disabled 1 auto auto 10/100BaseTX
```

Обратите внимание, что заголовки колонок не отображаются. Самое время попрактиковаться с форматированием вывода команды, чтобы отобразить заголовки.

Практикум

Выполните команду `show interfaces status | i disabled|Status` для отображения только отключенных портов и заголовков колонок.

Опять же, если у вас нет каких-либо отключенных портов на вашем коммутаторе, вы увидите только заголовки.

```
Switch1#show interfaces status | i disabled|Status
Port      Name  Status      Vlan    Duplex    Speed   Type  ◄── Заголовки отображены
Fa0/2           disabled    1       auto       auto   10/100BaseTX
Fa0/6           disabled    1       auto       auto   10/100BaseTX
Fa0/13          disabled    1       auto       auto   10/100BaseTX
```

```
Fa0/14          disabled    1       auto        auto    10/100BaseTX
Fa0/18          disabled    1       auto        auto    10/100BaseTX
Fa0/23          disabled    1       auto        auto    10/100BaseTX
```

Напомню (см. главу 3), что рабочая конфигурация системы IOS управляет тем, как работает коммутатор *прямо сейчас*. Это означает, что порты отключены командой в рабочей конфигурации системы IOS. Давайте вернемся к порту FastEthernet0/2 и разберемся, какая команда отключает его.

Практикум

Выполните команду `show run interface FastEthernet0/2`, чтобы увидеть конфигурацию интерфейса.

В вашем случае вывод может быть немного иным, но вы должны увидеть раздел рабочей конфигурации, отвечающий за управление портом FastEthernet0/2:

```
Switch1#show run interface FastEthernet0/2
Building configuration...

Current configuration : 43 bytes
!
interface FastEthernet0/2
 Shutdown  ◄── Порт не доступен
End  ◄── Указывает на конец раздела конфигурации порта
```

Этот небольшой фрагмент рабочей конфигурации называется *разделом конфигурирования интерфейса (порта)*. `shutdown` – *команда интерфейса*, которая отключает порт. Обратите внимание, что имя команды имеет отступ, который указывает на то, что действие команды распространяется только на данный раздел конфигурации порта. Ключевое слово `end` обозначает конец раздела.

Включение порта FastEthernet0/2 удалит команду `shutdown`. Как вы узнали из предыдущей главы, добавление слова `no`, как правило, достаточно, чтобы удалить или, по крайней мере, отменить команду. Давайте попробуем.

Практикум

Сначала перейдите в режим глобальной конфигурации, выполнив команду `configure terminal`. Далее, поскольку вам необходимо внести изменения в один конкретный интерфейс, FastEthernet0/2, выполните команду `interface fa0/2`.

Вы должны увидеть следующее:

```
Switch1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch1(config)#interface fa0/2
Switch1(config-if)#
```

Обратите внимание, что приглашение изменилось и вместо слова `config` отображается текст `config-if`, указывая, что вы находитесь в режиме конфигу-

рирования интерфейса (if – сокращение от слова interface). Теперь вы можете отменить команду shutdown и включить интерфейс.

Практикум

Пока вы находитесь в режиме конфигурирования интерфейса, введите команду no shutdown и нажмите клавишу **Enter**.

В выводе команды вы должны увидеть два сообщения, указывающие, что порт включен:

```
Switch1(config-if)#no shutdown
Switch1(config-if)#
*Mar  1 06:49:27.824: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to up
*Mar  1 06:49:28.831: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed
     state to up
```

Всегда нужно проверять внесенные изменения, поэтому выполните команду show interfaces status | i Fa0/2 |Status:

```
Port      Name  Status      Vlan    Duplex    Speed   Type
Fa0/2           connected   1       a-full    a-100   10/100BaseTX
```

Порт FastEthernet0/2 включен. Замечательно, но ваш менеджер хочет, чтобы были включены *все* порты. Если отключено несколько портов, их включение – трудоемкая задача. Вам нужен способ включить все порты сразу.

4.2.1. Команда interface range

Ранее вы попадали в режим конфигурирования интерфейса, выполняя команду interface fa0/2 в режиме глобальной конфигурации. Любые изменения, внесенные в режиме конфигурирования интерфейса, влияют только на порт, указанный в команде. Другими словами, вы выбрали только один порт. Но есть способ выбрать и настроить сразу несколько портов.

Команда interface range позволяет выбрать диапазон интерфейсов, указав их через разделитель – тире. Поскольку вам необходимо включить все порты на вашем коммутаторе, вы можете указать весь диапазон портов на коммутаторе, а затем сразу же выполнить команду no shutdown.

Практикум

Перейдите в режим глобальной конфигурации и введите следующие команды для включения портов Fast-Ethernet 0/1 – 0/24:

```
Interface range fa0/1-24
no shutdown
```

Если ваш коммутатор имеет 48 портов, вы можете указать 48 вместо 24. Независимо от того, сколько портов находится на вашем коммутаторе, вам нужно указать начальный и конечный порты.

Вы должны увидеть следующий результат:

```
Switch1(config)#interface range fa0/1-24   ◄— Выбраны интерфейсы FastEthernet0/1 – 0/24
Switch1(config-if-range)#no shutdown       ◄— Указан режим конфигурирования диапазона интерфейсов
Switch1(config-if-range)#
*Mar  1 14:40:53.438: %LINK-3-UPDOWN: Interface FastEthernet0/6, changed
     state to down
*Mar  1 14:40:53.455: %LINK-3-UPDOWN: Interface FastEthernet0/13, changed
     state to down
*Mar  1 14:40:53.463: %LINK-3-UPDOWN: Interface FastEthernet0/14, changed
     state to down
*Mar  1 14:40:53.480: %LINK-3-UPDOWN: Interface FastEthernet0/18, changed
     state to down
*Mar  1 14:40:53.496: %LINK-3-UPDOWN: Interface FastEthernet0/23, changed
     state to down
```

В выводе указано, что состояние некоторых интерфейсов изменено на `down` (`changed state to down`). Это может показаться немного странным, учитывая, что вы только что включили интерфейсы. Но не волнуйтесь, эта строка означает лишь то, что на этих портах нет активных устройств. Проверка с выводом состояния интерфейса с помощью команды `show interfaces status | i Fa0/6|Status` подтверждает мои слова:

```
Port      Name  Status      Vlan    Duplex    Speed   Type
Fa0/6           notconnect  1       auto      auto    10/100BaseTX
```

И действительно, интерфейс FastEthernet0/6 находится в состоянии `notconnect`, что означает, что он включен, но либо нет подключенного к нему устройства, либо подключенное устройство выключено. Важно то, что при включении всех портов коммутатора вы можете подключить устройство к любому порту и выполнить соединение с сетью.

4.3. Отключение портов

Вы только что узнали, как включить порты, теперь следует разобраться, как их отключать. Но для начала вам, наверное, интересно узнать, зачем вообще может понадобиться отключать порты. Почему бы не оставить их включенными? Это, конечно, удобно: подключил Ethernet-кабель к коммутатору – и все работает. Но иногда возникает требование политики безопасности – неиспользуемые порты должны быть отключены. Хотя такая политика и может показаться излишней бюрократией, для нее существуют веские причины.

Предположим, что каждые выходные какой-нибудь менеджер приходит в офис, садится за пустой стол и подключает свой персональный ноутбук к сети. Он не знает, что его ноутбук заражен нехорошим вирусом. Однажды утром в понедельник вы узнаете, что все компьютеры в сети заражены. Оказалось, что ноутбук этого менеджера заразил все остальные компьютеры в выходные дни, когда он подключил его к сети. А вот если бы порт коммутатора,

который он использовал, был отключен, его ноутбук не смог бы подключиться к сети. Это заставило бы владельца заручиться поддержкой ИТ-специалиста, который мог бы найти вирус до заражения других компьютеров.

Существует много других причин отключать неиспользуемые порты коммутатора, даже очень много. Просто воспинимайте этот прием как дешевый и простой способ избежать множества неприятностей.

4.3.1. Поиск неиспользуемых интерфейсов

Отключение портов только потому, что вы *считаете*, что они не используются, – плохая, даже очень плохая идея. Вы можете внести реальный хаос в работу сети, сделав это. Очень важно сначала убедиться, что порты, которые вы собираетесь отключить, фактически не используются. Вы можете сделать это, выполнив команду `notconnect`.

Практикум

Отобразите все интерфейсы в состоянии `notconnect`:

```
show interfaces status | i notconnect
```

Если у какого-либо порта нет подключенных устройств или они есть, но порт не включен, он будет отображаться в выводе следующим образом:

```
Switch1#show interfaces status | i notconnect
Fa0/5          notconnect   1     auto   auto 10/100BaseTX
Fa0/6          notconnect   1     auto   auto 10/100BaseTX
Fa0/8          notconnect   1     auto   auto 10/100BaseTX
Fa0/9          notconnect   1     auto   auto 10/100BaseTX
Fa0/10         notconnect   1     auto   auto 10/100BaseTX
Fa0/11         notconnect   1     auto   auto 10/100BaseTX
Fa0/13         notconnect   1     auto   auto 10/100BaseTX
Fa0/14         notconnect   1     auto   auto 10/100BaseTX
Fa0/15         notconnect   1     auto   auto 10/100BaseTX
Fa0/16         notconnect   1     auto   auto 10/100BaseTX
Fa0/17         notconnect   1     auto   auto 10/100BaseTX
Fa0/18         notconnect   1     auto   auto 10/100BaseTX
Fa0/19         notconnect   1     auto   auto 10/100BaseTX
Fa0/20         notconnect   1     auto   auto 10/100BaseTX
Fa0/21         notconnect   1     auto   auto 10/100BaseTX
Fa0/22         notconnect   1     auto   auto 10/100BaseTX
Fa0/23         notconnect   1     auto   auto 10/100BaseTX
Fa0/24         notconnect   1     auto   auto 10/100BaseTX
```

Суть в том, чтобы увидеть все порты, находящиеся в отключенном состоянии. Как вы можете видеть, их много. Поскольку порты FastEthernet0/13 – 0/24 пронумерованы последовательно, вы можете отключить их все сразу, перейдя в режим конфигурирования диапазона интерфейсов. Можно воспользоваться следующими командами:

```
Configure terminal
interface range fa0/13-24
Shutdown
```

Но даже после этого все еще остается несколько интерфейсов, которые необходимо отключить: FastEthernet0/5, 0/6 и 0/8 – 0/11. К счастью, вы также можете отключить их, используя команду `interface range`. Но на этот раз, помимо указания диапазона, вы будете отдельно указывать интерфейсы, разделяя их запятой:

```
Interface range fa0/5,fa0/6,fa0/8-11
shutdown
```

Теперь снова выполните команду `show interfaces status | i notconnect`:

```
Switch1#show interfaces status | i notconnect|Status
Port      Name          Status       Vlan       Duplex   Speed Type
```

На этот раз список пуст, а это значит, что все неиспользуемые интерфейсы теперь отключены.

Практикум

Определите все неиспользуемые интерфейсы на вашем коммутаторе:

```
show interfaces status | i notconnect
```

Если есть неиспользуемые, отключите их, используя команду `shutdown`.

Внимание! Если интерфейс находится в состоянии `not connect`, это не означает, что никто им не пользуется. Если пользователь ушел домой и выключил компьютер, порт, к которому подключен компьютер, будет переведен в состояние `not connect`. Если вы отключите этот порт, то на следующий рабочий день компьютер не сможет подключиться к сети, и вам придется его включать. Прежде чем отключать неиспользуемый порт, всегда рекомендуется проверить, что он не понадобится в ближайшем будущем.

4.4. Изменение скорости порта и дуплекса

В начале главы я сказал, что сети не являются автонастраиваемыми. И тот факт, что состояние порта определяется как `connected`, вовсе не означает, что все работает правильно. Порт коммутатора и устройство должны работать с одинаковой скоростью и дуплексом, чтобы данные передавались без искажений. Скорость порта и дуплекс – это концепции, с которыми вы уже знакомы, но они имеют особое значение для конфигурации портов коммутатора. Давайте кратко рассмотрим их.

4.4.1. Скорость

Скорость порта – это термин, определяющий *пропускную способность* – скорость, с которой данные могут передаваться между устройством и коммута-

тором. Хотя скорость 100 Мегабит в секунду (Мбит/с) встречается наиболее часто, иногда можно увидеть скорость и поменьше – 10 Мбит/с. Не существует единственной правильной скорости, но большинству устройств необходима скорость *не менее* 100 Мбит/с. Некоторые, более старые устройства могут быть неспособны к этому и работают на скорости 10 Мбит/с. Дело в том, что если вы видите устройство, работающее со скоростью 10 Мбит/с, требуется дальнейший анализ, но это не обязательно указывает на проблему.

Команда `show interfaces status | i connected|Status` дает вам краткую информацию о скорости работы каждого порта:

```
Switch1#show interfaces status | i connected|Status
Port      Name  Status      Vlan    Duplex    Speed   Type
Fa0/1           connected   1       a-full    a-100   10/100BaseTX   <-- Работает со скоростью
Fa0/2           connected   1       a-full    a-100   10/100BaseTX       100 Мбит/с, полный дуплекс
Fa0/3           connected   1       a-full    a-100   10/100BaseTX
Fa0/4           connected   1       a-half    a-10    10/100BaseTX   <-- Работает со скоростью
Fa0/7           connected   1       a-full    a-100   10/100BaseTX       10 Мбит/с, полудуплекс
Fa0/12          connected   1       a-full    a-100   10/100BaseTX
```

Соответствующая колонка со значениями скорости имеет заголовок **Speed**. Взгляните на строку для порта FastEthernet0/1. Значение `a-100` указывает, что порт работает со скоростью 100 Мбит/с, а коммутатор и устройство автоматически согласовали эту скорость. Такое поведение называется *автосогласованием*, и по умолчанию оно включено для всех портов. Коммутатор и устройство пытаются согласовать максимальную скорость, которую они поддерживают, в нашем случае – 100 Мбит/с.

4.4.2. Дуплекс

Дуплекс означает одновременную двустороннюю связь между портом коммутатора и устройством. Терминология дуплекса немного неудобна. Полный дуплекс означает, что и устройство, и коммутатор могут передавать данные одновременно, не конфликтуя друг с другом. *Полудуплекс* означает, что только одно устройство может передавать и принимать данные одновременно. Например, если с порта коммутатора отправляются данные, компьютер с другого конца должен дождаться конца передачи, прежде чем он сможет начать свою передачу.

Взгляните на колонку **Duplex**. Все порты поддерживают режим `a-full`, за исключением интерфейса FastEthernet0/4, который продемонстрировал результат `a-half`. Как и в случае со скоростью, буква `a` указывает, что дуплекс автосогласован. Большинство современных устройств применяет полный дуплекс по автосогласованию.

4.4.3. Автосогласование

В конце 1990-х и начале 2000-х годов обычной практикой было отключать автосогласование для скорости и дуплекса на всех портах коммутаторов, поскольку

такое поведение было ненадежным. Эти дни давно прошли, и вы не должны отключать автосогласование, если у вас нет конкретных причин для этого.

Отключение автосогласования – это не то, что вы делаете явно. Это побочный эффект *принудительного* задания скорости или дуплекса порта. Например, если вы принудительно указываете для порта скорость 10 Мбит/с, полудуплекс, вы сообщаете коммутатору, чтобы этот порт мог работать *только* с такой скоростью и дуплексом *все время*. Принудительное задание противоположно автосогласованию.

Хотя вы и не должны обязательно указать скорость и дуплекс для большинства портов, иногда это полезно. Некоторые устройства, такие как старые серверы печати HP JetDirect и адаптеры для аналоговых телефонов Cisco, хотят работать со скоростью 10 Мбит/с в полудуплексе. Хотя даже более старые устройства *поддерживают* автосогласование, многие из них относятся к той эпохе, когда автосогласование считалось ненадежным.

Но и сегодня некоторые устройства, такие, например, как IP-камеры и системы безопасности, сами часто жестко привязаны к определенной скорости и дуплексу, и ваша конфигурация порта коммутатора должна соответствовать этим характеристикам. Иногда производители устройства требуют принудительного задания скорости и дуплекса без уважительной причины, но эти требования необходимо соблюдать, иначе вы не сможете получить техподдержку.

Суть в том, что хотя изменение скорости порта и дуплекса – это не то, что хочется делать каждый день, но вполне вероятно, что у вас есть хотя бы одно устройство в сети, которое требует этого.

4.4.4. Изменение скорости порта

Давайте еще раз взглянем на состояние подключенных интерфейсов:

```
Switch1#show interfaces status | i connected|Status
Port      Name  Status      Vlan    Duplex    Speed   Type
Fa0/1           connected   1       a-full    a-100   10/100BaseTX
Fa0/2           connected   1       a-full    a-100   10/100BaseTX
Fa0/3           connected   1       a-full    a-100   10/100BaseTX
Fa0/4           connected   1       a-half    a-10    10/100BaseTX
Fa0/7           connected   1       a-full    a-100   10/100BaseTX
Fa0/12          connected   1       a-full    a-100   10/100BaseTX
```

Мой компьютер подключен к порту FastEthernet0/1, и он согласовал скорость и дуплекс со скоростью 100 Мбит/с, полный дуплекс с коммутатором. Посмотрим, что произойдет, когда я поменяю скорость и дуплекс на 10 Мбит/с:

```
Switch1(config)#interface fa0/1
Switch1(config-if)#speed 10
Switch1(config-if)#
*Mar  2 03:33:25.235: %LINEPROTO-5-UPDOWN: Line protocol on Interface
     FastEthernet0/1, changed state to down
*Mar  2 03:33:27.248: %LINEPROTO-5-UPDOWN: Line protocol on Interface
     FastEthernet0/1, changed state to up
```

Обратите внимание, что порт на мгновение переходит в состояние down, а затем возвращается. Люди, работающие с сетью, называют это *хлопком* или *скачком*. Тот факт, что порт вернулся, указывает на то, что мой компьютер действительно поддерживает скорость 10 Мбит/с. Если бы мой компьютер не поддерживал такую скорость, интерфейс не вернул бы свое состояние. Давайте снова проверим этот порт:

```
Switch1#show interfaces status | i Fa0/1 |Status
Port     Name  Status      Vlan    Duplex    Speed   Type
Fa0/1          connected   1       a-full    10      10/100BaseTX
```

На этот раз указана скорость 10 Мбит/с. Но не только это, буква a также отсутствует, что указывает на то, что скорость не была согласована автоматически.

Практикум

Найдите порт в состоянии connected, который, как вы знаете, не используется, и попытайтесь изменить скорость на 100 или 10 Мбит/с. Используйте встроенную справочную систему IOS для поддержки. Если вы не знаете, какую команду ввести, введите символ ? – и система IOS продемонстрирует доступные команды.
Когда вы закончите, включите автосогласование, используя команду конфигурирования интерфейса speed auto.

Кстати, сетевые пользователи используют сокращенное обозначение для скорости порта и дуплекса. Например, скорость 100 Мбит/с и полный дуплекс можно записывать просто как 100/full. Я буду записывать скорость и дуплекс именно таким образом всю оставшуюся часть книги.

4.4.5. Изменение дуплексного режима

Ранее мы видели, что порт FastEthernet0/4 работает в полудуплексном режиме. Давайте подробнее рассмотрим этот порт:

```
Switch1#show interfaces status | i Fa0/4|Status
Port     Name  Status      Vlan    Duplex    Speed   Type
Fa0/4          connected   1       a-half    a-10    10/100BaseTX
```

Он работает в режиме 10/half, и оба этих параметра автосогласованы. Если вы когда-либо увидите такие значения в сети, вы определенно захотите исследовать проблему. В худшем случае это может указывать на проблемы с кабелем. В лучшем случае это может привести вас к интересному устройству. Скорее всего, все, что работает в режиме 10/half, не является ни компьютером, ни сервером.

Обратите внимание, что скорость и дуплекс были автосогласованы. Давайте попробуем изменить дуплекс на полный:

```
Enter configuration commands, one per line. End with CNTL/Z.
Switch1(config)#interface fa0/4
Switch1(config-if)#duplex full
```

```
Switch1(config-if)#
*Mar  2 04:39:12.755: %LINEPROTO-5-UPDOWN: Line protocol on Interface
     FastEthernet0/4, changed state to down
*Mar  2 04:39:13.761: %LINK-3-UPDOWN: Interface FastEthernet0/4, changed
     state to down
```

Состояние порта меняется на down и не возвращается. Все, что находится на другом конце, безусловно, не поддерживает полнодуплексную связь. Также возможно, хотя и менее вероятно, что поврежден Ethernet-кабель. Именно поэтому всегда хочется исследовать любой порт с явным указанием скорости и дуплекса, это необычно.

Давайте снова взглянем на этот порт:

```
Switch1#show interfaces status | i Fa0/4|Status
Port      Name  Status      Vlan    Duplex    Speed   Type
Fa0/4           notconnect  1       full      auto    10/100BaseTX
```

Интересно! Порт теперь находится в состоянии notconnect. Я ничего не отключал, но из-за несогласованного дуплекса порт выглядит так, как будто к нему вообще ничего не подключено. Пример иллюстрирует, почему лучше позволить автосогласованию выполнять свою работу, если для ручного управления убедительной причины нет.

Практикум

Найдите неиспользуемый порт в состоянии connected и попробуйте изменить дуплекс на полудуплекс или, наоборот, на полный. Что происходит? Когда вы закончите, верните автосогласование, выполнив команду duplex auto.

4.5. Команды, использованные в этой главе

В этой главе вы узнали, как управлять отдельными портами в режиме конфигурирования интерфейса. При работе в оболочке командной строки IOS вам необходимо различать команды режима глобальной конфигурации и команды режима конфигурирования интерфейса. В табл. 4.1 приведены различия этих команд, а также перечислены некоторые команды show, используемые в этой главе.

Таблица 4.1. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
show interfaces status	–	Отображает состояние, скорость и дуплекс всех портов
show run interface fa0/2	–	Отображает подробную информацию о порте FastEthernet0/2
interface range fa0/5,fa0/ 6,fa0/8-11	Глобальный	Выбирает диапазон портов для конфигурации

Окончание табл. 4.1

Команда	Режим конфигурирования	Описание
`speed 10/100/auto`	Интерфейс	Принудительно задает скорость порта или позволяет автосогласование скорости
`duplex full/half/auto`	Интерфейс	Принудительно задает дуплекс порта или позволяет автосогласование
`(no) shutdown`	Интерфейс	Отключает или включает порт

4.6. Практическое задание

Практикуйте все, что вы узнали в этой главе, если вы еще этого не сделали. Обратитесь к тестовой сети, если она у вас есть:

1. Выведите только те порты, которые не подключены.
2. Выберите диапазон неиспользуемых портов и отключите их.
3. Попробуйте выбрать диапазон портов, больший, чем количество портов вашего коммутатора. Какая ошибка возникает?
4. Используя команды `show`, посмотрите, можете ли вы найти какие-нибудь интересные устройства, работающие со скоростью 10 Мбит/с или в полудуплексе.
5. Выберите порт, работающий со скоростью `100/full`. Измените скорость на 10 Мбит/с. Что произошло?
6. Что произойдет, если вы установите его скорость равной 100 Мбит/с?
7. Измените в настройках порта дуплекс на полудуплекс. Что произошло?
8. Измените полудуплексный режим на полный. Если он не возвращается, попробуйте отключить порт и затем включить его.
9. Когда вы закончите, не забудьте сохранить свою рабочую конфигурацию, используя команду `copy run start` или `write memory`.

Глава 5

Защита портов с помощью технологии Port Security

В предыдущей главе вы узнали, как защитить сеть, отключив неиспользуемые порты. Отключение неиспользуемых портов может помешать подключению устройства с вредоносным кодом к неиспользуемому порту или получению несанкционированного доступа к сети. Это также может помочь научить пользователей, особенно из удаленных офисов, звонить в ИТ-отдел, *прежде* чем пересаживаться за другие столы. После нескольких безуспешных попыток подключения компьютера к отключенным портам большинство из них поймет, что сначала нужно позвонить в ИТ-отдел.

Но хотя отключение – и наиболее безопасный вариант в случае с неиспользуемыми портами, оно ничего не дает в плане защиты рабочих портов. А в рабочей сети большинство портов коммутатора *будет* задействовано.

Port Security – это универсальная функция для предотвращения сетевых атак и несанкционированного доступа. Эта функция снижает возможности изменения сетевой конфигурации, ограничивая количество устройств с уникальными MAC-адресами, которые могут использовать данный порт. Напомню, что каждое устройство в сети имеет уникальный MAC-адрес, который используется для связи с другими устройствами в пределах широковещательного домена. Безопасность – это общее требование, а не разовый запрос. Некоторые организации предпочитают минимальный уровень безопасности, тогда как другие выдвигают требования, граничащие с паранойей. Вместо того чтобы рассказывать вам, насколько безопасной должна быть сеть, в этой главе я расскажу о конкретных рисках, которые можно снизить с помощью настройки функции Port Security, а вы уже сами решайте, насколько вам это необходимо. Затем я покажу вам, как настроить функцию Port Security для удовлетворения ваших требований.

Я не буду показывать вам все возможные способы настройки функции Port Security. Вместо этого я расскажу вам, как обеспечить минимальный и максимальный уровни безопасности, как показано в табл. 5.1.

Таблица 5.1. Уровни функции Port Security

Уровень защиты	Предотвращаемые атаки
Минимальный	Атака по MAC-адресу, отказ в обслуживании (DoS), проверка трафика
Максимальный	Все вышеперечисленное, а также предотвращение несанкционированного доступа к устройствам и препятствование распространению вредоносного программного обеспечения

В табл. 5.1 перечислены типы атак, которые могут быть предотвращены на соответствующем уровне функции Port Security. Начнем с минимального уровня.

5.1. Конфигурация минимального уровня функции Port Security

Хотя я и не могу предсказать, насколько безопасной вы хотите видеть вашу сеть, я могу сказать, что вы определенно захотите обеспечить хотя бы минимальную безопасность на всех портах конечного пользователя.

Безопасность всегда является компромиссом. Вам нужно учитывать, стоят ли усилия по защите от конкретных рисков затраченных на эти усилия времени и денег. Меры по защите портов уже предусмотрены в системе IOS, поэтому никаких дополнительных материальных затрат не требуется, а время и усилия, необходимые для настройки функции Port Security на минимальный уровень, незначительны. Но взамен вы получаете спокойствие и защиту. Устранение последствий атаки, называемой атакой по MAC-адресу, может оказаться трудоемким и дорогостоящим.

5.1.1. Предотвращение атаки по MAC-адресу

Напомню (см. главу 2), что коммутатор поддерживает таблицу MAC-адресов, содержащую MAC-адрес каждого устройства и имя порта, к которому он подключен. Таблица 5.2 содержит пример информации, которую вы найдете в таблице MAC-адресов. Контролируя месторасположение всех устройств, коммутатор избегает рассылки всех кадров каждому устройству.

Таблица 5.2. Пример таблицы MAC-адресов

Устройство	MAC-адрес	Порт коммутатора
Компьютер Бена	`0800.2700.ec26`	FastEthernet0/1

Во время атаки по MAC-адресу программа злоумышленников постоянно отправляет кадры с поддельными MAC-адресами в качестве исходного адреса. Поскольку каждый кадр словно поступает с разных MAC-адресов, коммутатор заносит эти адреса в таблицу и рассылает эти кадры на все доступные порты. Результат заключается в том, что компьютер, запускающий вредоносную программу, становится анализатором трафика (сниффером), который способен перехватывать каждый кадр в сети.

Рисунок 5.1 иллюстрирует, как злоумышленник может использовать MAC-адресную атаку для перехвата трафика.

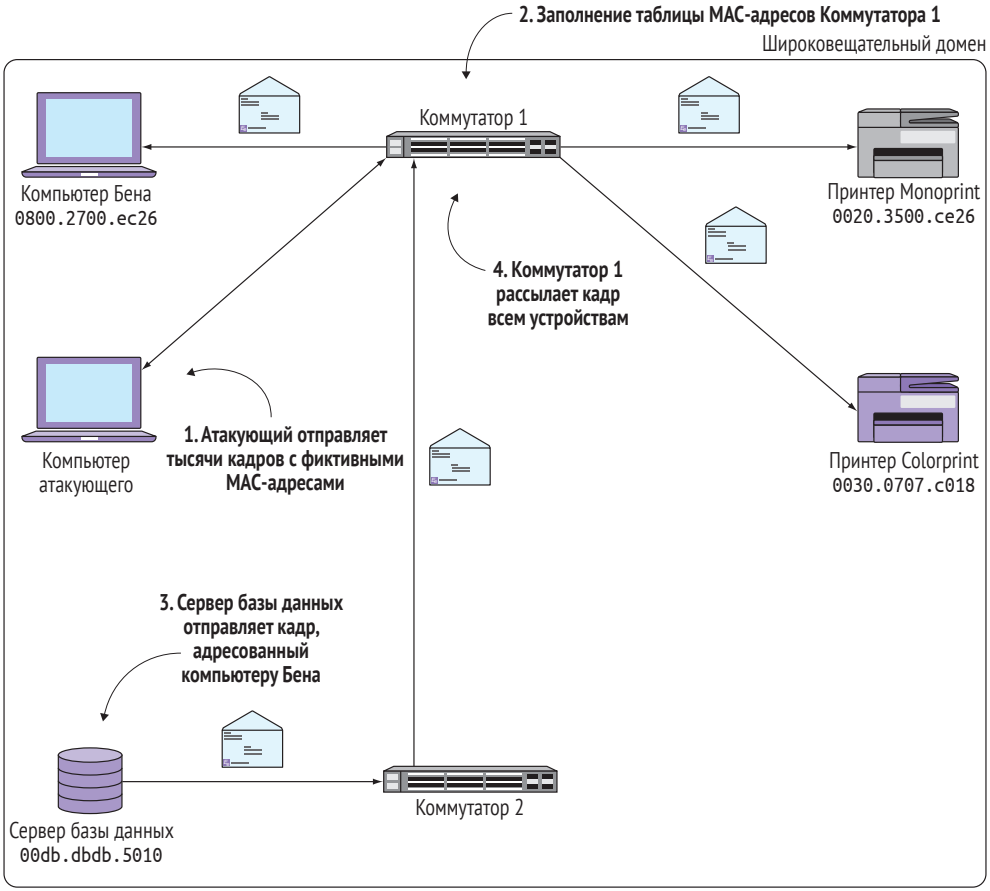

Рис. 5.1 ❖ Атака по MAC-адресу

На шаге 1 злоумышленник отправляет тысячи кадров с MAC-адресами фиктивного источника на Коммутатор 1. На шаге 2 заполняется таблица MAC-адресов. На шаге 3 сервер базы данных отправляет кадр, адресованный моему компьютеру. Коммутатор 2 передает этот кадр Коммутатору 1. Наконец, на шаге 4 Коммутатор 1 рассылает кадр по всем портам, включая тот, который подключен к компьютеру злоумышленника.

Но это еще не самое худшее. Атака по MAC-адресу может фактически привести к отказу в обслуживании всех пользователей. Помните высказывание из главы 2: «Когда все говорят, никто не слушает». MAC-адресная атака значительно снижает производительность сети до такой степени, что сеть практически

парализует. Представьте себе, что десятки клиентских вызовов не могут быть переданы в сеть, потому что не могут дождаться своей очереди из-за перегруженности IP-трафика. Функция Port Security может помочь в защите от подобных сценариев. Только один лишь незащищенный порт – это все, что требуется для атаки по MAC-адресу, которая уничтожит вашу сеть. Поэтому очень важно настроить функцию Port Security на каждом порту.

Вы можете защититься от MAC-адресной атаки с помощью антивирусного программного обеспечения на клиентских компьютерах и серверах, убедившись, что конечные пользователи не имеют административного доступа на своих компьютерах. Но этот метод надежен не на 100%. Функция Port Security – это самый надежный способ предотвратить MAC-атаку, даже если другие меры безопасности не срабатывают.

Обычно коммутатор не учитывает, сколько разных MAC-адресов приписано одному и тому же порту. Так обеспечивается соединение независимо от MAC-адреса источника. Напомню, что MAC-адреса были разработаны для того, чтобы можно было подключить устройство к сети и обеспечить его работу. Но именно из-за этого MAC-атака становится возможной.

Очевидным решением является ограничение количества MAC-адресов источников, которые могут быть одновременно связаны с данным портом. Именно так работает функция Port Security. Вы настраиваете эту функцию, чтобы разрешить определенное количество MAC-адресов, и она обеспечивает доступ по принципу «первым пришел – первым обслужен». Давайте рассмотрим пример.

Предположим, что у вас есть пользователь с двумя устройствами – компьютером и IP-телефоном Cisco, подключенными к одному порту. Телефон физически подключен к коммутатору, а компьютер физически подключен к телефону, и данные передаются через телефон. Таблица 5.3 показывает, как это будет выглядеть в таблице MAC-адресов.

Таблица 5.3. Таблица MAC-адресов

Устройство	MAC-адрес	Порт
Компьютер	`0123.4567.8901`	FastEthernet0/23
IP телефон	`0123.4598.7654`	FastEthernet0/23

Эти два устройства имеют два уникальных MAC-адреса, поэтому вы можете ограничить максимальное количество MAC-адресов до двух, выполнив команду конфигурирования интерфейса `switchport port-security maximum 2`.

Практикум

Найдите порт с двумя подключенными к нему устройствами. Если у вас есть компьютер, подключенный к IP-телефону, – это идеально. Если такого порта нет, вы все равно можете выполнить упражнение; просто измените команду, чтобы разрешить только один MAC-адрес.

Выполните следующие команды, чтобы настроить максимальное количество разрешенных MAC-адресов порта, равное двум:

```
interface fa0/1
switchport mode access
switchport port-security maximum 2
```

На данный момент ничего не должно произойти. Это связано с тем, что эта команда фактически не задействует функцию Port Security. На первый взгляд это кажется противоречием, но на самом деле это благословение. Функция Port Security имеет возможность указывать на порт, непригодный для использования, если тот неправильно настроен. Очень важно, чтобы вы узнали, сколько MAC-адресов должно поддерживаться на каждом порту, прежде чем включить функцию Port Security.

Если вы не можете точно определить количество используемых MAC-адресов, можно указать значение с запасом, например `10`, а точнее отрегулировать позже. И теперь, если у вашего шефа есть секретный коммутатор под его столом с восемью различными MAC-адресами, вы узнаете это, но уже не от него, а от системы IOS.

Практикум

После того как вы настроили максимальное количество MAC-адресов, включите функцию Port Security, выполнив команду конфигурирования интерфейса `switchport port-security`.

Теперь проверьте свою конфигурацию, выполнив команду `show port-security`.

Вы должны увидеть нечто следующее:

```
Switch1#show port-security
Secure Port   MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                  (Count)       (Count)        (Count)
------------------------------------------------------------------------------
   Fa0/1            2             2              0                Shutdown
------------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)    : 1
Max Addresses limit in System (excluding one mac per port) : 6144
```

Информация в выводе не слишком подробна, но ее достаточно, чтобы понять, что происходит. Когда вы включаете функцию Port Security, она отмечает MAC-адреса, которые в это время задействованы на порту, и сохраняет их в количестве, не превышающем максимального значения, указанного вами. Оно указано в колонке `MaxSecureAddr`. В данном случае максимальное количество MAC-адресов, разрешенных для порта Fa0/1, равно `2`.

Колонка `CurrentAddr` информирует, сколько MAC-адресов коммутатор обнаружил на порту с момента включения функции Port Security. В этой колонке значение также равно 2, потому что подключено только два устройства.

В колонке SecurityViolation указано, сколько раз коммутатор обнаружил количество MAC-адресов, превышающее допустимый максимум этого порта. Как и следовало ожидать, это значение равно 0.

Последняя колонка, Security Action, является, пожалуй, самой важной. В ней указано действие функции Port Security, которое будет выполнено при обнаружении *нарушения* – ситуации, когда количество MAC-адресов превышает установленный максимум. Это действие в Cisco называется *режимом нарушения*.

5.1.2. Режим нарушения

Давайте научимся настраивать два режима нарушения: отключение и ограничение.

Отключение

Этот режим нарушения отключает порт. Это значит, что если функция Port Security обнаруживает нарушение, то есть еще один MAC-адрес, помимо двух разрешенных, она полностью отключает порт. Без предупреждения, не задавая никаких вопросов.

Этот режим используется в устройствах компании Cisco по умолчанию. Я подозреваю, что это способ побудить выполнить настройку функции Port Security, когда встанет вопрос, почему сеть не работает. Если порт отключается сразу же после включения функции Port Security, это трудно не заметить.

Ограничение

Альтернативный режим нарушения – ограничение – несколько «вежливее». В этом режиме в случае нарушения функция Port Security не отключает порт, а запрещает передачу с дополнительных MAC-адресов. В некотором смысле это похоже на динамический список доступа, который запрещает MAC-адреса за пределами разрешенного максимума.

Вероятно, вам не понравится, если функция Port Security полностью отключит порт, когда обнаружит нарушение. В этом случае вам нужно вручную указать действие в режиме нарушения. Для этого используется команда конфигурирования интерфейса switchport port-security violation restrict.

Практикум

Включите ограничение в режиме нарушения, выполнив команду, указанную ниже:

```
switchport port-security violation restrict
```

Как обычно, проверьте результат, выполнив команду show port-security.

В последней колонке вывода вы должны увидеть изменение значения с Shutdown на Restrict. Все остальное останется неизменным:

```
Switch1#show port-security
Secure Port   MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
```

```
                (Count)       (Count)        (Count)
------------------------------------------------------------------------------
   Fa0/1            2             2              0               Restrict
------------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)    : 1
Max Addresses limit in System (excluding one mac per port) : 6144
```

Теперь при обнаружении функцией Port Security нарушений она не будет отключать порт или иным образом влиять на работу устройств с первыми двумя MAC-адресами. Они будут продолжать обмениваться трафиком нормально, и только дополнительные адреса будут блокироваться.

Дополнительно

Режим нарушения `restrict` можно использовать для того, чтобы никто не смог настроить беспроводную точку доступа по технологии Power over Ethernet (PoE). Когда система IOS отключает порт на коммутаторе с поддержкой технологии PoE, она отключает и питание устройства, подключенного к нему. Вот почему использовать режим нарушения `shutdown` на портах с IP-телефонией как-то не хочется.

5.2. Проверка функции Port Security

Один из самых интересных аспектов работы с функцией Port Security – это тестирование. Для этого вам не нужно инициировать собственную атаку по MAC-адресу. Все, что вам нужно сделать, – это получить еще один MAC-адрес, с которого будут поступать обращения к защищенному порту. Существует несколько способов сделать это.

Если вы имеете дело с компьютером и IP-телефоном, отключите компьютер от телефона и подключите ноутбук. После того как коммутатор увидит MAC-адрес ноутбука, функция Port Security зарегистрирует нарушение и заблокирует MAC-адрес ноутбука.

Если у вас только один компьютер, подключите коммутатор рабочей группы между коммутатором Cisco и компьютером. Возьмите пару ноутбуков или IP-телефонов и подключите их к коммутатору рабочей группы. Так вы получите три MAC-адреса на одном и том же порту – достаточно, чтобы вызвать нарушение для функции Port Security.

Практикум

Важно, чтобы вы внимательно отслеживали происходящие события, когда тестируете функцию Port Security. Система IOS позволяет в реальном времени выводить информацию о состоянии функции Port Security. Введите в оболочке командной строки команду `terminal monitor`.

Затем используйте один из методов, которые я только что перечислил, для проверки работы функции Port Security.

После подключения третьего устройства вы увидите сообщение, подобное этому:

```
%PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by
    MAC address 0800.27ba.dbad on port FastEthernet0/1.
```

В сообщении указаны порт, на котором произошло нарушение, и MAC-адрес, который его вызвал. Это полезная информация, которую нужно знать при тестировании.

Теперь если вы опять выполните команду `show port-security`, то должны увидеть увеличение значения в колонке `SecurityViolation`:

```
Switch1#sh port-security
Secure Port   MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                 (Count)       (Count)        (Count)
-----------------------------------------------------------------------------
   Fa0/1            2             2              18             Restrict
-----------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)    : 1
Max Addresses limit in System (excluding one mac per port) : 6144
```

Значение 18 удивляет, учитывая, что функция Port Security должна была заблокировать только один MAC-адрес. Секрет в том, что счетчик `SecurityViolation` увеличивается каждый раз, когда неавторизованный MAC-адрес пытается передать кадр данных. Если вы правильно настроили допустимое количество MAC-адресов, это число не должно быть очень большим. Если это значение не равно нулю, обратите внимание на устройства, подключенные к этому порту.

Дополнительно

Вы можете сбросить счетчик `SecurityViolation`, отключив порт и снова включив его. На момент написания этой статьи нет специальной команды, чтобы непосредственно сбрасывать счетчики.

5.3. Перемещение устройств

Я уже упоминал ранее, что функция Port Security определяет порядок обслуживания запросов по принципу «первым пришел – первым обслужен». Когда вы физически отключаете устройство от защищенного порта, функция Port Security забывает все MAC-адреса, зафиксированные на этом порту. Таким образом, если вы подключите другое устройство к этому порту, функция Port Security допустит его. Это удобно в случаях, когда перемещение устройства влечет за собой физическое отсоединение от коммутатора. Например, когда пользователь переходит за другой стол, он физически отключает свой компьютер и IP-телефон от коммутатора.

Но есть и другая возможность. Предположим, что системному администратору необходимо одновременно подключить пять новых компьютеров к сети,

установить на них программное обеспечение, загрузить обновления и т. д., чтобы подготовить их для новых пользователей. Но есть проблема: в офисе, где они работают, доступен только один сетевой разъем. Чтобы сохранить работоспособность всей системы и включить в сеть новые компьютеры, он подключает небольшой коммутатор рабочей группы с восемью портами и подключает к нему все новые компьютеры.

5.3.1. Port Security помнит все!

В аппаратном шкафу коммутатор рабочей группы подключен к порту FastEthernet0/12 коммутатора. Вы выполнили задание и знаете, что нельзя одновременно подключить более пяти устройств с MAC-адресами к порту, к которому подключен коммутатор рабочей группы. Вы настроили функцию Port Security на ограничение количества (5) доступных к подключению устройств.

Практикум

Вполне возможно, что у вас нет коммутатора рабочей группы. Он нужен лишь для практики. Используйте следующие команды для настройки функции Port Security, чтобы разрешить не более пяти MAC-адресов на порту FastEthernet0/12:

```
interface fa0/12
switchport port-security maximum 5
switchport port-security violation restrict
switchport port-security
```

После того как пять компьютеров загружены администратором, каждый из них начинает пересылать трафик со своим уникальным MAC-адресом. Все работает так, как ожидалось, и компьютеры могут нормально взаимодействовать с сетью. Команда `show port-security` подтверждает, что функция Port Security включена и ничего не заблокировано:

```
Switch1#show port-security
Secure Port   MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                 (Count)       (Count)        (Count)
-------------------------------------------------------------------------------
   Fa0/1            2             0              0              Restrict
   Fa0/12           5             5              0              Restrict
-------------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)    : 4
Max Addresses limit in System (excluding one mac per port) : 6144
```

После окончания настройки администратор завершает работу компьютеров и подключает к коммутатору рабочей группы пять новых устройств, чтобы настроить их. Но теперь возникает следующая проблема. Ни один из компьютеров не может подключиться к сети. Вы снова проверяете функцию Port Security и видите следующее:

```
Switch1#show port-security
Secure Port   MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                 (Count)       (Count)        (Count)
------------------------------------------------------------------------------
   Fa0/1            2             0              0             Restrict
   Fa0/12           5             5              30            Restrict      ← MAC-адреса
------------------------------------------------------------------------------  заблокированы
Total Addresses in System (excluding one mac per port)    : 4
Max Addresses limit in System (excluding one mac per port) : 6144
```

Функция Port Security не забыла первоначальные пять MAC-адресов. Она все еще помнит их и, следовательно, не позволяет новым компьютерам подключаться.

Важно понять, почему это происходит. Функция Port Security не в курсе, что исходные пять компьютеров были отключены от сети. Коммутатор рабочей группы с восемью портами скрывает это. Функция Port Security помнит пять уникальных MAC-адресов, а затем видит пять новых. В соответствии с настройками функция Port Security допускает только первые пять MAC-адресов и блокирует последующие.

Можно было посоветовать системному администратору просто выключать или перезагружать коммутатор рабочей группы каждый раз, когда он меняет компьютеры, но это непрактично, раздражает и преждевременно изнашивает коммутатор. Вам нужен другой способ, чтобы функция Port Security забыла эти MAC-адреса.

5.3.2. Время старения

Время старения – это параметр, который вы можете настроить, чтобы функция Port Security через определенное время забывала MAC-адреса, которые зафиксировала.

После того как системный администратор завершит работу с первой группой из пяти компьютеров, потребуется около 10 минут, чтобы отключить их, переместить и подключить новую группу. Нужно, чтобы MAC-адреса первой группы устройств устарели, и к тому моменту, когда вторая группа окажется подключена, функция должна Port Security забыть о первых пяти компьютерах.

Практикум

Время старения, как и все другие параметры функции Port Security, устанавливается для каждого порта. Используйте следующие команды, чтобы установить время старения, равное 10 минутам:

```
interface fa0/12
switchport port-security aging time 10
```

Для проверки конфигурации выполните следующую команду:

```
show port-security interface fa0/12
```

Ниже представлен пример результирующего вывода:

```
Switch1#show port-security interface fa0/12
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Restrict
Aging Time                 : 10 mins  ◄── Время старения – 10 минут
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 5
Total MAC Addresses        : 5
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0800.271c.0b57:1
Security Violation Count   : 30
```

В четвертой строке вы можете видеть время старения в минутах. Функция Port Security отслеживает время старения для каждого MAC-адреса независимо от остальных, основываясь лишь на том времени, когда его впервые зафиксировала. Это можно проверить, выполнив команду `show port-security address`:

```
Switch1#show port-security address
          Secure Mac Address Table
------------------------------------------------------------------
Vlan   Mac Address      Type            Ports      Remaining Age
                                                       (mins)
----   -----------      ----            -----      -------------
  1    0800.2742.aab8   SecureDynamic   Fa0/12           6
  1    0800.2782.4c93   SecureDynamic   Fa0/12           6
  1    0800.27b8.b488   SecureDynamic   Fa0/12           6
  1    0800.27e4.bb01   SecureDynamic   Fa0/12           6
  1    0800.7200.3131   SecureDynamic   Fa0/12           6
------------------------------------------------------------------
Total Addresses in System (excluding one mac per port) : 4
Max Addresses limit in System (excluding one mac per port) : 6144
```

Обратите внимание, что для каждого MAC-адреса указано одно и то же значение в колонке `Remaining Age`. Это неудивительно, потому что системный администратор загружал все пять компьютеров одновременно.

Теперь предположим, что он закончил с четырьмя из пяти компьютеров и выключил их. Но один остался включенным, допустим, из-за каких-то проблем. Далее администратор приносит четыре новых компьютера и подключает их. Все по-прежнему работает.

Опять выполняем команду `show port-security address`:

```
Switch1#show port-security address
          Secure Mac Address Table
------------------------------------------------------------------
Vlan   Mac Address      Type            Ports      Remaining Age
                                                       (mins)
----   -----------      ----            -----      -------------
```

```
  1    0800.2708.e69b   SecureDynamic   Fa0/12          9
  1    0800.27b6.b091   SecureDynamic   Fa0/12          9
  1    0800.27c1.5607   SecureDynamic   Fa0/12          9
  1    0800.27f4.803e   SecureDynamic   Fa0/12          9
  1    0800.7200.3131   SecureDynamic   Fa0/12          8      <-- Это устройство не было
----------                                                      заменено и устаревает
Total Addresses in System (excluding one mac per port) : 4      независимо от других адресов
Max Addresses limit in System (excluding one mac per port) : 6144
```

Обратите внимание, что первые четыре MAC-адреса изменились, и время старения у них – 9 минут. Последний адрес, который принадлежит компьютеру, который не отключили от сети, не изменился и имеет время старения 8 минут. Поскольку MAC-адрес этого компьютера уже был в списке допустимых MAC-адресов, он все равно сможет получить доступ к сети даже после завершения работы таймера. Как только таймер достигнет нуля, он будет сброшен до 10 минут.

Теперь, когда вы настроили время старения, вам, вероятно, никогда не придется связываться с функцией Port Security по поводу этого конкретного порта. Если у системного администратора возникла проблема с подключением, все, что ему нужно сделать, – это подождать несколько минут и повторить попытку.

Вероятно, вам придется пробовать несколько раз и разбираться с ошибками, чтобы подобрать подходящее время старения. Если вновь подключенные устройства не могут получить доступа к сети, следует уменьшить время старения. Помните о потребностях пользователя и не думайте, что вам нужно установить чрезмерно длительное время старения. Даже если вы установите очень короткое время старения, скажем, 1 минуту, порт будет по-прежнему защищен от MAC-атак. Установка более длительного времени не требуется по соображениям безопасности. Но если вам требуется более высокий уровень безопасности, функция Port Security может его предоставить.

5.4. Запрещение доступа неавторизованных устройств

Теперь вы знаете, как настроить функцию Port Security для предотвращения атак по MAC-адресам, не ограничивая трафика легитимного пользователя. С опытом, путем проб и ошибок, вы сможете настраивать функцию Port Security на любых портах конечного пользователя, даже не зная, что к ним подключено.

Но хотя минимальная конфигурация функции Port Security и выгодна конечному пользователю, в плане производительности сети она не всем подходит. Некоторые организации предъявляют высокие требования безопасности, например запрещают подключение посторонних устройств к сети. Для этого недостаточно ограничить количество допустимых MAC-адресов, вам нужно указать MAC-адреса *конкретных* устройств, которые имеют доступ к порту. Это кажется сложным, но, как вы увидите, функция Port Security позволяет выполнить настройку на удивление легко.

Даже если ваша организация не требует столь высокого уровня безопасности, я все же настоятельно рекомендую прочитать этот раздел. И вот почему:

из главы 4 вы узнали, что одна из причин отключения неиспользуемых портов заключается в необходимости предотвращения подключения к сети постороннего человека с его зараженным ноутбуком. Но даже если проверять, отключены ли неиспользуемые порты, один раз в рабочий день и пару раз в выходные, это не помешает сотруднику отключить рабочий компьютер и подключить свой зараженный ноутбук.

Вероятно, вы знаете и другие причины, в силу которых ограничение доступа весьма желательно. В начале главы я обещал, что расскажу, как настроить функцию Port Security на максимальный уровень безопасности. Теперь, когда у вас есть основания для этого, я научу вас этим настройкам.

Дополнительно

Безопасность – это наличие защиты. Несмотря на то что любая организация с малой толикой здравого смысла принимает меры, чтобы физически предотвратить появление посторонних людей с вредоносными устройствами, это не отменяет необходимости принимать технические меры для защиты сети. Любая защита может быть разрушена, поэтому все, на что вы можете надеяться, – это затруднить атаку настолько, что злоумышленники сдадутся и перейдут к более легкой цели. Функция Port Security – одна из технологий, которая поможет осложнить жизнь такого недоброжелателя.

5.4.1. Обеспечение максимальной защиты с помощью функции Port Security

Напомню, что при активации функция Port Security запоминает MAC-адреса по мере появления в сети информации о них, вплоть до настроенного максимального количества. Когда устройство, физически подключенное к порту, отключается, функция Port Security забывает его MAC-адрес. Если вы установили время старения равным, скажем, 5 минутам, функция Port Security забывает MAC-адрес через 5 минут после его появления в сети. В сети с максимальным уровнем безопасности функция Port Security должна работать несколько иначе.

Во-первых, необходимо, чтобы запоминались и разрешались только MAC-адреса конкретных устройств, которым разрешен доступ к порту. Во-вторых, не требуется, чтобы когда-нибудь эти MAC-адреса были забыты! Даже если кто-то отключает порт, выключает устройство или перезагружает коммутатор, нужно, чтобы эти MAC-адреса сохранялись в качестве единственных MAC-адресов, допустимых для использования этим портом. Вы можете достичь этого поведения, используя так называемые «*липкие*» *MAC-адреса*.

5.4.2. «Липкие» MAC-адреса

«Липкий» MAC-адрес постоянно хранится в конфигурации запуска, в разделе конфигурирования интерфейса. Причина, по которой он называется «липким», заключается в том, что вам не нужно настраивать его вручную. Вместо этого вы

позволяете функции Port Security обнаружить его как обычно, и система IOS автоматически запишет этот MAC-адрес в рабочую конфигурацию. Это хороший способ добиться высокого уровня безопасности с небольшими усилиями.

Допустим, в вашей организации есть компьютер, который находится в зоне относительно свободного доступа, например в вестибюле или приемной. Вы не хотите, чтобы посторонний мог зайти и подключить свое вредоносное устройство к порту этого компьютера. Поскольку на этом порту допустим только один MAC-адрес, вы указываете максимальное количество адресов – один. Затем вы сообщаете функции Port Security о том, что этот MAC-адрес надо запомнить навсегда, выполнив команду `switchport port-security mac-address sticky`.

Практикум

Выберите порт с одним подключенным устройством и настройте функцию Port Security так, чтобы разрешить только один «липкий» MAC-адрес:

```
interface fa0/1
switchport port-security maximum 1
switchport port-security mac-address sticky
```

И случится чудо! Как только функция Port Security определит MAC-адрес, она запишет его в рабочую конфигурацию. Вы можете проверить это, выполнив команду `show run interface fa0/1`:

```
Switch1#show run interface fa0/1
Building configuration...

Current configuration : 233 bytes
!
interface FastEthernet0/1
 switchport mode access
 switchport port-security
 switchport port-security violation restrict
 switchport port-security mac-address sticky  <-- Вы ввели эту команду
 switchport port-security mac-address sticky 0800.7200.3131  <-- Функция Port Security добавила «липкий» MAC-адрес
End
```

Обратите внимание, что последние две строки конфигурации интерфейса почти идентичны, за исключением MAC-адреса. Первая команда – это та, которую ввели вы, а вторую – добавила функция Port Security.

Дополнительно

Вы могли обратить внимание, что команда `switchport port-security maximum 1` не отображается в конфигурации. Это не ошибка, и это не значит, что вы сделали что-то неправильно. Иногда система IOS изменяет или удаляет определенные команды конфигурации, если они избыточны или не требуются. Функция Port Security по умолчанию допускает только один MAC-адрес для каждого порта, поэтому явно указывать ограничение в один адрес не нужно.

Теперь, для сравнения, выполните команду `show port-security address`:

```
Switch1#sh port-security address
          Secure Mac Address Table
------------------------------------------------------------------
Vlan   Mac Address       Type            Ports      Remaining Age
                                                        (mins)
----   -----------       ----            -----      -------------
  1    0800.7200.3131    SecureSticky    Fa0/1            -
------------------------------------------------------------------
Total Addresses in System (excluding one mac per port) : 0
Max Addresses limit in System (excluding one mac per port) : 6144
```

← Обратите внимание, что тип SecureSticky не имеет времени старения

Здесь отображается один и тот же адрес, а колонка `Remaining Age` пуста, поскольку запись никогда не устареет. Пока вы вручную не удалите конфигурацию, функция Port Security будет помнить этот MAC-адрес.

Практикум

Физически отключите компьютер от порта, на котором вы настроили «липкий» MAC-адрес, и подключите другое устройство. Что произошло?

Вы должны увидеть вывод, подобный показанному ниже:

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
     changed state to down
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to down
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
     changed state to up
%PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred,
     caused by MAC address 2c27.d737.9ad1 on port FastEthernet0/1.
```

← Отключение разрешенного компьютера

Подключение неавторизованного устройства

Функция Port Security заблокировала MAC-адрес неавторизованного устройства

При реальной попытке взлома злоумышленник может потратить несколько минут, пытаясь понять, почему он не может получить доступа к сети. Он может попытаться изменить сетевые настройки, перезагрузиться или подключиться к другому порту. Важно то, что функция Port Security препятствует попыткам получить несанкционированный доступ при любом физическом подключении.

Но не все так гладко, и в этой конфигурации есть один недостаток.

Дополнительно

MAC-адреса можно легко подделать. Изощренный злоумышленник может узнать MAC-адрес авторизованного ПК и клонировать его. Но это потребует времени. Учитывайте, что цель не в том, чтобы полагаться на функцию Port Security как панацею безопасности. Единственная ее задача – затруднить взлом, породив проблемы у злоумышленника.

5.4.3. Предостережение о «липких» MAC-адресах

Дополнительный уровень безопасности «липких» MAC-адресов связан с компромиссом. Если вам когда-либо понадобится заменить устройство, то придется вручную изменить конфигурацию порта, удалить старый MAC-адрес, чтобы новый мог занять его место. В предыдущем примере функция Port Security добавила следующую строку в рабочую конфигурацию:

```
switchport port-security mac-address sticky 0800.7200.3131
```

Алгоритм удаления состоит в следующем: сначала следует убедиться, что этот конкретный MAC-адрес больше не используется на этом порту. Затем надо войти в режим конфигурирования интерфейса и выполнить команду `no`.

Практикум

Отключите сетевой кабель компьютера от порта FastEthernet0/1 или просто выключите порт. Затем выполните следующие команды, чтобы удалить «липкий» MAC-адрес. Обязательно измените MAC-адрес в команде в соответствии с вашей конкретной конфигурацией:

```
int fa0/1
no switchport port-security mac-address sticky 0800.7200.3131
```

Выполните команду `show run int fa0/1`, и «липкий» MAC-адрес должен быть удален.

Вот оно! Функция Port Security автоматически добавит следующий MAC-адрес, который определит как «липкий» MAC-для рабочей конфигурации. Еще один момент, о котором следует помнить, заключается в том, что после того, как функция Port Security запишет «липкий» MAC-адрес в рабочую конфигурацию, вам все равно придется вручную сохранить конфигурацию запуска, чтобы адрес сохранился при перезагрузке коммутатора.

5.5. Команды, использованные в этой главе

Когда вы будете просматривать список команд в табл. 5.4, имейте в виду, что два различных порта могут иметь совершенно разные настройки безопасности. Благодаря этому Port Security – универсальная функция, но это также означает, что при устранении неполадок вам нужно проверять конфигурацию каждого порта.

Таблица 5.4. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`switchport port-security maximum 5`	Интерфейс	Допустимо не более 5 MAC-адресов
`switchport port-security violation restrict`	Интерфейс	Все адреса сверх разрешенного количества блокируются

Окончание табл. 5.4

Команда	Режим конфигурирования	Описание
`switchport port-security violation shutdown`	Интерфейс	Любой MAC-адрес вне разрешенного количества вызывает отключение порта
`switchport port-security`	Интерфейс	Активация функции Port Security
`switchport port-security mac-address sticky`	Интерфейс	Записывает допустимые MAC-адреса в рабочую конфигурацию
`show port-security`	–	Сообщает, на каких портах включена функция Port Security
`show port-security interface fa0/1`	–	Выводит подробную информацию о настройках функции Port Security
`show port-security address`	–	Выводит информацию о разрешенных на порту MAC-адресах
`show run interface fa0/1`	–	Выводит информацию об уровне безопасности FastEthernet0/1

5.6. Практическое задание

Теперь, когда у вас появились практические навыки настройки функции Port Security на нескольких портах, вы готовы включить ее на всех портах подключения конечных устройств. Один лишь незащищенный порт – прямая угроза атаки по MAC-адресу, способной нарушить работу вашей сети.

Выполняя описанные ниже шаги практического задания, не забудьте использовать команду `interface range` для одновременного применения конфигурации к нескольким портам:

1. Начните с настройки максимально допустимого количества MAC-адресов для каждого порта. Если у вас уже есть инструкция о том, сколько может быть подключено к каждому порту, ограничьте предел, выполнив команду `switchport-security maximum`. Если инструкции нет, поэкспериментируйте и установите большое число, например `50`. Максимальное количество MAC-адресов, допустимых для каждого порта, составляет 3072.
2. Затем установите режим нарушения на всех портах, чтобы ограничить их использование, командой `switchport port-security violation restrict`. Позднее вы можете сменить режим на `Shutdown`, если это необходимо, но не начинайте с него.
3. Наконец, задействуйте функцию Port Security, выполнив команду конфигурирования интерфейса `switchport port-security`. Если вы все сделали правильно, ничего драматичного произойти не должно (если вы не находитесь в эпицентре атаки по MAC-адресам). Используйте команды `show`, которые вы изучили ранее, чтобы проверить свою конфигурацию.

Глава 6

Управление виртуальными локальными сетями

Из главы 2 вы узнали, что широковещательные домены не должны быть больших размеров во избежание проблем с производительностью, таких как лавинная передача и прерывания. В большой сети это приводит к неприятному побочному эффекту – необходимости разделять устройства на множество отдельных широковещательных доменов.

Но разделение на домены связано не только с необходимостью поддержания разветвленных сетей, такое разделение имеет дополнительные плюсы. Многие организации, даже небольшие, разбивают свои сети на отдельные домены для обеспечения собственной безопасности. Например, компьютеры отдела кадров образуют один домен, а компьютеры отдела маркетинга – другой.

В главе 2 я проиллюстрировал один из способов построения такой конфигурации с использованием двух отдельных, не связанных между собой напрямую коммутаторов. Хотя это, безусловно, выход, но он не очень эффективен. Наличие как минимум одного коммутатора на один домен требует больших затрат, занимает больше места и требует более дорогого обслуживания. Размещение всех устройств в одном широковещательном домене, конечно, и проще, и дешевле.

Что же делать? Сначала может показаться, что на выбор только два варианта, и оба плохие. Что выбрать – надежную сеть или снижение издержек? К счастью, есть третий вариант, который совместит плюсы первых двух: использовать *виртуальные локальные сети* (virtual LAN, VLAN).

6.1. Что такое виртуальная локальная сеть?

Виртуальная локальная сеть – это не что иное, как широковещательный домен. Уникальность виртуальной сети заключается в том, что это широковещательный домен, который вы определяете произвольно, назначая ему отдельные порты коммутатора. На рис. 6.1 показаны два компьютера, подключенных к одному коммутатору, но в разных виртуальных сетях.

Используя виртуальную сеть, вы можете настроить несколько широковещательных доменов на одном коммутаторе. Нет необходимости устанавливать отдельный коммутатор для каждого домена. Разные порты коммутатора могут принадлежать различным широковещательным доменам в различных виртуальных сетях. Это позволяет произвольно размещать устройства в разных доменах с помощью нескольких простых команд конфигурации.

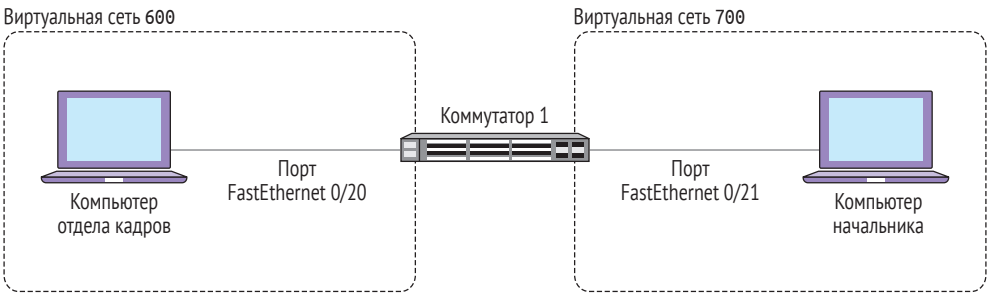

Рис. 6.1 ❖ Компьютеры в отдельных виртуальных локальных сетях. Хотя оба компьютера подключены к одному коммутатору, они находятся в разных широковещательных доменах

6.2. Инвентаризация виртуальных локальных сетей

Как и в любых других ситуациях, вам нужно начать со сбора информации о том, какие виртуальные локальные сети в настоящее время существуют в вашей сети. Учитывайте, что виртуальные сети настраиваются на каждом отдельном коммутаторе. Возможно, на разных коммутаторах настроены различные виртуальные сети. Чтобы убедиться, что вы хорошо понимаете, какие виртуальные сети установлены в вашей сети, выполняйте упражнения этой главы на вашем центральном (основном) коммутаторе. Если вы построили свою собственную тестовую сеть в соответствии с рекомендациями на сайте книги, используйте Коммутатор 1.

6.2.1. База данных виртуальной сети

Виртуальные сети определены на коммутаторе и хранятся в так называемой *локальной базе виртуальных сетей*. Эта база данных хранится во флеш-памяти в файле *vlan.dat* и отличается от конфигурации запуска.

Вы можете просмотреть версию базы данных виртуальных сетей, выполнив команду `show vlan brief`.

Практикум

Авторизуйтесь на коммутаторе и войдите в режим enable.
Выполните следующую команду для просмотра всех виртуальных сетей:

```
show vlan brief
```

Вы увидите список всех виртуальных сетей, настроенных на коммутаторе:

```
Switch1#show vlan brief

VLAN Name Status Ports
---- ------------------- --------- -------------------------------
1    default             active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                   Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                   Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                   Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                   Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                   Gi0/1, Gi0/2
1002 fddi-default        act/unsup
1003 token-ring-default  act/unsup
1004 fddinet-default     act/unsup
1005 trnet-default       act/unsup
```

Результат в вашем случае может слегка (или даже не слегка) отличаться. Это не важно, главное то, чтобы вы сумели понять то, что увидите.

Номер виртуальной сети

В крайней левой колонке, `VLAN`, указаны числовые идентификаторы виртуальных сетей. Коммутатор отслеживает виртуальные сети, используя их числовые идентификаторы. Вы можете создавать новые виртуальные сети с идентификаторами от `2` до `1001` включительно, но не обязательно назначать идентификаторы последовательно. Номера идентификаторов сами по себе не имеют значения, за исключением виртуальной сети `1`. Так называется виртуальная сеть Ethernet по умолчанию, которая присутствует на всех коммутаторах Cisco. Вы не можете удалить или переименовать ее. Я расскажу о виртуальной сети с идентификатором `1` через мгновение.

Имя

Следующая колонка, с заголовком `Name`, содержит имена виртуальных сетей. Вы можете назначить алфавитно-числовые имена всем виртуальным сетям, хотя это и не обязательно. Если вы не присвоите имена вручную, коммутатор сам создаст их на основе идентификаторов виртуальных сетей.

Состояние

Третья колонка, `Status`, должна содержать значение `active` для всех используемых виртуальных сетей. Обратите внимание, что состояние виртуальных сетей `1002` – `1005` указано как `act/unsup` – т. е. активные, но не поддерживаемые. Эти виртуальные сети – это дефолтные не Ethernet-сети, которые предустановлены на каждом коммутаторе Catalyst, и вам не нужно беспокоиться о них. Просто знайте, что вы не можете их использовать, изменять или удалять.

Порты

Колонка `Ports` наиболее важна. Значения в ней сообщают, какие порты какой виртуальной сети назначены. В моем примере все порты назначены виртуальной сети `1`.

Дополнительно

Поскольку база данных виртуальных сетей хранится в отдельном файле, все настроенные виртуальные сети сохраняются даже после полного удаления конфигурации запуска. Если вы хотите удалить базу данных виртуальных сетей, вам нужно удалить файл *vlan.dat*, используя команду `delete flash:vlan.dat`.

6.2.2. Виртуальная сеть по умолчанию

По умолчанию каждый порт коммутатора назначается виртуальной сети 1. Когда вы достаете совершенно новый коммутатор Cisco из коробки или удаляете базу данных виртуальных сетей, все порты коммутатора автоматически присваиваются виртуальной сети 1. Это означает, что все устройства, подключенные к коммутатору, находятся в одном широковещательном домене. Фактически это то же самое, что не использовать виртуальную сеть вообще!

Привязка к виртуальной сети 1, т. е. наличие одного широковещательного домена, является обычной практикой, особенно в небольших организациях, которые не нуждаются или не имеют желания в организации отдельных доменов по отделам или функциональным признакам. И вашей организации может не требоваться дополнительная виртуальная сеть, кроме сети по умолчанию с идентификатором 1. Но всегда сохраняется возможность роста, приобретения **дополнительного** оборудования, поэтому, как создавать новые виртуальные сети, нужно знать.

6.2.3. Сколько виртуальных сетей создавать?

Короткий ответ – столько, сколько нужно вам, и не более того.

Не существует жесткого правила в вопросе, сколько устройств может или должно быть в одном широковещательном домене. Если ваша организация планирует расширять штат, то при наличии 150 компьютеров уже следует задуматься об их размещении в отдельных виртуальных сетях. Если компьютеров порядка 200, создание новых виртуальных сетей уже практически обязательно.

6.2.4. Планирование новой виртуальной сети

Виртуальные сети целесообразно создавать таким образом, чтобы они совпадали с бизнес-единицами, но были достаточно велики, чтобы оправдать свое существование. Например, если в отделе бухгалтерии установлено 50 компьютеров, а в отделе маркетинга – только 10 компьютеров, имеет смысл объединить их в одну виртуальную сеть вместо двух. Нет ничего плохого в том, что в виртуальной сети будет лишь 10 компьютеров, но с административной точки зрения более эффективно минимизировать общее количество виртуальных сетей.

Безопасность и доступность также являются важными критериями. Вероятно, вы не захотите, чтобы компьютеры отдела кадров находились в той же виртуальной сети, что и менее безопасные компьютеры штатного персона-

ла. Потому как если один из компьютеров персонала будет заражен вирусом и начнет лавинную передачу в сеть, было бы желательно, чтобы это не повлияло на работу в сети компьютеров отдела кадров.

Не стоит пропускать эту главу, если вы думаете, что вам не нужны новые виртуальные сети. Дело в том, что, будучи сетевым администратором, вы рано или поздно с этим столкнетесь – и скорее рано, чем поздно. Даже если нет нужды создавать новые виртуальные сети прямо сейчас, все равно следует выполнять упражнения, чтобы получить навыки настройки виртуальных сетей и быть готовым, когда придет время.

6.3. Создание виртуальных локальных сетей

Вы можете создавать, указывать имена и удалять виртуальные сети в *режиме конфигурации виртуальных сетей*. Несмотря на то что база данных виртуальных сетей не сохраняется в конфигурации запуска, все равно необходимо перейти в режим глобальной конфигурации, чтобы внести в нее изменения. Предположим, вы решили создать две новые виртуальные сети: с идентификатором `600` для отдела кадров и с идентификатором `700` для начальства. Начнем с первой.

Практикум

Перейдите в режим глобальной конфигурации, выполнив команду `configure terminal`. Затем создайте виртуальную сеть с идентификатором `600` с помощью следующей команды:

```
vlan 600
```

На этом этапе вы увидите приглашение режима конфигурации виртуальной сети:

```
Switch1(config-vlan)#
```

Выполните команду `exit`, чтобы выйти из режима конфигурации виртуальной сети и сохранить изменения.
Теперь выполните команду `show vlan brief` для проверки результата.

Вы должны увидеть созданную виртуальную сеть:

```
Switch1#show vlan brief

VLAN Name Status Ports
---- ------------------ --------- -------------------------------
1    default            active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                  Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                  Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                  Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                  Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                  Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                  Gi0/1, Gi0/2
600  VLAN0600           active
```

Обратите внимание, что имя виртуальной сети – VLAN0600. Это имя, которое коммутатор назначил для виртуальной сети автоматически, потому что вы явно не указали его. Опять же, пользовательское имя не обязательно, но удобно. Давайте добавим его.

Практикум

Вернитесь в режим конфигурирования виртуальной сети и назначьте виртуальной сети 600 имя HR:

```
vlan 600
name HR
```

Выполните команду exit для выхода из режима конфигурации виртуальной сети. Проверьте результат, выполнив команду show vlan brief.

```
Switch1#show vlan brief

VLAN Name                Status    Ports
---- ------------------- --------- -------------------------------
1    default             active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                   Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                   Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                   Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                   Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                   Gi0/1, Gi0/2
600  HR                  active    ◄── Виртуальная сеть, которую вы только что создали
```

Теперь вывод из базы данных виртуальных сетей отображает название новой сети – HR. Далее создадим виртуальную сеть для начальства.

Практикум

Создайте виртуальную сеть с идентификатором 700 и присвойте ей имя Executives, используя следующие команды:

```
vlan 700
name Executives
exit
```

Проверьте результат, выполнив команду show vlan brief:

```
Switch1#show vlan brief

VLAN Name                Status    Ports
---- ------------------- --------- -------------------------------
1    default             active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                   Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                   Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                   Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20
```

```
                                  Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                  Gi0/1, Gi0/2
600  VLAN0600             active
700  Executives           active    ◄── Обратите внимание, что порты не указаны
```

Отлично! Вы можете сомневаться, но это все, что нужно для создания виртуальной сети. Но создать виртуальную сеть – это лишь полдела. Сейчас все порты коммутатора по умолчанию назначены виртуальной сети 1. У виртуальных локальных сетей 600 и 700 нет портов, поэтому к ним нет доступа ни у одного конечного устройства.

6.4. Назначение виртуальных локальных сетей

Для упрощения назначим порт FastEthernet0/20 виртуальной сети 600 и FastEthernet0/21 – виртуальной сети 700. Как всегда, если вы работаете в рабочей сети, выбирайте свободные порты, чтобы не прерывать работу сети.

6.4.1. Проверка конфигурации порта

Прежде чем изменять назначение порта, всегда рекомендуется проверять его рабочую конфигурацию, чтобы быстро отменить изменения, если что-то пойдет не так. На практике, даже если вы будете осторожны, всегда можно ошибиться и перенастроить не тот порт коммутатора, поэтому очень важно суметь быстро восстановить его первоначальную конфигурацию.

Практикум

Проверьте конфигурацию интерфейса FastEthernet0/20:

```
show run interface fa0/20
```

Вы должны увидеть конфигурацию только для указанного порта:

```
Switch1#show run interface fa0/20
Building configuration...

Current configuration : 68 bytes
!
interface FastEthernet0/20
 switchport mode access    ◄── Порт назначен одной из виртуальных сетей
 shutdown
End
```

Первая строка конфигурации интерфейса – `switchport mode access` – может показаться немного странной. Когда порт назначен виртуальной сети, Cisco называет его *портом статического доступа*, поскольку подключенное к нему устройство всегда обращается к этой виртуальной сети. Но какой виртуальной сети он назначен? Команда `show vlan brief` сообщала, что он назначен виртуальной сети 1, но в конфигурации интерфейса нет указаний на виртуальную сеть 1.

В предыдущей главе я упомянул, что коммутатор удаляет некоторые команды конфигурации, если они избыточны или не нужны. Поскольку виртуальная сеть 1 используется по умолчанию, в конфигурации интерфейса она явно не обозначается.

6.4.2. Настройка доступа к виртуальной сети

Назначение порта для виртуальной сети на языке Cisco называется открытием доступа на этом порту к виртуальной сети. Любое устройство, подключенное к этому порту, становится членом этой виртуальной сети. Команда для открытия доступа к виртуальной сети – `switchport access vlan`, а за ней следует указать идентификатор сети.

Практикум

Назначьте порт FastEthernet0/20 виртуальной сети 600, используя следующие команды:

```
interface fa0/20
switchport access vlan 600
```

Результат опять проверяем командой `show run interface fa0/20`.

Вы должны увидеть отсылку на виртуальную сеть 600:

```
Switch1#show run interface fa0/20
Building configuration...

Current configuration : 96 bytes
!
interface FastEthernet0/20
 switchport access vlan 600   <-- Порт назначен виртуальной сети 600
 switchport mode access       <-- Это порт статического доступа
 shutdown
End
```

Теперь намного понятнее! Нет сомнений, что этот порт принадлежит виртуальной сети 600. Единственный вопрос, который может быть не понятен начинающему, – что такое виртуальная сеть 600. Очевидно, вы знаете, что это виртуальная сеть отдела кадров, потому что вы только что настраивали ее. Но если вы забыли это или если это была другая виртуальная сеть, которую вы не настраивали, нужно проверить базу данных виртуальных сетей, чтобы уточнить информацию.

Практикум

Определите, что такое виртуальная сеть 600, и убедитесь, что ей назначен порт FastEthernet0/20:

```
show vlan brief
```

Теперь посмотрим, сумеете ли вы увидеть изменение в выводе:

```
Switch1#show vlan brief

VLAN Name                Status    Ports
---- ------------------- --------- -------------------------------
1    default             active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                   Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                   Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                   Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                   Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                   Gi0/1, Gi0/2
600  HR                  active    Fa0/20   <-- Обратите внимание, что виртуальной сети 600
700  Executives          active              назначен один порт
```

Сетевой администратор Cisco должен уделять пристальное внимание деталям, которые другие могут пропустить. FastEthernet0/20 – единственный член виртуальной сети 600, и запись Fa0/20 почти сливается с массивом портов виртуальной сети 1, не так ли? Чтобы сделать сообщение более понятным, вы можете использовать несколько иную команду.

Практикум

Добавьте фильтр в команду `show vlan brief` для отображения информации, относящейся только к виртуальной сети 600:

```
show vlan brief | i VLAN|600
```

Вы должны увидеть что-то вроде этого:

```
VLAN Name                Status    Ports
600  HR                  active    Fa0/20
```

Намного лучше! Виртуальная сеть 600, единственный член которой – FastEthernet0/20. На работе у вас, скорее всего, будет больше виртуальных сетей и больше портов, поэтому вывод будет еще более запутанным. Использование встроенных функций фильтрации повысит читабельность вывода и избавит вас от потенциальных ошибок.

6.4.3. Настройка режима доступа

Вы назначили порт FastEthernet0/20 для виртуальной сети с идентификатором 600 и именем HR, но у виртуальной сети 700 портов нет. Сделайте FastEthernet0/21 портом доступа к виртуальной сети 700. Но прежде чем это сделать, следует проверить рабочую конфигурацию данного порта.

Практикум

Определите, на работу с какой виртуальной сетью настроен порт FastEthernet0/21:

```
show run interface fa0/21
```

Вы должны увидеть следующий раздел интерфейса рабочей конфигурации:

```
Switch1#show run int fa0/21
Building configuration...

Current configuration : 44 bytes
!
 interface FastEthernet0/21
 shutdown
end
```

Сообщение выглядит как-то не так. Отсутствует команда `switchport mode access`. Чтобы добавить этот порт в виртуальную сеть `700`, вам нужна еще и команда `switchport access vlan 700`.

Практикум

Сделайте FastEthernet0/21 портом статического доступа в виртуальной сети `700`:

```
interface fa0/21
switchport mode access
switchport access vlan 700
```

Затем проверьте настройки, выполнив команду `show run interface fa0/21`.

Теперь выходное сообщение содержит обе команды:

```
Switch1#show run interface fa0/21
Building configuration...

Current configuration : 96 bytes
!
 interface FastEthernet0/21
 switchport access vlan 700
 switchport mode access
 shutdown
end
```

Важно понимать, что эти две команды служат двум различным целям. Команда `switchport mode access` заставляет порт всегда быть портом статического доступа для виртуальной сети, а сама сеть указывается в команде `switchport access vlan 700`.

Один порт может быть назначен нескольким виртуальным сетям, подробнее вы узнаете об этом в главе 10. Единственная причина, по которой я сейчас упоминаю об этом, чтобы было понято, зачем нужны обе команды.

6.5. Виртуальная сеть для пропуска голосового трафика

Многие организации используют существующую инфраструктуру компьютерной сети для обеспечения телефонии с поддержкой VoIP. Если вы знакомы с тем, как IP-телефоны подключаются к сети, то знаете, что телефон подклю-

чается к порту коммутатора, а компьютер подключается к телефону. Другими словами, телефон находится между коммутатором и компьютером. Это порождает интересную дилемму, когда речь заходит о виртуальной сети.

Предположим, вы подключили IP-телефон к порту FastEthernet0/20, который является членом сети 600 (HR). Затем вы подключаете компьютер к телефону. И компьютер, и телефон находятся в виртуальной сети 600. А проблема вот в чем: трафик голосовых сообщений особенно чувствителен к задержкам. Любая интенсивность в виртуальной сети HR может исказить или прервать голосовые вызовы. Например, если сотрудник отдела кадров проводит собеседование с потенциальным сотрудником по IP-телефону, то загрузка на компьютере может вытеснить голосовой трафик.

Оборудование Cisco позволяет избежать этой проблемы путем назначения отдельной виртуальной сети для IP-телефонии. Получается так называемая *виртуальная сеть для пропуска голосового трафика*. Такая сеть не представляет собой особый тип виртуальной сети. Это просто термин, который Cisco использует для описания виртуальных сетей, выделенных под IP-телефонию. Если ваша организация использует IP-телефоны, у вас уже должна быть настроена сеть для голосового трафика. Скорее всего, она называется наподобие VOICE.

Ну а если виртуальная сеть для голосового трафика уже есть, создавать другую не нужно. Но даже если вам это не нужно, создайте виртуальную сеть с идентификатором 20 и именем VOICE для практики.

Практикум

Используйте следующие команды для создания виртуальной сети 20 и присвойте ей имя VOICE:

```
vlan 20
name VOICE
exit
```

Убедитесь, что созданная виртуальная сеть внесена в базу данных виртуальных сетей:

```
sh vlan brief | i VLAN|20
```

Не беспокойтесь, назначены ли порты виртуальной сети для голосового трафика. Все, что имеет значение на данный момент, – это то, что она существует:

```
Switch1#sh vlan brief | i VLAN|20
VLAN Name                Status    Ports
20   VOICE               active
600  HR                  active    Fa0/20
```

Просто создать виртуальную сеть и назвать ее VOICE недостаточно. Вам нужен способ отправить голосовой трафик телефона в виртуальную сеть VOICE, а трафик компьютера – в виртуальную сеть HR. Решение состоит в том, чтобы настроить порт FastEthernet0/20 на использование виртуальной сети 20 в ка-

честве сети для голосового трафика. Этого можно добиться, выполнив команду `switchport voice vlan 20`. Эта команда заставляет коммутатор отправлять специальные инструкции на IP-телефон, сообщая ему, что виртуальная сеть `20` используется в качестве сети для голосового трафика.

Практикум

Настройте виртуальную сеть `20` на порту FastEthernet0/20:

```
interface fa0/20
switchport voice vlan 20
```

Проверьте результат, выполнив команду `sh vlan brief | i VLAN|20`.

Теперь в выводе должны отображаться и виртуальная сеть отдела кадров (`HR`), и виртуальная сеть для голосового трафика (`VOICE`):

```
Switch1#sh vlan brief | i VLAN|20
VLAN Name                Status    Ports
20   VOICE               active    Fa0/20
600  HR                  active    Fa0/20
```

Обратите внимание, что один и тот же порт является членом обеих виртуальных сетей. Команда `switchport voice vlan 20` превращает FastEthernet0/20 в *порт мультидоступа к виртуальным сетям*, который может одновременно обращаться к сетям `VOICE` и `HR`. Это не означает, что обе виртуальные сети находятся в одном широковещательном домене, – совсем наоборот. Когда IP-телефон передает голосовой трафик, он добавляет специальный *тег* к каждому голосовому пакету, который указывает на то, что он предназначен для виртуальной сети `20`. Коммутатор использует этот тег, чтобы обеспечить передачу голосового трафика с IP-телефона в виртуальную сеть `20`, а весь прочий трафик с компьютера отправляется в виртуальную сеть `600`.

6.6. Работа в созданных виртуальных сетях

В начале этой главы я представил виртуальную сеть как простой способ создания отдельных широковещательных доменов без необходимости приобретать дополнительное оборудование.

Но как вы помните из главы 2, разделение устройств на различные широковещательные домены создает еще одну проблему: устройство в одном широковещательном домене не может связываться с устройством в другом, используя его MAC-адрес.

Если вы хотите, чтобы устройства из различных широковещательных доменов связывались между собой, вам необходимо предоставить им IP-адреса в разных подсетях, а также указать шлюз для маршрутизации пакетов между доменами. В следующей главе вы узнаете, как создать новые IP-подсети, связать их с двумя виртуальными сетями, которые вы создали, и установить между ними связь.

6.7. Команды, использованные в этой главе

Создание и назначение виртуальных сетей требует использования трех разных режимов конфигурации. Таблица 6.1 поможет вам вспомнить, какие команды работают в различных режимах конфигурации.

Таблица 6.1. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`vlan 700`	Глобальный	Вход в режим конфигурирования виртуальной сети `700`
`name Executives`	Виртуальная сеть	Назначает понятное человеку имя `Executives` указанной виртуальной сети
`exits`	Виртуальная сеть	Сохраняет базу данных виртуальных сетей и переходит в режим глобальной конфигурации
`switchport access vlan 700`	Интерфейс	Назначает порт виртуальной сети `700`
`switchport mode access`	Интерфейс	Назначает порт статического доступа в указанной виртуальной сети
`switchport voice vlan 20`	Интерфейс	Настраивает виртуальную сеть `20` для пропуска голосового трафика
`show vlan brief`	–	Отображает базу данных виртуальных сетей

6.8. Практическое задание

Выполните все упражнения, приведенные в этой главе, если вы этого еще не сделали. По окончании сохраните рабочую конфигурацию, выполнив команду `write memory`.

Прежде чем продолжить, я рекомендую выделить два тестовых компьютера, подключенных к портам FastEthernet0/20 и FastEthernet0/21 или любым другим, которые вы использовали в упражнениях. Если у вас нет свободных компьютеров, не беспокойтесь. Вы все равно сможете выполнять упражнения.

Глава 7

Преодоление барьера виртуальной сети с помощью коммутируемых виртуальных интерфейсов

Хотя разделение устройств на отдельные виртуальные локальные сети предотвращает лавинную передачу (в плане широковещательных доменов) и обеспечивает некоторую безопасность, это создает еще одну проблему: устройства в отдельных виртуальных сетях не могут взаимодействовать друг с другом – по крайней мере, без небольшой помощи.

В конце прошлой главы я рекомендовал подключить два тестовых компьютера к двум настроенным портам доступа виртуальных сетей. Если вы не смогли этого сделать, не волнуйтесь. Вы можете продолжать выполнять большинство упражнений, и я предоставил снимки экрана, используя в качестве примера компьютеры, показанные на рис. 7.1.

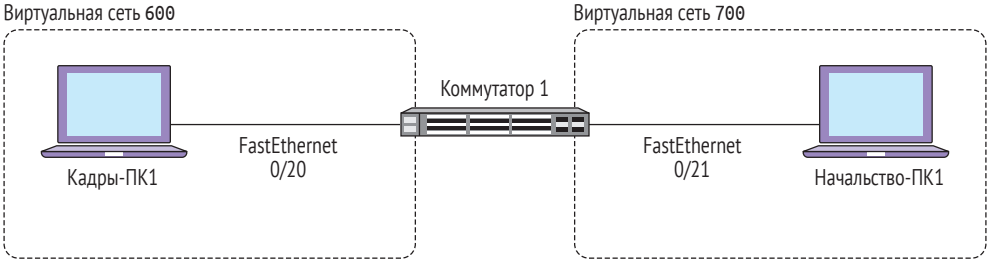

Рис. 7.1 ❖ Два компьютера в разных виртуальных сетях

Как вы можете видеть, компьютер Кадры-ПК1 подключен к FastEthernet0/20 – порту доступа в виртуальную сеть 600, а Начальство-ПК1 – к порту

FastEthernet0/21 в виртуальной сети 700. Как показано, эти два устройства не могут взаимодействовать друг с другом. Фактически они не могут обмениваться данными с любыми устройствами вне их соответствующих виртуальных сетей. Создание новых виртуальных сетей и настройка портов доступа – это первый шаг, но вам нужно сделать еще кое-что, прежде чем вы сможете начать *использовать* созданные виртуальные сети.

В этой главе я расскажу вам, как сделать так, чтобы эти два компьютера смогли обмениваться данными друг с другом с помощью технологии, называемой коммутируемыми виртуальными интерфейсами (*switched virtual interfaces, SVI*).

7.1. Соединение «виртуальная сеть – подсеть»

Вы не можете присваивать IP-адреса компьютерам наугад. В дополнение к уникальной адресации каждого устройства в виртуальной сети IP-адрес должен каким-то образом обозначать, в *какой* виртуальной сети находится устройство.

Прежде чем вы сможете выделить конкретные IP-адреса, вам необходимо определить, какие IP-*подсети* использовать для каждой виртуальной сети. Учитывайте, что подсеть – это всего лишь ряд IP-адресов, которые обычно находятся в одном и том же широковещательном домене. Конкретные IP-адреса, которые вы определите для каждого устройства, будут выбираться из диапазона доступных IP-адресов в подсети.

В табл. 7.1 перечислены подсети, которые вы будете использовать для каждой виртуальной сети. Вы можете подобрать другую пару подсетей, но если вы делаете это в рабочей сети, убедитесь, что выбранные подсети не конфликтуют с уже существующими. Большинство организаций могут предоставить список подсетей, которые они используют в своей сети.

Таблица 7.1. Подсети и соответствующие виртуальные сети

Виртуальная сеть	Адрес подсети	Маска подсети	Используемый диапазон IP
600	172.31.60.0	255.255.255.0	172.31.60.1–172.31.60.254
700	172.31.70.0	255.255.255.0	172.31.70.1–172.31.70.254

Вы можете заметить, что я употребил новый термин во втором столбце: *адрес подсети*. Адрес подсети вместе с маской подсети фактически определяет конкретный диапазон IP-адресов, которые находятся в подсети. Например, посмотрите адрес и маску подсети для виртуальной сети 700, указанные в табл. 7.2.

Таблица 7.2. Адрес подсети и маска подсети

	Первый октет	Второй октет	Третий октет	Четвертый октет
Адрес подсети	172	31	70	0
Маска подсети	255	255	255	0

Вы можете представить маску подсети как секретный код, который говорит вам, как читать адрес подсети. Число 255 означает «Этот октет должен оставать-

ся неизменным». Число 0 означает «Это может быть любое число от 1 до 254!». Как вариант вы можете представить, что 0 – это переменная, такая как *x*.

Используя эту информацию, давайте выберем IP-адрес для компьютера Начальство-ПК1 (вы вручную настроите компьютер Начальство-ПК1с этим IP-адресом через мгновение). Первые три октета должны оставаться неизменными, поэтому IP-адрес должен начинаться с чисел 172.31.70.*x*. Последний октет, однако, может быть любым числом в диапазоне от 1 до 254. Выбор 1 для последнего октета дает IP-адрес 172.31.70.1.

Дополнительно

Если вы не можете использовать подсети, которые я предложил, то можете выбрать из следующих диапазонов адресов, которые зарезервированы для частных сетей:

```
10.0.0.0–10.255.255.0
172.16.0.0–172.31.255.0
192.168.0.0–192.168.255.0
```

При расчете отдельных IP-адресов не забудьте использовать маску подсети 255.255.255.0.

Я продолжу и назначу этот IP-адрес компьютеру Начальство-ПК1, как показано на рис. 7.2.

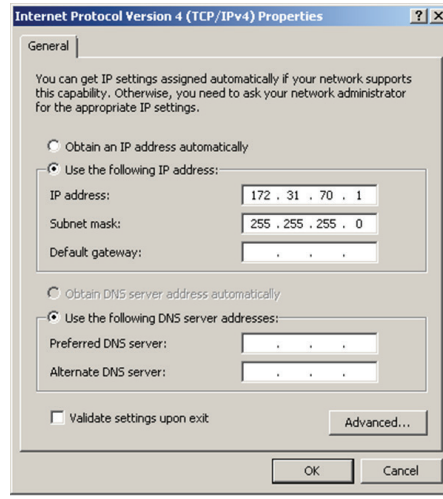

Рис. 7.2 ❖ IP-адрес и маска подсети, настроенные для компьютера Начальство-ПК1

Убедитесь, что поле **Default Gateway** (Шлюз по умолчанию) пустое. Вы заполните это поле немного позже.

Затем вам нужно придумать IP-адрес для компьютера Кадры-ПК1 в виртуальной сети 600. Это подсеть 172.31.60.0. Я выберу адрес 172.31.60.1 и назначу его компьютеру, как показано на рис. 7.3. Обратите внимание, что поле **Default Gateway** (Шлюз по умолчанию) также пустое.

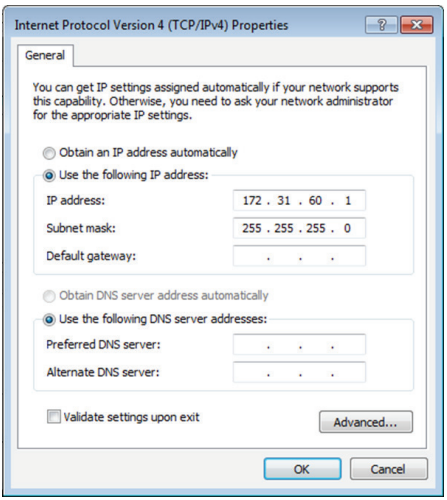

Рис. 7.3 ❖ IP-адрес и маска подсети, настроенные для компьютера Кадры-ПК1

Практикум

Если у вас есть два тестовых компьютера, включенных в виртуальные сети 600 и 700, перейдите к ним и настройте следующим образом:

```
VLAN 600, 172.31.60.1, 255.255.255.0
VLAN 700, 172.31.70.1, 255.255.255.0
```

Даже если у вас нет тестовых компьютеров, продолжайте читать. Вы можете следовать примерам, анализируя их. Самое главное, что вы практикуете команды IOS.

Взглянув на рис. 7.4, вы должны увидеть, что связь между виртуальными сетями и подсетями начинает становиться яснее. Вы произвольно выбрали две IP-подсети и связали их с виртуальными сетями, созданными в предыдущей главе. Но на данный момент компьютеры все еще не могут взаимодействовать друг с другом.

Важно понимать, почему эти компьютеры, несмотря на то что имеют IP-адреса, не могут взаимодействовать друг с другом. Используя компьютер Кадры-ПК1 в качестве примера, давайте вновь рассмотрим, что происходит,

когда компьютер пытается отправить IP-пакет на другой компьютер в другой подсети.

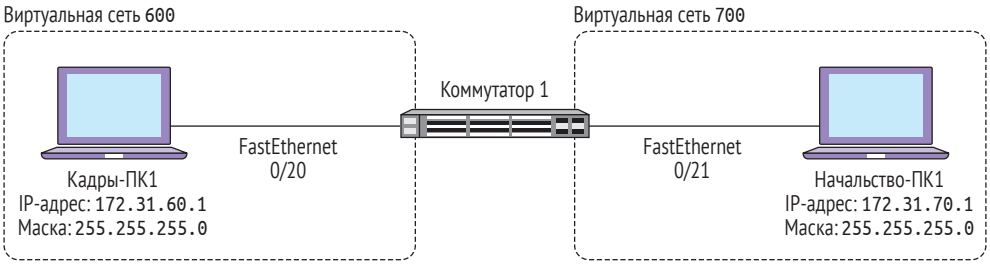

Рис. 7.4 ❖ Компьютеры в отдельных виртуальных сетях с настроенными IP-адресами и масками подсети

Компьютер Кадры-ПК1 имеет IP-адрес `172.31.60.1` с маской подсети `255.255.255.0`. Когда он пытается отправить IP-пакет на компьютер Начальство-ПК1 по адресу `172.31.70.1`, то замечает, что его IP-адрес находится в другой подсети. Он знает это на основе комбинации IP-адреса и маски подсети. Таблица 7.3 иллюстрирует вычисления.

Таблица 7.3. Адрес подсети и маска подсети

Кадры-ПК1	172	31	60	1
Маска подсети	255	255	255	0
Начальство-ПК1	172	31	70	1

Исходя из этого, компьютер Кадры-ПК1 знает, что компьютер Начальство-ПК1 находится в другой подсети. Все, что компьютеру Кадры-ПК1 необходимо сделать, – это создать IP-пакет, адресованный `172.31.70.1`, поместить его внутрь Ethernet-кадра и отправить. Но кому он должен адресовать Ethernet-кадр? Он не может адресовать его адресу управления доступом к сети (MAC) компьютера Начальство-ПК1, потому что даже если бы компьютер Кадры-ПК1 знал бы его (а он не знает), Ethernet-кадр будет остановлен на границе виртуальной сети `600`. Запомните, что Ethernet-кадры не пересекают границ широковещательных доменов.

На рис. 7.5 представлены виртуальные локальные сети `600` и `700` в виде отдельных контейнеров внутри Коммутатора 1. Запомните, что эти виртуальные сети представляют собой изолированные широковещательные домены. Соединяя их вместе, вы создадите единый широковещательный домен, что сделает бесполезным само использование виртуальной сети.

Между этими виртуальными сетями существует невидимый барьер, и единственный способ пересечь его – иметь шлюз по умолчанию – маршрутизатор, подключенный как к виртуальной сети `600`, так и к виртуальной сети `700`.

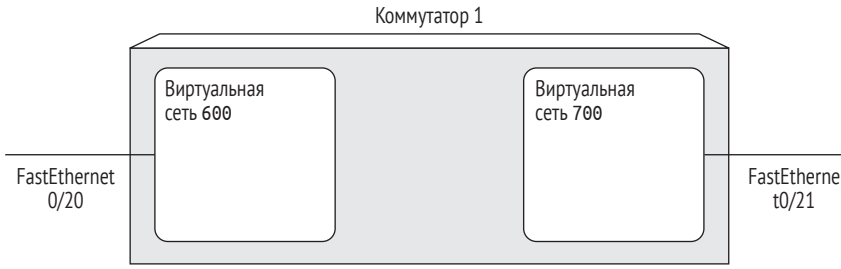

Рис. 7.5 ❖ Крупный план виртуальной сети в Коммутаторе 1. Обратите внимание, что виртуальные сети 600 и 700 никак не связаны

7.2. Коммутаторы или маршрутизаторы?

В главе 2 я подробно объяснил, чем занимаются коммутаторы и маршрутизаторы. Но я не стал делать догматические утверждения, такие как «Маршрутизаторы только маршрутизируют IP-пакеты!» и «Коммутаторы только передают Ethernet-кадры!». Я сознательно избегал таких заявлений, потому что они не всегда верны. Существует довольно много общего между функциональными возможностями, которые предоставляют коммутаторы и маршрутизаторы.

Вы практикуетесь с тем, что сетевые администраторы называют коммутатором уровня 3. Коммутатор уровня 3 обеспечивает функции коммутации и маршрутизации в одном устройстве. Можно представить, что коммутатор уровня 3 имеет крошечный виртуальный маршрутизатор внутри. Вы настроите этот виртуальный маршрутизатор для объединения виртуальных сетей 600 и 700, как показано на рис. 7.6.

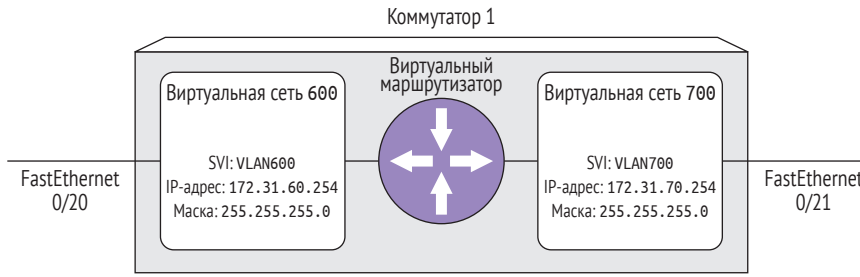

Рис. 7.6 ❖ Крупный план виртуальных сетей, виртуального маршрутизатора и SVI-интерфейсов внутри Коммутатора 1

Вы сделаете три шага, чтобы все заработало. Я кратко изложу каждый из них сейчас, а затем расскажу вам подробности. Во-первых, вам нужно включить функцию IP-маршрутизации на коммутаторе. Она, по сути, включает виртуальный маршрутизатор. По умолчанию он не включен. Во-вторых, вы создадите

два коммутируемых виртуальных интерфейса (SVI) для обеспечения соединения виртуального маршрутизатора с каждой из виртуальных сетей. Наконец, вы назначите IP-адреса каждому из SVI-интерфейсов. Эти IP-адреса станут IP-адресами шлюза по умолчанию, которые будут использоваться компьютерами Кадры-ПК1 и Начальство-ПК1. Может показаться, что всего так много, но не волнуйтесь, если вы еще не совсем поняли, что вы будете делать. Я расскажу подробнее, когда вы дойдете до каждого шага.

Дополнительно

Термин *уровень 3* относится к уровню модели Open Systems Interconnect (OSI), которая примерно описывает назначение IP-адресов и подсетей. Это часто приводится как контраст с *уровнем 2*, который соответствует MAC-адресам и Ethernet-кадрам.

7.2.1. Включение IP-маршрутизации

Первым делом нужно включить IP-маршрутизацию на вашем коммутаторе с помощью команды глобальной конфигурации `ip routing`. В работающей сети IP-маршрутизация, скорее всего, включена, но это не помешает еще раз выполнить команду.

Практикум

Перейдите в режим глобальной конфигурации и включите IP-маршрутизацию:

```
ip routing
```

Проверьте результат, выполнив команду `show ip route`.

Если IP-маршрутизация включена, вы должны увидеть довольно подробный вывод:

```
Switch1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

          192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Vlan1
L        192.168.1.101/32 is directly connected, Vlan1
```

Это так называемая *таблица маршрутизации IP*. Не беспокойтесь, если ваш результат значительно отличается от моего. На самом деле он наверняка будет выглядеть иначе. Детали не важны. Важно то, что вы видите, как минимум, список кодов вверху и, по меньшей мере, одну подсеть IP, указанную внизу. Если это так, IP-маршрутизация включена.

7.3. Что такое коммутируемые виртуальные интерфейсы?

Физический маршрутизатор имеет как минимум два физических интерфейса. В случае виртуального маршрутизатора внутри вашего коммутатора вам необходимо создать и настроить *виртуальный* интерфейс для каждой виртуальной сети. Эти виртуальные интерфейсы называются *коммутируемыми виртуальными интерфейсами*, или SVI для краткости.

Интерфейсы SVI служат двум целям. Первая цель – обеспечить связь между отдельными виртуальными сетями и виртуальным маршрутизатором. Поскольку маршрутизатор и виртуальны сети являются виртуальными, а не физическими, вы не можете просто подключить их с помощью Ethernet-кабелей. Вам нужно подключить их с помощью команд в системе Cisco Internetwork Operating System (IOS).

Вторая цель – предоставить IP-адрес шлюза по умолчанию для устройств в каждой виртуальной сети. Хотя вы узнали о значении шлюза по умолчанию в главе 2, я рассмотрю его снова более подробно. Обратите внимание, что каждый SVI-интерфейс имеет адрес в каждой виртуальной сети, как показано в табл. 7.4.

Таблица 7.4. IP-адреса и маски подсетей SVI-интерфейсов

Виртуальная сеть	IP-адрес	Маска подсети
600	172.31.60.254	255.255.255.0
700	172.31.70.254	255.255.255.0

Эта информация должна выглядеть знакомой. Эти IP-адреса взяты из тех подсетей, которые были созданы для виртуальных сетей 600 и 700 ранее. Непосредственные IP-адреса, выбранные для SVI-интерфейсов, произвольны. Например, я мог бы выбрать адрес 172.31.60.177 для SVI-интерфейса виртуальной сети 600. Но я выбрал 172.31.60.254, потому что 254 – это самое большое значение, разрешенное для последнего октета, что упрощает запоминание. Как вариант вы могли бы вместо этого выбрать адрес 172.31.60.1.

Дополнительно

Хотя 255 является допустимым значением для октета IP-адреса, он зарезервирован для широковещательного IP-адреса. Широковещательный IP-адрес – это IP-наследование широковещательного MAC-адреса (FFFF.FFFF.FFFF). Если вы попытаетесь отправить IP-пакет на адрес 172.31.60.255, ваш компьютер поместит

этот пакет в Ethernet-кадр, адресованный FFFF.FFFF.FFFF. Он никогда не покинет широковещательный домен! Вот почему вы не станете использовать его для обращения к отдельным устройствам.

7.3.1. Создание и конфигурирование SVI-интерфейсов

Теперь, когда вы знаете, что такое SVI-интерфейс, и у вас есть общее представление о его назначении, пришло время создать его для каждой виртуальной сети. После того как вы их создадите, я покажу вам, как их использовать.

Система IOS рассматривает SVI как еще один интерфейс, такой как FastEthernet0/20. Но есть одно исключение: SVI-интерфейс не возникает по умолчанию. Простое создание виртуальной сети, как это описано в предыдущей главе, не создает SVI-интерфейса. Также к этому не приводит включение IP-маршрутизации. Вам нужно явно создавать SVI-интерфейсы.

Практикум

Начнем с создания SVI-интерфейса для виртуальной сети 600. В режиме глобальной конфигурации выполните команду:

```
interface vlan 600
```

Сразу же после выполнения команды вы должны получить хотя бы одно сообщение:

```
Switch1(config)#interface vlan 600
Switch1(config-if)#
*Mar  1 00:10:33.306: %LINK-3-UPDOWN: Interface Vlan600, changed state to up
*Mar  1 00:10:33.314: %LINEPROTO-5-UPDOWN: Line protocol on Interface
     Vlan600, changed state to up
```

Обратите внимание, что после выполнения команды система IOS переводит вас в режим конфигурирования интерфейса. Конкретные сообщения, которые он дает после этого, могут различаться в зависимости от нескольких факторов. Если у вас нет устройства, подключенного к порту FastEthernet0/20 (ваш порт доступа для виртуальной сети 600), или если порт выключен, то интерфейс Vlan600 не появится. Если вы получаете сообщения о том, что интерфейс Vlan600 не работает, не беспокойтесь. До конца главы я расскажу вам, как это исправить. Сейчас важно, что интерфейс Vlan600 существует, даже если он не работает.

После того как интерфейс Vlan600 создан, вам необходимо назначить ему IP-адрес. Это критически важно: назначенный вами IP-адрес *должен* находиться в той же подсети, которую вы создали для виртуальной сети 600. В этом случае я собираюсь использовать адрес 172.31.60.254.

В режиме конфигурирования интерфейса я выполнил команду ip address ?:

```
Switch1(config-if)#ip address ?
  A.B.C.D  IP address
```

```
  dhcp     IP Address negotiated via DHCP
  pool     IP Address autoconfigured from a local DHCP pool
```

Существует три способа назначить IP-адрес SVI-интерфейсу, но почти во всех случаях требуется назначить *статический* IP-адрес. Для этого я наберу адрес `172.31.60.254`, а затем еще один вопросительный знак (`?`):

```
Switch1(config-if)#ip address 172.31.60.254 ?
  A.B.C.D  IP subnet mask
```

Затем система IOS запрашивает маску подсети. Это будет `255.255.255.0`:

```
Switch1(config-if)#ip address 172.31.60.254 255.255.255.0
```

Практикум

В режиме конфигурирования SVI-интерфейса `Vlan600` назначьте IP-адрес `172.31.60.254` с маской подсети `255.255.255.0`:

```
ip address 172.31.60.254 255.255.255.0
```

Проверьте результат, выполнив команду `show ip interface brief | i Vlan600|Status`.

Вот что вы должны увидеть:

```
Interface              IP-Address     OK?  Method  Status   Protocol
Vlan600                172.31.60.254  YES  manual  up       up
```

Если вы это видите, вы успешно настроили SVI-интерфейс для виртуальной сети `600`. Опять же, не беспокойтесь, если в колонке `Status` или `Protocol` указано значение `down`. Вы еще не закончили!

Теперь пришло время создать и настроить SVI-интерфейс для виртуальной сети `700`.

Практикум

Создайте и настройте SVI-интерфейс для виртуальной сети `700` с IP-адресом `172.31.70.254` и маской подсети `255.255.255.0`:

```
interface vlan 700
ip address 172.31.70.254 255.255.255.0
```

Проверьте результат, выполнив команду `show ip interface brief | i 00|Status`.

Вы должны получить следующий вывод:

```
Interface              IP-Address     OK?  Method  Status   Protocol
Vlan600                172.31.60.254  YES  manual  up       up
Vlan700                172.31.70.254  YES  manual  up       up
```

Отлично! Так как вы видите имена и IP-адреса SVI-интерфейсов, вы успешно создали и настроили два новых SVI-интерфейса для виртуальных сетей `600` и `700`. Теперь большой вопрос: что вы будете делать с этими SVI-интерфейсами?

7.4. Шлюзы по умолчанию

Ранее я упоминал, что IP-адреса, назначенные SVI-интерфейсам, станут IP-адресами шлюза по умолчанию, которые будут использоваться компьютерами Кадры-ПК1 и Начальство-ПК1. Напомню, что когда я настроил IP-адреса и маски подсети для этих устройств, я оставил поле **Default Gateway** (Шлюз по умолчанию) пустым. Теперь пришло время его заполнить.

Дополнительно

Если вы заранее знаете, какой IP-адрес будете использовать для SVI-интерфейса, то можете заполнить поле **Default Gateway** (Шлюз по умолчанию) перед созданием SVI-интерфейса. Указание несуществующего шлюза по умолчанию не вызовет никаких проблем, но с этого устройства не будет доступа к другим подсетям до тех пор, пока вы не настроите SVI-интерфейс.

На рис. 7.7 показана текущая топология сети, а также IP-адрес шлюза по умолчанию для каждой виртуальной сети/подсети. Обратите внимание, что IP-адрес шлюза по умолчанию соответствует подсети виртуальной сети, в которой он находится.

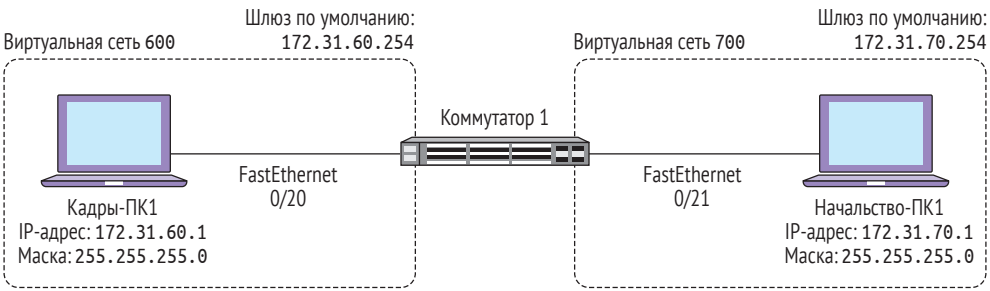

Рис. 7.7 ❖ IP-адрес шлюза по умолчанию для каждой подсети

Я начну с настройки компьютера Кадры-ПК1, как показано на рис. 7.8.

Запомните, что IP-адрес шлюза по умолчанию должен находиться в той же подсети, что и IP-адрес компьютера. В тестовой сети это не проблема, но когда вы работаете в реальной среде, это то, что всегда стоит проверять.

Затем я настрою компьютер Начальство-ПК1 со своим шлюзом по умолчанию, как показано на рис. 7.9.

Практикум

Настройте IP-адрес шлюза по умолчанию на каждом из ваших тестовых компьютеров.

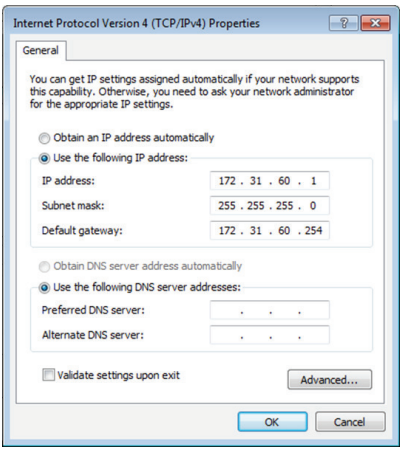

Рис. 7.8 ❖ Настройка IP-адреса, маски подсети и шлюза по умолчанию на компьютере Кадры-ПК1

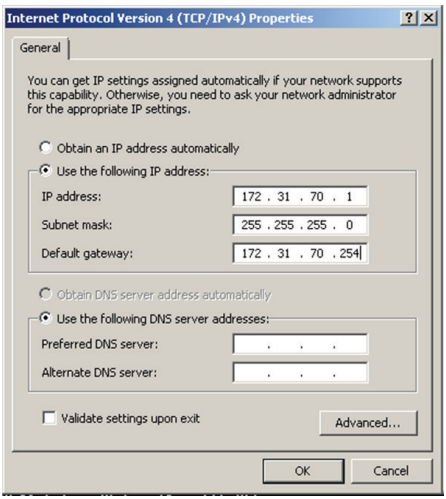

Рис. 7.9 ❖ Конфигурация IP-адреса, маски подсети и шлюза по умолчанию на компьютере Начальство-ПК1

7.4.1. Тестирование соединения между виртуальными сетями

Теперь пришло время проверить, действительно ли компьютер Кадры-ПК1 в виртуальной сети `600` может взаимодействовать с компьютером Начальство-ПК1 в виртуальной сети `700`. Один из наиболее распространенных способов проверки – использовать команду `ping`.

С компьютера Начальство-ПК1 я буду пинговать IP-адрес компьютера Кадры-ПК1, `172.31.60.1`. На рис. 7.10 показаны многообещающие результаты.

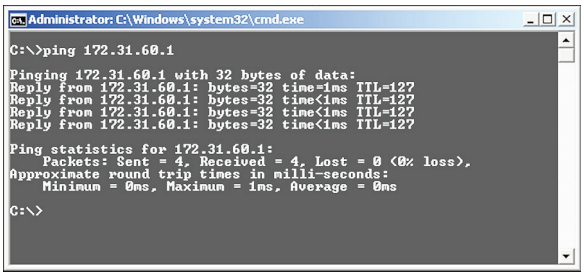

Рис. 7.10 ❖ Успешный пинг с компьютера Начальство-ПК1 (`172.31.70.1`) компьютера Кадры-ПК1 (`172.31.60.1`)

Сообщение `Reply from` демонстрирует, что компьютер Кадры-ПК1 получил и ответил на мой тестовый пинг.

Практикум

Используйте команду `ping` для проверки подключения между вашими тестовыми компьютерами. Предполагая, что вы используете ту же схему IP-адресов, что и я, применяйте следующие команды:

```
ping 172.31.60.1
ping 172.31.70.1
```

В случае успеха вы получите сообщение `Reply from`.

7.5. Команды, использованные в этой главе

Для включения IP-маршрутизации и создания новых SVI-интерфейсов требуется лишь небольшое количество команд. Взгляните на команды в табл. 7.5, а затем попрактикуйтесь их использовать при выполнении практического задания.

Таблица 7.5. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`ip routing`	Глобальный	Включает IP-маршрутизацию
`interface vlan 600`	Глобальный	Создает SVI-интерфейс для виртуальной сети `600` и переходит в режим конфигурирования интерфейса
`ip address 172.31.70.254 255.255.255.0`	Интерфейс	Задает IP-адрес и маску подсети выбранного интерфейса
`show ip route`	–	Отображает таблицу IP-маршрутизации

7.6. Практическое задание

Теперь было бы неплохо попрактиковаться в том, что вы узнали в этой главе. Выполните следующие задачи:

1. Создайте новую виртуальную сеть на коммутаторе.
2. Настройте интерфейс коммутатора как порт доступа в этой новой виртуальной сети.
3. Создайте SVI-интерфейс в новой виртуальной сети.
4. Настройте SVI-интерфейс с IP-адресом и маской подсети.
5. Не забудьте сохранить конфигурацию!

Глава 8

Назначение IP-адресов с использованием протокола DHCP

В предыдущей главе вы настроили две виртуальные локальные сети (VLAN), которые обмениваются данными друг с другом, создав две IP-подсети и настроив два коммутируемых виртуальных интерфейса (SVI-интерфейса), по одному для каждой виртуальной сети. Настройка SVI-интерфейса для виртуальной сети – это то, что обычно приходится делать единожды для виртуальной сети, потому что для каждой подсети нужен только один шлюз по умолчанию для доступа к другим подсетям.

Но когда дело доходит до устройств *внутри* виртуальной сети – компьютеров, принтеров, систем безопасности и т. д., задание IP-адреса вручную одного за другим, как вы это делали в предыдущей главе, – слишком утомительное занятие, подверженное ошибкам. Вам нужен способ автоматического назначения IP-адресов из подсети для устройств в любой виртуальной сети.

Протокол динамической конфигурации хоста (*Dynamic Host Configuration Protocol, DHCP*) – это служба, которая принимает ряд IP-адресов из подсети и автоматически назначает их устройствам в виртуальной сети. Она также может автоматически назначать другие сетевые параметры, такие как шлюзы по умолчанию и *серверы доменных имен* (*Domain Name System, DNS*). Многие организации еще используют протокол DHCP для автоматического указания IP-телефонам Cisco, к какому серверу Call Manager подключаться. Независимо от приложения, цель протокола DHCP – избавить вас от необходимости даже касаться конфигурации сети на сетевом устройстве.

В этой главе вы узнаете, как настроить коммутатор Cisco для автоматического назначения IP-адресов с помощью протокола DHCP. Если в вашей организации уже есть DHCP-сервер, вы узнаете, как настроить ваш коммутатор, чтобы эффективно с ним взаимодействовать.

8.1. Коммутатор или не коммутатор?

DHCP не является проприетарным протоколом компании Cisco. Фактически большинство серверных операционных систем, включая семейство Windows Server и Linux, предоставляют услуги DHCP из коробки. Даже маршрутизаторы и точки доступа потребительского класса обеспечивают работы с протоколом DHCP (именно так вы получаете IP-адрес при подключении смартфона к домашней сети Wi-Fi). Для этой книги я буду ссылаться на эти устройства как на DHCP-серверы, отличные от Cisco.

Коммутаторы и маршрутизаторы Cisco также могут предоставлять услуги DHCP. Они не всегда делают это по умолчанию, поэтому потребуется некоторая настройка. К счастью, настройка коммутатора Cisco для автоматического назначения IP-адресов требует ввода лишь пары интуитивно понятных команд системы IOS. Это быстро, понятно, дешево и безболезненно.

Тем не менее большинство сетей, которые я видел, используют DHCP-серверы не Cisco, а другой фирмы. Возможно, в вашей организации уже есть DHCP-сервер не Cisco, настроенный для предоставления IP-адресов. Если это так, вы все равно прочитайте эту главу, потому что DHCP-сервер, даже отличный от Cisco, требует определенной специализированной конфигурации на коммутаторах Cisco.

8.2. Конфигурирование DHCP-сервера Cisco

Предположим, ваша организация решила открыть новый филиал, который изначально не имел достаточного количества людей, чтобы оправдывать наличие серверов. Вместо этого пользователи в филиале подключаются к центру обработки данных для доступа к сетевым ресурсам, таким как серверы файлов и баз данных.

Дополнительно

У вас есть несколько вариантов, когда речь заходит о подключении территориально разделенных расположений. Частные линии T1/E1, виртуальные частные сети MPLS (VPN) и защищенные VPN, которые передают данные через Интернет, – три самых популярных метода. Когда дело доходит до протокола DHCP, конкретный тип соединения расположений обычно не имеет значения.

Взгляните на рис. 8.1. Предположим, что Коммутатор 1 и два компьютера, с которыми мы работали, Начальство-ПК1 и Кадры-ПК1, находятся в этом новом гипотетическом филиале. Вспомните из прошлой главы, что вы настроили на каждом из этих компьютеров статические IP-адреса.

Ваша задача – настроить протокол DHCP на Коммутаторе 1 для предоставления IP-адреса каждой из машин в этом филиале. Поскольку коммутатор находится в том же офисе, что и компьютеры, он будет работать даже в случае сбоя подключения к центру обработки данных.

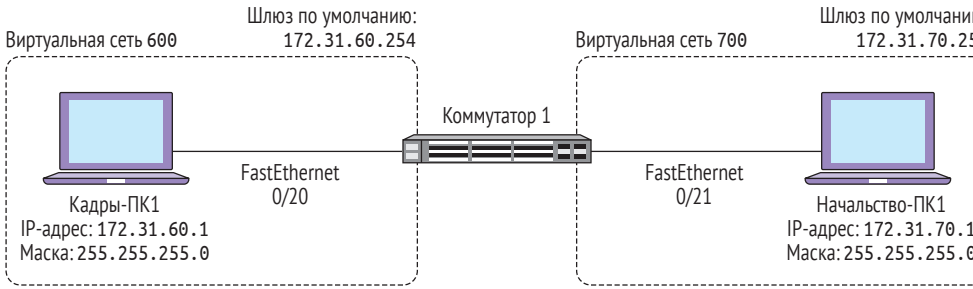

Рис. 8.1 ❖ Компьютеры Кадры-ПК1 и Начальство-ПК1 с жестко заданными IP-адресами

8.2.1. Области адресов

Прежде чем вы сможете перейти к настройке DHCP-сервера, вам нужно узнать еще кое-что, помимо того, какой IP-подсети DHCP-сервер должен назначать адреса.

Цель DHCP-сервера – автоматизировать назначение не только IP-адресов, но и всех параметров конфигурации сети, которые вам в противном случае пришлось бы настраивать вручную. Чтобы вспомнить некоторые настройки, еще раз взгляните на сетевые параметры для компьютера Начальство-ПК1 на рис. 8.2.

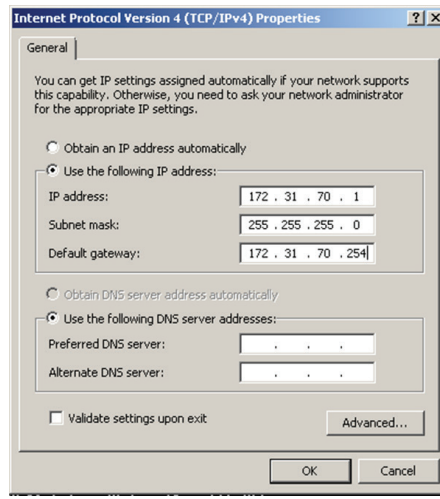

Рис. 8.2 ❖ Сетевые настройки компьютера Начальство-ПК1

Вы уже заполнили несколько полей, включая **IP Address** (IP-адрес), **Subnet Mask** (Маска подсети) и **Default Gateway** (Шлюз по умолчанию). Также есть два пустых поля: предпочтительный и альтернативный DNS-серверы.

DHCP-сервер может автоматически заполнять каждое из этих полей, а затем некоторые другие. Напомню, что подсеть IP и маска подсети вместе предоставляют ряд IP-адресов в этой подсети. В отношении к протоколу DHCP этот диапазон называется *областью адресов*. При минимальной настройке DHCP-сервера вам нужно определить область адресов, обозначая *пул*, из которого коммутатор выбирает и назначает IP-адреса.

8.2.2. Опции

Другие поля – **Default Gateway** (Шлюз по умолчанию) и оба поля **DNS server** (DNS-сервер) – это *опции DHCP*. Они называются опциями, потому что протокол DHCP их не требует. Но это не значит, что они не важны. Современные организации почти повсеместно используют шлюзы IP и DNS для доступа к внутренним ресурсам и Интернету.

Другая менее известная опция DHCP – это *доменное имя*. Современные операционные системы позволяют указать имя домена, чтобы помочь с разрешением имен DNS. Например, предположим, что у меня есть сервер с именем **fileserver.benpiper.com**. На своем компьютере я хотел бы получить доступ к этому серверу, просто набрав «fileserver» и не набирая «benpiper.com». Я могу это сделать, настроив доменное имя моего компьютера как **benpiper.com**. Когда я пытаюсь получить доступ к «fileserver», мой компьютер автоматически добавит «benpiper.com», чтобы получить полное доменное имя **fileserver.benpiper.com**. DHCP-сервер позволяет автоматически назначать имя домена компьютерам в сети, чтобы можно было использовать этот вид сокращений в сети.

8.2.3. Время аренды

Не имеет смысла ситуация, если бы DHCP-сервер выдал IP-адрес компьютеру и позволил этому компьютеру использовать его всегда. В конце концов, в подсети закончатся IP-адреса. Это называется *исчерпанием DHCP*. Чтобы этого избежать, протокол DHCP требует, чтобы вы указали, как долго компьютер может использовать IP-адрес, прежде чем он должен вернуться к DHCP-серверу. Такое поведение называется *временем аренды*.

Время аренды может варьироваться от пары минут до нескольких лет. Продолжительность срока аренды зависит от того, как часто компьютеры перемещаются и выносятся из здания. Если пользователи приносят свои ноутбуки, подключая их к сети по утрам, а затем берут их домой на ночь, относительно короткий срок аренды, например 8 часов, имеет смысл. С другой стороны, если у вас есть офис, полный рабочих столов, которые не перемещаются чаще, чем один или два раза в год, вы можете установить более длительное время аренды, например 30 дней.

Важно отметить, что время аренды – это *минимальный* период времени, в течение которого компьютер может использовать данный IP-адрес. Это не максимум. До истечения срока аренды (примерно на половине срока) компьютер свяжется с DHCP-сервером и запросит *продление*. Если DHCP-сервер не возражает, аренда адреса для компьютера будет продлена.

8.2.4. Подсети и виртуальные локальные сети

Вы настраиваете DHCP-сервер на основе определенной виртуальной сети. Запомните, что рекомендуется поддерживать одну виртуальную сеть, связанную только с одной подсетью. Поскольку все DHCP-серверы выполняют назначение IP-адресов из подсети, использование этого подхода сделает вашу жизнь намного проще. В табл. 8.1 указаны области адресов, параметры и время аренды DHCP-сервера, которые вы будете настраивать для каждой из виртуальных сетей.

Таблица 8.1. Области адресов, опции и время аренды DHCP-сервера для каждой виртуальной сети

Виртуальная сеть	Подсеть	Маска	Шлюз по умолчанию	DNS1	DNS2	Время аренды
600	172.31.60.0	255.255.255.0	172.31.60.254	192.168.100.10	192.168.100.11	7 дней
700	172.31.70.0	255.255.255.0	172.31.70.254	192.168.100.10	192.168.100.11	3 дня, 12 часов

Кстати, вы можете определить адреса шлюза по умолчанию. Это переключаемые виртуальные интерфейсы (SVI-интерфейсы), которые вы настроили в главе 7. Поскольку мы рассматривали компьютер Начальство-ПК1 в виртуальной сети 700, вы начнете с настройки DHCP-сервера для виртуальной сети 700.

8.3. Настройка пула DHCP

Комбинация настроек области адресов, опций и времени аренды DHCP-сервера формирует то, что в компании Cisco называют *пулом* DHCP. Вы настроите основные параметры DHCP-сервера в режиме конфигурации пула DHCP.

Практикум

На Коммутаторе 1 войдите в режим глобальной конфигурации и создайте новый пул DHCP с именем Executives:

```
ip dhcp pool Executives
```

Обратите внимание, что приглашение командной строки изменилось, отражая то, что вы находитесь в режиме конфигурирования DHCP:

```
Switch1(dhcp-config)#
```

Затем вам нужно указать область адресов – подсеть IP и маску. Для виртуальной сети 700 подсетью будет 172.31.70.0 с маской 255.255.255.0:

```
Switch1(dhcp-config)#network 172.31.70.0 255.255.255.0
```

Система IOS назначит IP-адреса из диапазона, специфичного для комбинации этой подсети и маски, – от 172.31.70.1 до 172.31.70.254 включительно. Система IOS выдаст IP-адреса с нумерацией с начала области и двигаясь к концу.

Дополнительно

Некоторые реализации DHCP-сервера, как и многие DHCP-серверы не компании Cisco, не требуют подсети и маски. Вместо этого они запрашивают лишь ряд последовательных IP-адресов. Но за кадром они используют эту информацию для получения подсети и маски. Реализация DHCP-сервера в Cisco пропускает дополнительный шаг и позволяет напрямую вводить подсеть и маску.

Практикум

Теперь пришло время указать параметры. Начните с настройки шлюза по умолчанию для виртуальной сети 700, которым является адрес 172.31.70.254:

```
Switch1(dhcp-config)#default-router 172.31.70.254
```

У сотрудников компании Cisco есть привычка называть одно и то же несколькими именами. Например, в режиме конфигурации DHCP системе IOS требуется команда `default-router`, тогда как в других местах вместо этого требуется команда `default-gateway`. Всегда используйте встроенную справочную систему, чтобы узнать, какую команду вам нужно использовать и когда.

Практикум

Укажите два DNS-сервера, 192.168.100.10 и 192.168.100.11, с помощью единой команды:

```
Switch1(dhcp-config)#dns-server 192.168.100.10 192.168.100.11
```

Если у вас более двух DNS-серверов, которые необходимо указать, вы можете добавить их в команду. Система IOS позволяет указать до восьми уникальных DNS-серверов.

Затем укажите имя домена. Я укажу **benpiper.com**, но вы можете использовать свой собственный:

```
Switch1(dhcp-config)#domain-name benpiper.com
```

Потом укажите время аренды – 3 дня 12 часов 0 минут:

```
Switch1(dhcp-config)#lease 3 12 0
```

Введите команду `exit` и нажмите клавишу **Enter**, чтобы выйти из режима конфигурации DHCP:

```
Switch1(dhcp-config)# exit
```

Наконец, проверьте свою конфигурацию, выполнив команду `show run | section dhcp`.

Вот что вы должны увидеть:

```
Switch1#show run | section dhcp
ip dhcp pool Executives
```

```
network 172.31.70.0 255.255.255.0
dns-server 192.168.100.10 192.168.100.11
default-router 172.31.70.254
domain-name benpiper.com
lease 3 12
```

Обратите внимание, что все подкоманды после `ip dhcp pool Executives` имеют отступы, указывая, что они применяются только к DHCP-пулу `Executives`.

8.4. Исключение адреса из списка выдаваемых адресов

Взгляните на команду `network` в выводе. Ранее я сказал, что система IOS будет назначать IP-адреса последовательно, начиная с `172.31.70.1`, и перемещаться по подсети, пока не назначит последний доступный адрес, `172.31.70.254`. Тут возникает интересная проблема.

Запомните, что Коммутатор 1 уже использует адрес `172.31.70.254` для своего SVI-интерфейса виртуальной сети `700`. Как и следовало ожидать, система IOS достаточно умна, чтобы не назначать свой собственный IP-адрес. Но что, если другие устройства используют статически назначенные IP-адреса? Возможно, есть файловые серверы, брандмауэры, точки беспроводного доступа или принтеры в той же подсети, которым вы хотите статически назначать адрес. Или, может быть, у вас нет таких устройств в настоящее время, но вы хотите исключить диапазон – подмножество IP-адресов из числа назначаемых протоколом DHCP, чтобы вы могли использовать их в будущем.

Вы можете запретить системе IOS назначать IP-адреса, выполнив в режиме глобальной конфигурации команду `ip dhcp excluded-address`. Конкретные адреса, которые вы исключаете, если таковые имеются, вы определяете самостоятельно. В этом примере я исключу диапазоны адресов, указанные в табл. 8.2.

Таблица 8.2. IP-адреса, которые DHCP-сервер не назначает

Диапазоны адресов, исключенные из присвоения DHCP-сервером
172.31.70.1 – 49
172.31.70.251 – 254

Практикум

Перейдите в режим глобальной конфигурации и исключите диапазоны, указанные в табл. 8.2, из числа назначаемых DHCP-сервером:

```
ip dhcp excluded-address 172.31.70.1 172.31.70.49
ip dhcp excluded-address 172.31.70.251 172.31.70.254
```

Для проверки результата выполните команду `show run | section dhcp`.

Вы должны увидеть следующее:

```
Switch1#sh run | s dhcp
ip dhcp excluded-address 172.31.70.1 172.31.70.49
```

```
ip dhcp excluded-address 172.31.70.251 172.31.70.254
ip dhcp pool Executives
 network 172.31.70.0 255.255.255.0
 dns-server 192.168.100.10 192.168.100.11
 default-router 172.31.70.254
 domain-name benpiper.com
 lease 3 12
```

Присваивая адреса устройствам в виртуальной сети 700, Коммутатор 1 будет выбирать их только в диапазоне 172.31.70.50 – 172.31.70.250 включительно.

Обратите внимание, что вы не вводите команды ip-dhcp excluded-address в режиме конфигурации DHCP. Причина в том, что исключения DHCP не зависят от каких-либо пулов DHCP. Фактически вы можете настроить исключения до того, как будут существовать все пулы DHCP! Когда система IOS анализирует рабочую конфигурацию, она помещает исключение перед любым определением пула DHCP.

Практикум

Даже если вы еще не настроили DHCP-сервер для виртуальной сети 600, включите и настройте исключения. Исключите IP-адреса в диапазонах 172.31.60.1 – 49 и 172.31.60.251 – 254:

```
ip dhcp excluded-address 172.31.60.1 172.31.60.49
ip dhcp excluded-address 172.31.60.251 172.31.60.254
```

Следующий шаг – инструктаж компьютера Начальство-ПК1 прекратить использовать статически назначенный адрес и вместо этого запросить адрес, назначенный DHCP-сервером.

8.5. Настройка устройств для запроса адресов у DHCP-сервера

В настоящее время большинство сетевых устройств настроено изначально так, чтобы автоматически запрашивать IP-адрес у DHCP-сервера. Но в данном случае компьютер Начальство-ПК1 настроен на использование статического IP-адреса. Чтобы инструктировать его запросить адрес у DHCP-сервера, вам необходимо установить переключатель в положение **Obtain an IP address automatically** (Получить IP-адрес автоматически), как показано на рис. 8.3.

Положение **Obtain an IP address automatically** (Получить IP-адрес автоматически) переключателя сообщает компьютеру, что он должен получить свой IP-адрес, маску подсети и шлюз по умолчанию от любого DHCP-сервера, который будет обслуживать виртуальную сеть, к которой он подключен.

Другой переключатель и его положение **Obtain DNS server address automatically** (Получить адрес DNS-сервера автоматически) указывает устройству настроить DNS-серверы в соответствии с параметрами DHCP-сервера. В неко-

торых случаях, например при тестировании или устранении неполадок, может потребоваться вручную настроить DNS-серверы. Однако при нормальных обстоятельствах вы будете использовать DNS-серверы, настроенные с помощью DHCP-сервера.

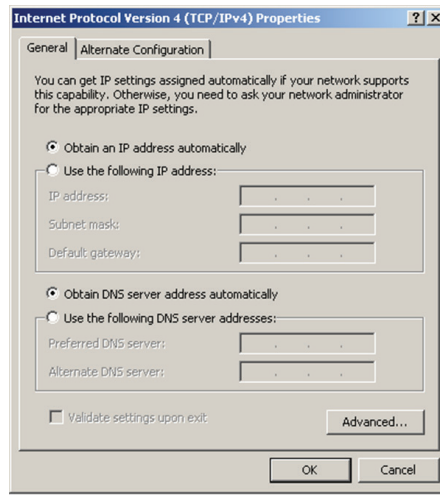

Рис. 8.3 ❖ Настройка компьютера Начальство-ПК1 на работу с DHCP-сервером

После нажатия кнопки **OK**, чтобы принять новые настройки, компьютер Начальство-ПК1 попытается найти DHCP-сервер, чтобы получить адрес. Хотя технические детали того, как устройство запрашивает и получает адрес, выходят за рамки этой книги, есть несколько технических терминов, которые вам нужно знать.

Первое, что делает компьютер Начальство-ПК1, – отправляет сообщение *DHCP Discover*, содержащееся внутри Ethernet-кадра, на MAC-адрес широковещательной передачи. Коммутатор 1 видит этот кадр, проверяет, есть ли у него какие-либо IP-адреса для передачи, доступные в пуле `Executives`, а затем отправляет компьютеру Начальство-ПК1 сообщение *DHCP Offer* о предложении. Предложение содержит IP-адрес, маску подсети, шлюз по умолчанию, DNS-серверы и имя домена – все параметры, которые вы указали при настройке пула `Executives` на Коммутаторе 1.

Дополнительно

Напомню, что Ethernet-кадры, адресованные MAC-адресу широковещательной передачи, остаются в пределах виртуальной сети. Это означает, что DHCP-сервер и клиентские компьютеры должны находиться в одной виртуальной сети, чтобы устройства могли запросить IP-адреса.

Когда компьютер Начальство-ПК1 принимает предложение, он автоматически применяет сетевые настройки. Весь процесс обычно занимает несколько секунд.

Рисунок 8.4 иллюстрирует вывод команды `ipconfig /all` на компьютере Начальство-ПК1, после того как он получил и принял предложение DHCP-сервера.

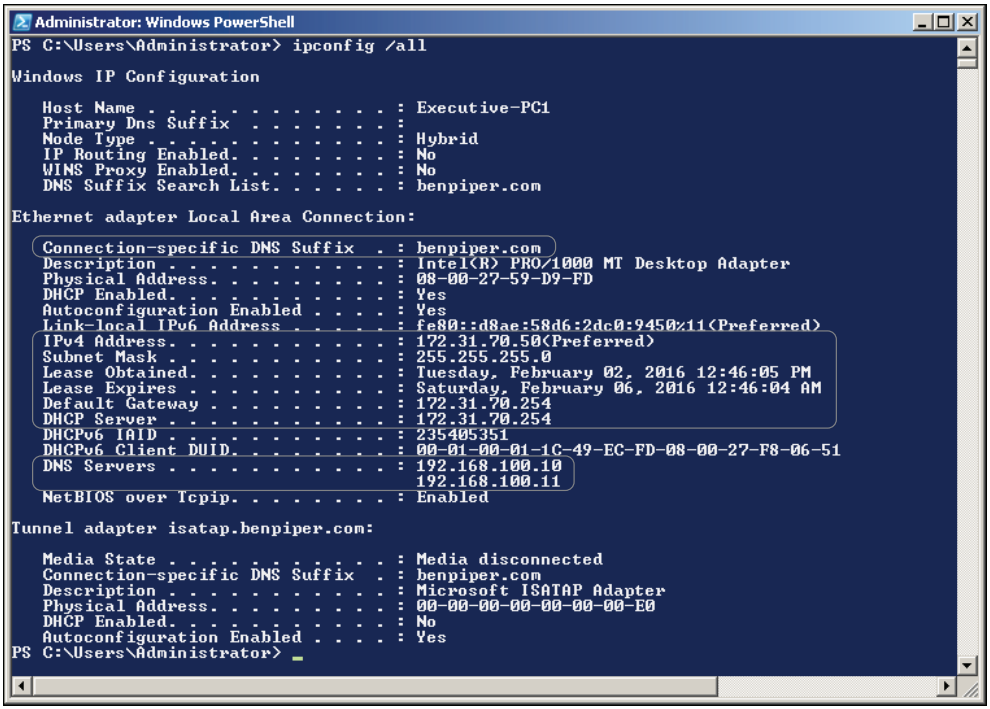

Рис. 8.4 ❖ Результат выполнения команды `ipconfig /all` на компьютере Начальство-ПК1

Все данные внутри выделенных прямоугольных областей получены от DHCP-сервера! Представьте, сколько времени потребуется, чтобы вручную ввести те же сетевые настройки на сотнях компьютеров.

8.6. Ассоциирование пулов DHCP с виртуальными сетями

Пока что вы настроили пул DHCP для виртуальной сети `700`. С этого момента любое устройство, которое подключится к виртуальной сети `700` и запросит IP-адрес у DHCP-сервера, получит один из них с Коммутатора 1.

Но как насчет виртуальной сети 600? Сейчас компьютер Кадры-ПК1 настроен со статическим IP-адресом в подсети 172.31.60.0, как показано на рис. 8.5. Даже если вы хотите перенастроить его для запроса адреса, назначенного DHCP-сервером, Коммутатор 1 полностью проигнорирует его запрос. Причина в том, что вы не настроили пул DHCP для виртуальной сети 600.

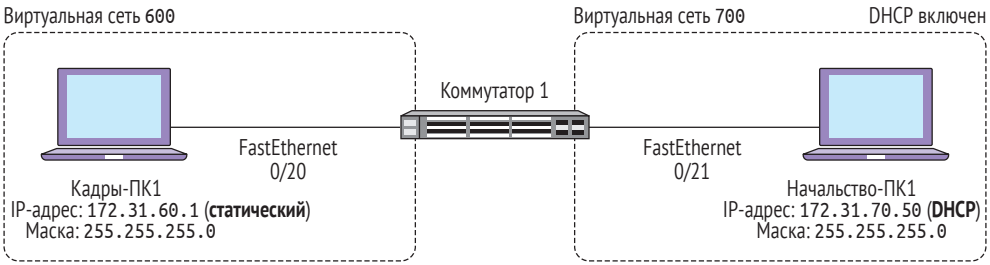

Рис. 8.5 ❖ Протокол DHCP задействован только для виртуальной сети 700

В этот момент я хочу рассказать о том, что вы можете подозревать. Ниже представлена конфигурация пула DHCP для виртуальной сети 700:

```
Switch1#sh run | section pool Executives
ip dhcp pool Executives
 network 172.31.70.0 255.255.255.0
 dns-server 192.168.100.10 192.168.100.11
 default-router 172.31.70.254
 domain-name benpiper.com
 lease 3 12
```

Обратите внимание, что в выводе не говорится о виртуальной сети 700. Никакая строка в этом выводе даже не сообщает о том, что Коммутатор 1 должен предоставлять адреса устройствам только в виртуальной сети 700. Однако ни при каких обстоятельствах Коммутатор 1 никогда не попытается назначить какие-либо IP-адреса в подсети 172.31.70.0 на устройства в виртуальной сети 600 или любой другой.

Причину такого поведения объясняет следующий вывод:

```
Switch1#show ip interface brief vlan 700
Interface   IP-Address            OK? Method Status   Protocol
Vlan700     172.31.70.254         YES manual up       up
```

Коммутатор 1 имеет SVI-интерфейс в виртуальной сети 700, а SVI-интерфейс находится в подсети 172.31.70.0. Поэтому Коммутатор 1 предполагает, что эта подсеть принадлежит исключительно виртуальной сети 700. Когда он видит свою конфигурацию DHCP для пула Executives, он распознает, что подсеть IP, указанная с помощью команды network, соответствует подсети, в которой находится SVI-интерфейс виртуальной сети 700.

Вследствие, когда вы собираетесь настроить новый пул DHCP и указать подсеть 172.31.60.0, Коммутатор 1 поймет, что он должен назначать IP-адреса в этой подсети только для устройств в виртуальной сети 600. Суть в том, что это делает намного проще вашу жизнь как администратора Cisco!

8.7. Создание второго пула DHCP

Теперь пришло время настроить еще один пул DHCP для виртуальной сети 600 (HR). Напомню, что вы уже настроили исключения IP-адресов. Таблица 8.3 содержит все необходимые данные для создания второго пула для виртуальной сети HR.

Таблица 8.3. Область адресов, опции и время аренды DHCP для виртуальной сети 600

Виртуальная сеть	Подсеть	Маска	Шлюз по умолчанию	DNS1	DNS2	Время аренды
600	172.31.60.0	255.255.255.0	172.31.60.254	192.168.100.10	192.168.100.11	7 дней

Практикум

Создайте новый пул DHCP с именем HR. Используйте табл. 8.3 в качестве подсказки:

```
ip dhcp pool HR
 network 172.31.60.0 255.255.255.0
 dns-server 192.168.100.10 192.168.100.11
 default-router 172.31.60.254
 domain-name benpiper.com
 lease 7
```

Проверьте результат, выполнив команду show run | section pool HR.

Компьютер Кадры-ПК1 все еще имеет статический IP-адрес, поэтому следующим шагом будет его настройка, чтобы выдать запрос на получение адреса от DHCP-сервера. На рис. 8.6 показаны два переключателя, которые должны быть выбраны: **Obtain an IP address automatically** (Получить IP-адрес автоматически) и **Obtain DNS server address automatically** (Получить адрес DNS-сервера автоматически).

На рис. 8.7 показаны результаты выполнения команды ipconfig /all на компьютере Кадры-ПК1.

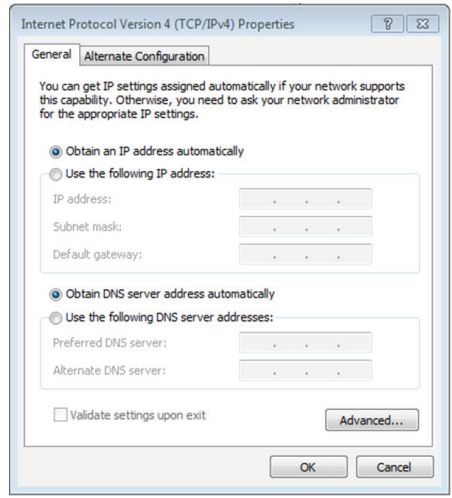

Рис. 8.6 ❖ Настройка компьютера Кадры-ПК1 для использования DHCP-сервера

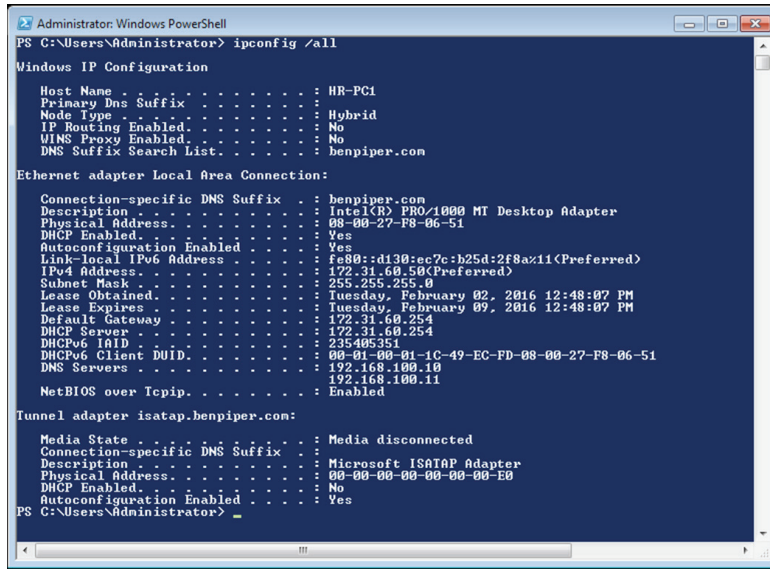

Рис. 8.7 ❖ Вывод команды `ipconfig /all` на компьютере Кадры-ПК1

8.8. Просмотр аренды DHCP

В реальной сети могут быть сотни компьютеров, и если вам нужно просматривать данные каждого из них, переход на каждую машину и выполнение команды `ipconfig` крайне затруднительно. К счастью, есть простой способ просмотреть эту информацию в системе IOS.

Вы можете просматривать активные аренды DHCP непосредственно в оболочке командной строки IOS с помощью команды `show ip dhcp binding`.

Практикум

Просмотрите текущие аренды DHCP, выполнив команду `show ip dhcp binding`.

Вывод должен показать две аренды:

```
Switch1#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address        Client-ID/            Lease expiration       Type
                  Hardware address/
                  User name
172.31.60.50      0108.0027.f806.51     Feb 09 2016 12:47 PM   Automatic
172.31.70.50      0108.0027.59d9.fd     Feb 06 2016 12:45 AM   Automatic
```

Взгляните на первую строку сведений об аренде – для IP-адреса `172.31.60.50`. Он относится к компьютеру Кадры-ПК1. Обратите внимание, что срок аренды близок ко времени истечения срока аренды (строка **Lease Expires**) в выводе команды `ipconfig /all`, показанном на рис. 8.7. Разница примерно в минуту, потому что часы на компьютере Кадры-ПК1 и Коммутаторе 1 не синхронизированы.

Дополнительно

В рабочей сети нередко случается так, что часы отдельных устройств разнятся на несколько секунд или даже минуту или две. Это не проблема, поскольку устройство, настроенное на использование DHCP-сервера, попытается возобновить аренду задолго до ее истечения.

8.9. Использование DHCP-серверов других компаний

В начале этой главы я признал, что могут быть случаи, когда вы просто не хотите использовать коммутатор Cisco в качестве своего DHCP-сервера. Возможно, у вас в сети уже есть рабочий DHCP-сервер другой компании либо вы хотите его настроить.

Настройка DHCP-сервера Windows или Linux выходит за рамки этой книги. Но если вы решите использовать его, вам нужно внести некоторые изменения в конфигурацию коммутатора Cisco, чтобы сеть работала должным образом.

На рис. 8.8 показан гипотетический сервер DHCP сторонней компании в виртуальной сети 900. В остальной части этой главы предположим, что Коммутатор 1 вообще не предоставляет службы DHCP. Также предположим, что компьютеры Кадры-ПК1 и Начальство-ПК1 не имеют назначенных DHCP-сервером IP-адресов.

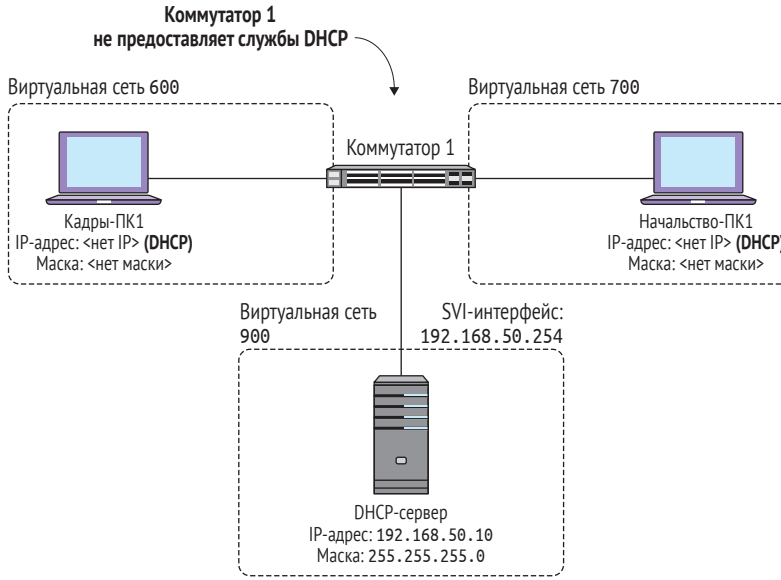

Рис. 8.8 ❖ DHCP-сервер сторонней компании в виртуальной сети 900

На Коммутаторе 1 в виртуальной сети 900 настроен SVI-интерфейс с IP-адресом 192.168.50.254 и маской подсети 255.255.255.0. DHCP-сервер имеет жестко заданный IP-адрес 192.168.50.10. Предположим, что этот DHCP-сервер уже настроен с соответствующими областями и опциями DHCP для виртуальных сетей 600 и 700, которые соответствуют подсетям 172.31.60.0 и 172.31.70.0 соответственно.

Ранее я сказал, что когда устройство пытается получить IP-адрес, оно отправляет сообщение DHCP Discover, содержащееся в Ethernet-кадре, на широковещательный адрес Ethernet. Предположим, что компьютер Кадры-ПК1 отправляет сообщение DHCP Discover в виртуальную сеть 600. Это сообщение распространяется по всей виртуальной сети, но не достигает DHCP-сервера, потому что этот сервер находится в виртуальной сети 900.

Тут возникает, как кажется, непреодолимая проблема. Запомните, что по умолчанию коммутатор не будет перенаправлять Ethernet-кадр за пределы виртуальной сети, где тот возник. Но для того, чтобы DHCP-сервер назначил адрес компьютеру Кадры-ПК1, он *должен* получить сообщение DHCP Discover,

которое находится внутри Ethernet-кадра. Поэтому, чтобы это сработало, некоторые правила должны быть нарушены.

8.9.1. Решение проблемы передачи DHCP Discover с помощью команды ip helper-address

Решение состоит в том, чтобы настроить Коммутатор 1 для пересылки сообщения DHCP Discover из виртуальной сети 600 в виртуальную сеть 900. Возможно, вы думаете, что это приведет к тому, что виртуальные сети 600 и 900 превратятся в один широковещательный домен, полностью уничтожив саму суть виртуальной сети. Но тут есть одна хитрость. Коммутатор 1 не должен пересылать *все* Ethernet-кадры между виртуальными сетями 600 и 900. Вместо этого ему нужно переслать *только* кадры, содержащие DHCP-сообщения!

Команда конфигурирования интерфейса ip helper-address инструктирует коммутатор пересылать *только* выбранные Ethernet-кадры на заданный IP-адрес. В данном случае вы хотите, чтобы Коммутатор 1 переадресовывал все DHCP-сообщения из виртуальной сети 600 на DHCP-сервер по IP-адресу 192.168.50.10.

Практикум

Настройте Коммутатор 1 так, чтобы он пересылал все DHCP-сообщения из виртуальной сети 600 на адрес 192.168.50.10:

```
interface vlan600
ip helper-address 192.168.50.10
```

Проверьте результат, выполнив команду show run interface vlan 600.

Результат выполнения команды ip helper-address в рабочей конфигурации должен выглядеть следующим образом:

```
interface Vlan600
 ip address 172.31.60.254 255.255.255.0
 ip helper-address 192.168.50.10
end
```

Имейте в виду, что эта конфигурация инструктирует Коммутатор 1 пересылать широковещательные сообщения DHCP *только из* виртуальной сети 600. Если вы хотите пересылать широковещательные DHCP-сообщения из виртуальной сети 700, вам придется выполнить ту же команду в режиме конфигурирования SVI-интерфейса виртуальной сети 700.

Практикум

Выполните команду ip helper-address для SVI-интерфейса виртуальной сети 700 следующим образом:

```
interface vlan700
ip helper-address 192.168.50.10
```

8.10. Команды, использованные в этой главе

Если вы помните, как войти в режим конфигурирования пула DHCP, вы можете полагаться на встроенную справочную систему IOS, которая проинформирует вас об остальных командах. Используйте табл. 8.4 в качестве контрольного списка, чтобы убедиться, что вы ничего не забыли.

Таблица 8.4. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`ip dhcp pool MoL`	Глобальный	Создает пул DHCP с именем `MoL`
`network 172.29.1.0 255.255.255.0`	Пул DHCP	Задает область, из которой пул DHCP будет выдавать адреса
`default-router 172.29.1.254`	Пул DHCP	Задает шлюз по умолчанию, который будет передан клиентам DHCP-сервером системы IOS
`dns-server 192.168.10.100 192.168.10.101`	Пул DHCP	Задает DNS-серверы, которые будут переданы клиентам DHCP-сервером системы IOS
`domain-name benpiper.com`	Пул DHCP	Задает доменное имя DNS, которое DHCP-сервер системы IOS будет добавлять к имени клиента
`lease 7 4 15`	Пул DHCP	Задает время аренды IP-адресов, равное 7 дням 4 часам 15 минутам
`ip dhcp excluded-address 172.29.1.1 172.29.1.19`	Глобальный	Исключает адреса `172.29.1.1` – `19` из диапазона адресов, выдаваемых клиентам
`ip helper-address 192.168.5.5`	Интерфейс	Перенаправляет полученные широковещательные сообщения DHCP на указанный адрес (DHCP-сервер сторонней компании по адресу `192.168.5.5`)
`show ip dhcp binding`	–	Показывает текущие выданные DHCP-сервером адреса

8.11. Практическое задание

В этой лабораторной работе вы будете практиковаться в создании нового пула DHCP с нуля. Выполните следующие действия:

1. Создайте пул DHCP с именем `MoL`.
2. Выберите подсеть и настройте пул для назначения IP-адресов из подсети.
3. В подсети выберите IP-адрес для шлюза по умолчанию. Настройте пул, чтобы сообщить клиентам DHCP-сервера об этом шлюзе.
4. Настройте время аренды, равное 14 дням 6 часам.
5. Сохраните конфигурацию!

Глава 9

Обеспечение безопасности сети с помощью списков контроля доступа

В предыдущей главе вы настроили IP-маршрутизацию и коммутируемые виртуальные интерфейсы (SVI-интерфейсы), чтобы позволить хостам в одной подсети обмениваться информацией с хостами в другой. По умолчанию система IOS не ограничивает такую связь между виртуальными сетями. Любое устройство в одной подсети может подключиться к любому устройству в другой, если вы правильно настроили маршрутизацию.

Если бы это были 1990-е годы, вы, вероятно, могли бы остановиться на этом. Но сегодня безопасность представляет собой сложную и важную задачу, и многие организации требуют жесткого контроля за тем, как трафик проходит между устройствами. Если вы хотите производить впечатление серьезного сетевого администратора Cisco, вам нужно знать, как настроить коммутаторы и маршрутизаторы для ограничения IP-трафика в соответствии с этими требованиями.

Наиболее распространенный способ сделать это – использовать *списки контроля доступа* (*access control lists, ACL*). Список ACL – это набор правил, который определяет, может ли данный IP-адрес добраться до другого IP-адреса. Начнем с того, что мысль написать такие правила может показаться невыносимо утомительной. В сети из 5000 устройств вы не сможете написать правила ACL для каждого устройства – и не должны. Хорошая новость в том, что списки ACL чрезвычайно гибкие и мощные, и вы можете охватить большинство ситуаций с небольшим количеством правил.

Когда дело доходит до ограничения трафика, вы чаще всего сталкиваетесь с тремя основными сценариями:

- блокирование доступа с одного IP-адреса к другому IP-адресу;
- блокирование доступа с одного IP-адреса к другой подсети;
- блокирование доступа из одной подсети к другой подсети.

В этой главе я расскажу вам, как настроить списки ACL для каждого из этих сценариев. Чтобы все было правильно, изучите шаги, которые вы будете выполнять:

1) создание списка ACL;
2) добавление правил в список;
3) применение списка ACL к интерфейсу.

9.1. Блокирование трафика «IP–IP»

Взгляните на рис. 9.1. В этом случае любое устройство с IP-адресом в виртуальной сети `600` может обмениваться данными с любым устройством в виртуальной сети `700` и наоборот.

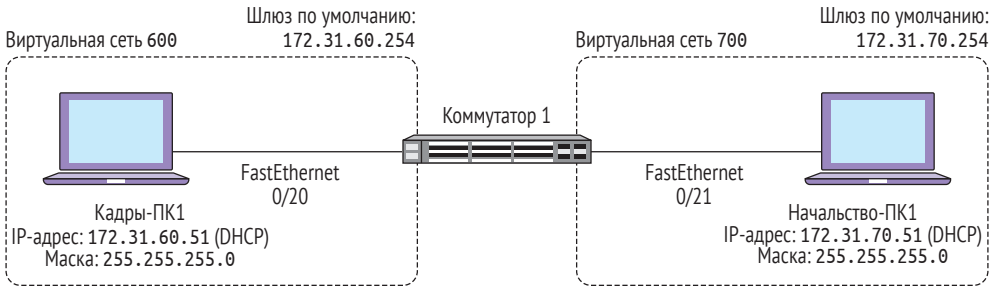

Рис. 9.1 ❖ Компьютеры Кадры-ПК1 и Начальство-ПК1 с назначенными DHCP-сервером IP-адресами

Предположим, вам необходимо предотвратить доступ с компьютера Кадры-ПК1 к компьютеру Начальство-ПК1, но вы не хотите ограничивать трафик с компьютера Кадры-ПК1 любым другим способом. Крайне важно, чтобы вы как сетевой администратор могли понять такое требование и перевести его в IOS-совместимое, перефразируя требование с точки зрения IP-адресов источника и назначения.

В этом случае компьютер Кадры-ПК1 с IP-адресом `172.31.60.51` не сможет достичь компьютера Начальство-ПК1 с IP-адресом `172.31.70.51`. На языке системных администраторов компьютер Кадры-ПК1 является *источником*, а Начальство-ПК1 – *назначением*, как показано в табл. 9.1.

Таблица 9.1. Прототип правила для запрета трафика с компьютера Кадры-ПК1 к компьютеру Начальство-ПК1

Действие	Источник	Назначение
`Deny`	`172.31.60.51` (Кадры-ПК1)	`172.31.70.51` (Начальство-ПК1)

Обратите внимание, что я употребил другой термин: *действие*. Действие – это то, что коммутатор должен сделать с трафиком, который соответствует исходному и целевому адресам. В этом случае, если компьютер с IP-адресом 172.31.60.51 отправляет любой трафик на адрес 172.31.70.51, коммутатор должен *отказать* или заблокировать его.

Используя эту информацию, вы готовы создать свой список контроля доступа, который вы примените к интерфейсу позже в этой главе.

9.1.1. Создание списка контроля доступа

Чтобы заблокировать трафик IP–IP, вы создаете так называемый *расширенный список доступа к IP*. Запомните это длинное имя – оно ассоциируется с такой же длинной командой, которую применяют для создания списка и о которой вы узнаете далее. Вы также должны назначить списку доступа имя или номер. Нет никаких технических оснований для использования одного или другого, но мое личное предпочтение – придерживаться чисел.

Практикум

Перейдите в режим глобальной конфигурации. Затем создайте список контроля доступа с именем 100, используя следующую команду:

```
ip access-list extended 100
```

Вы должны увидеть приглашение для расширенного именного списка управления доступом (NACL):

```
Switch1(config-ext-nacl)#
```

Как я уже сказал, ACL – это всего лишь список правил. Вы создали список ACL с именем 100, но вы не создали никаких правил. Используя информацию из табл. 9.1, вам нужно создать новое правило в списке ACL.

Практикум

Убедитесь, что вы все еще видите приглашение (config-ext-nacl)#, и выполните следующую команду, чтобы запретить трафик от компьютера с IP-адресом 172.31.60.51 к компьютеру 172.31.70.51:

```
Switch1(config-ext-nacl)#deny ip host 172.31.60.51 host 172.31.70.51
```

Кстати, сотрудники компании Cisco называют это правило записью контроля доступа (*access control entry – ACE*), но я предпочитаю термин *правило*, которое точнее отражает смысл.

Формат правила в списках ACL довольно прост, это можно увидеть на рис. 9.2. Действие указывается первым делом – в этом случае отказ (deny). Затем значение IP указывает, что правило применимо только к IP-трафику. Следующий параметр – host 172.31.60.51 – указывает, что правило применяется к трафику

с IP-адреса одного *источника* с IP-адресом 172.31.60.51. Наконец, значение host 172.31.70.51 указывает, что правило применяется, если данные *предназначаются* IP-адресу 172.31.70.51.

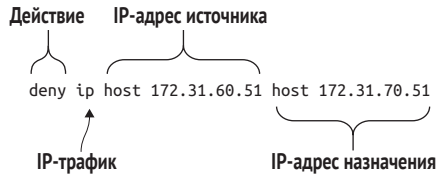

Рис. 9.2 ❖ Разбор простого правила из списка ACL

Несмотря на то что *исходный* адрес и адрес *назначения* не обозначаются явно в команде, система IOS интерпретирует их в правильном порядке.

Имейте в виду, что при создании правила IOS не проверяет, действительны ли указанные IP-адреса. При создании списка доступа всегда дважды проверяйте IP-адреса самостоятельно.

Практикум

Выйдите из режима конфигурирования списка ACL:

```
Switch1(config-ext-nacl)#exit
```

Проверьте список ACL и наличие нового правила:

```
Switch1#show access-lists 100
```

Вы должны увидеть следующее:

```
Switch1#sh access-lists 100
Extended IP access list 100
    10 deny ip host 172.31.60.51 host 172.31.70.51 number 10 to the beginning
```

Система IOS добавила правило

Важно отметить, что сам по себе список ACL ничего не делает. Как следует из названия, это список правил. На данный момент Коммутатор 1 не блокирует какой-либо трафик, потому что вы явно не указали Коммутатору 1 использовать список контроля доступа 100. Позже в этой главе я покажу вам, как настроить Коммутатор 1 для использования созданного списка ACL.

Обратите внимание, что перед созданным вами правилом присутствует число 10. Когда вы создаете правило, система IOS назначает ему *порядковый номер* с шагом 10. Первое правило имеет номер 10, второе – 20 и т. д. Номер называется порядковым, потому что когда система IOS сканирует список ACL, чтобы определить, что делать с данным пакетом, то применяет каждое правило по порядку, начиная с меньшего порядкового номера. Чтобы понять, как это работает, давайте рассмотрим два примера.

Первый пример может вызвать недоумение, но он иллюстрирует важный момент. Если система IOS видит пакет с исходным IP-адресом 172.31.60.51 и IP-

адресом назначения 172.31.70.51, она сначала смотрит на правило 10. Поскольку исходные и целевые IP-адреса соответствуют заданным в правиле, система IOS выполняет действие, требуемое правилом, т. е. отказывает. Система IOS отбрасывает пакет, и дело с концом.

Теперь давайте рассмотрим обратный пример. Предположим, что система IOS видит пакет с исходным IP-адресом 172.31.60.51 и IP-адресом назначения 192.168.100.1. Сначала система IOS рассматривает правило с наименьшим номером, т. е. 10. Исходный IP-адрес – 172.31.60.51 – соответствует указанному в правиле. Но IP-адрес назначения – 192.168.100.1 – нет. Поскольку пакет точно не соответствует правилу, система IOS не применяет действие deny, указанное в правиле 10. Но что же происходит с пакетом?

Вы можете предположить, что система IOS разрешает передачу пакета, и это разумное предположение. В конце концов, задачей списка ACL является блокировка пакетов, и единственное правило, указанное в правиле 10 списка ACL, не запрещает передачу данного пакета. Но есть еще одно правило – скрытое, – запрещающее передачу пакета.

Правило неявного запрета

Каждый список ACL после создания содержит секретное скрытое правило, называемое *правилом неявного запрета*. Правило выглядит так:

```
deny ip any any
```

Вы можете обвинить компанию Cisco в том, что ее оборудование не интуитивно понятно во множестве вещей, но эта команда делает именно то, что она значит. Она запрещает трафик с любого IP-адреса на любой IP-адрес. Другими словами, правило неявного запрета блокирует весь IP-трафик.

Система IOS гарантирует, что правило скрытого запрета указано в нижней части всех списков ACL. Независимо от того, сколько правил вы явно определяете в списке ACL, правило неявного запрета всегда будет указано в нижней части списка. Правило скрыто, поэтому вы не можете его увидеть, но оно есть! В табл. 9.2 показано, как система IOS обрабатывает список контроля доступа 100.

Таблица 9.2. Список контроля доступа 100 с правилом неявного запрета

Последовательность	Действие	Источник	Назначение
10	Deny	172.31.60.51 (Кадры-ПК1)	172.31.70.51 (Начальство-ПК1)
Последняя (скрыта)	Deny	Все	Все

Последовательность 10 соответствует только что настроенному правилу. Но сразу после него есть правило – последнее в последовательности – правило неявного запрета, которое система IOS вставляет автоматически.

Ваша цель – не допустить, чтобы компьютер Кадры-ПК1 связался с компьютером Начальство-ПК1, и список контроля доступа 100 выполняет это, но чрезвычайно тщательно. Правило неявного запрета приводит к тому, что список ACL запрещает весь IP-трафик, включая тот, который вы хотите разрешить.

Очевидным решением было бы избавиться от правила неявного запрета, но вы не можете этого сделать. Вместо этого вам нужно обойти его, создав другое правило, *разрешающее* весь трафик.

Практикум

Войдите в режим конфигурирования списка контроля доступа 100:

```
Switch1(config)#ip access-list extended 100
```

Создайте еще одно правило, разрешающее весь IP-трафик:

```
Switch1(config-ext-nacl)# permit ip any any
```

Система IOS автоматически поместит это правило в нижнюю часть списка. Выйдите из режима конфигурирования списка ACL:

```
Switch1(config-ext-nacl)#exit
```

Проверьте, что новое правило появилось:

```
Switch1#sh access-lists 100
```

Вы должны увидеть новое правило под номером 20:

```
Switch1#sh access-l 100
Extended IP access list 100
    10 deny ip host 172.31.60.51 host 172.31.70.51
    20 permit ip any any
```

◄— Явное разрешение всего IP-трафика замещает правило неявного запрета

Возвращаясь к более раннему примеру, если система IOS обнаружит пакет с исходным IP-адресом 172.31.60.51 и IP-адресом назначения 192.168.100.1, она сначала сравнит пакет с правилом в последовательности 10. Пакет не соответствует правилу, поэтому система IOS продолжает работу по списку. Правило в последовательности 20 – permit ip any any – соответствует пакету, поэтому система IOS разрешает его согласно указанному действию (permit).

Так вы можете эффективно обойти правило неявного запрета, поскольку как только система IOS предпринимает определенные действия по разрешению пакета или отказу, она прекращает обработку последующих правил.

Теперь самое время указать, что в сильно ограниченных средах типа военных объектов вам может потребоваться, чтобы список ACL блокировал весь IP-трафик по умолчанию. Если это так, вы не должны помещать правило permit ip any any в список ACL. Вместо этого вам нужно явно разрешить *только* желаемый трафик, а затем позволить правилу неявного запрета отказать в передаче всего прочего трафика. Кроме того, вы можете настроить явный запрет – указать строку deny ip any any в нижней части списка ACL.

В нашем случае список ACL функционирует точно так, как вы этого хотите. Он запрещает трафик от компьютера Кадры-ПК1 к компьютеру Начальство-ПК1 и допускает весь прочий трафик. Но, как я упоминал ранее, список ACL сам по себе ничего не делает. Чтобы настроить Коммутатор 1 для работы со

списком ACL и начать блокировку трафика, вам нужно применить список ACL к интерфейсу.

9.2. Применение списка контроля доступа к интерфейсу

Как правило, нужно применять список ACL к интерфейсу, ближайшему к источнику трафика, который вы хотите заблокировать. В этом случае список контроля доступа `100` блокирует трафик от компьютера Кадры-ПК1 к компьютеру Начальство-ПК1, поэтому наилучшим местом применения списка ACL будет порт, к которому подключен компьютер Кадры-ПК1, – FastEthernet0/20. Чтобы понять логику, взгляните на рис. 9.3.

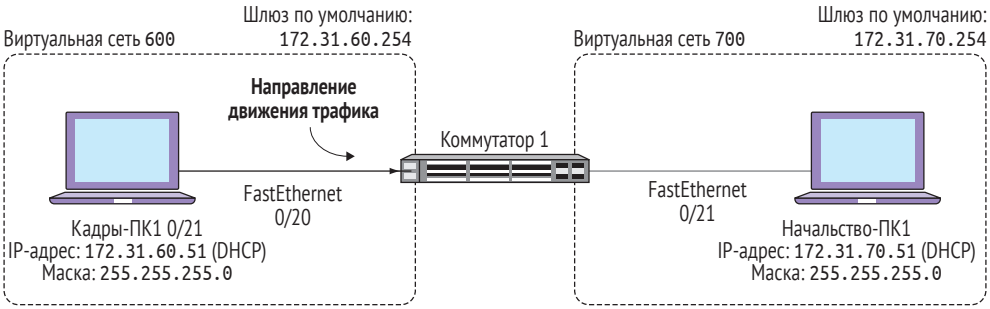

Рис. 9.3 ❖ Определение интерфейса для применения списка контроля доступа `100`. Порт FastEthernet0/20 – интерфейс, ближайший к источнику трафика

Исходный трафик, идущий от компьютера Кадры-ПК1, *поступает* на порт FastEthernet0/20. Если вы хотите, чтобы Коммутатор 1 блокировал любой трафик, исходящий с компьютера Кадры-ПК1, лучше всего выбрать интерфейс, на который поступает этот трафик. Следовательно, вам нужно применить список ACL к интерфейсу FastEthernet0/20.

Практикум

Примените список контроля доступа `100` к порту FastEthernet0/20, используя указанные ниже команды.
Войдите в режим конфигурирования интерфейса FastEthernet0/20:

```
Switch1(config)#interface fastEthernet 0/20
```

Примените список контроля доступа `100` к интерфейсу:

```
Switch1(config-if)#ip access-group 100 in
```

Значение `ip access-group 100` в команде озадачивает многих новичков. Несмотря на то что в команде ничего не говорится о «списке доступа», фактиче-

ски команда ссылается на список контроля доступа 100. Как я уже говорил ранее (и, скорее всего, еще буду), в компании Cisco любят использовать несколько терминов (группы и списки доступа) для обозначения одного и того же.

Ключевое слово in в конце команды означает, что система IOS должна применять правила ACL только для входящего трафика, то есть трафика, поступающего в порт FastEthernet0/20. Это не должно касаться трафика, исходящего из порта FastEthernet0/20.

Практикум

Проверьте, чтобы список контроля доступа 100 был применен к порту FastEthernet0/20:

```
Switch1#show ip interface fastEthernet 0/20
```

Вы должны увидеть следующее:

```
Switch1#sh ip interface fastEthernet 0/20
FastEthernet0/20 is up, line protocol is up
  Inbound access list is 100
```

Если вы еще не убедились, что команда access-group действительно относится к списку доступа, последняя строка в выводе должна развеять любые сомнения. На этом этапе система IOS проверяет весь входящий в порт FastEthernet0/20 IP-трафик по списку контроля доступа 100.

Рекомендуется всегда проверять конфигурации несколькими способами, если это возможно. На данный момент вы научились проверять, что список ACL существует, содержит корректные правила и применяется к правильному интерфейсу. Но лучший способ проверить, что все эти моменты соблюдаются, – это практика. Попробуйте пропинговать компьютер Начальство-ПК1 с компьютера Кадры-ПК1. Если вы все сделали правильно, команда ping должна потерпеть неудачу.

Практикум

Используйте команду ping, чтобы проверить, что компьютер Кадры-ПК1 (172.31.60.51) не может получить доступ к компьютеру Начальство-ПК1 (172.31.70.51). Если ваши тестовые компьютеры имеют те же IP-адреса, что и мои, выполните следующую команду на компьютере Кадры-ПК1:

```
ping 172.31.70.51
```

Вывод на компьютере Кадры-ПК1 должен быть следующим:

```
C:\>ping 172.31.70.51
Pinging 172.31.70.51 with 32 bytes of data:
Request timed out.
```

Запомните, что целью является запрещение доступа с компьютера Кадры-ПК1 к компьютеру Начальство-ПК1 без ограничения прочего трафика с компьютера Кадры-ПК1. Постоянная неудача пинга указывает, что список ACL блокирует трафик с компьютера Кадры-ПК1 на компьютер Начальство-ПК1. Но чтобы подтвердить, что Коммутатор 1 разрешает любой другой трафик с компьютера Кадры-ПК1, попробуем выполнить пинг другого IP-адреса.

Практикум

С компьютера Кадры-ПК1 пропингуйте SVI-интерфейс виртуальной сети `700` (`172.31.70.254`):

```
ping 172.31.70.254
```

Вы должны увидеть следующее:

```
C:\>ping 172.31.70.254

Pinging 172.31.70.254 with 32 bytes of data:
Reply from 172.31.70.254: bytes=32 time=4ms TTL=255
```

Отлично! Коммутатор 1 блокирует доступ с компьютера Кадры-ПК1 к компьютеру Начальство-ПК1, не мешая другому IP-трафику с компьютера Кадры-ПК1.

9.3. Блокировка трафика «IP-подсеть»

Как и должно быть, список ACL `100`, который вы применяли для входящего трафика на порту FastEthernet0/20, блокирует только трафик от компьютера Кадры-ПК1 к компьютеру Начальство-ПК1. Теперь предположим, что вы хотите заблокировать трафик от компьютера Кадры-ПК1 на все устройства в виртуальной сети `Executives`. Самый простой и безопасный способ достичь этого – создать новый список ACL с правилами, перечисленными в табл. 9.3, и применить его к порту FastEthernet0/20.

Таблица 9.3. Правила ACL для блокирования IP-трафика к целой подсети

Последовательность	Действие	Источник	Назначение
`10`	`Deny`	`172.31.60.51` (Кадры-ПК1)	`172.31.70.0/255.255.255.0` (виртуальная сеть `Executives` (`700`))
`20`	`Permit`	Все	Все

Виртуальная сеть `Executives` соответствует подсети `172.31.70.0`. Обратите внимание, что для последовательности `10` пункт назначения содержит как подсеть (`172.31.70.0`), так и маску подсети (`255.255.255.0`). Напомню, что маска подсети – это код, который сообщает IP-адреса, являющиеся частью виртуальной сети. При указании всей подсети в качестве пункта назначения в списке ACL вам нужно сообщить системе IOS маску подсети. Но есть сложность: вы не можете указывать маску подсети в списках ACL.

Практикум

Так как список контроля доступа `100` уже используется, вам не нужно изменять или удалять его. Вместо этого создайте совершенно новый список ACL с именем `101`:

```
Switch1(config)#ip access-list extended 101
```

Добавьте правило для блокирования доступа к виртуальной сети `Executives` с компьютера Кадры-ПК1:

```
Switch1(config)#deny ip host 172.31.60.51 172.31.70.0 0.0.0.255
```

Последний параметр выглядит немного странным, и я объясню его через минутку. Но сначала вам нужно заместить правило неявного запрета:

```
Switch1(config)#permit ip any any
```

Проверьте список ACL, выполнив команду `show access-lists 101`.

Вы должны увидеть следующее:

```
Switch1#sh access-lists 101
Extended IP access list 101
    10 deny ip host 172.31.60.51 172.31.70.0 0.0.0.255
    20 permit ip any any
```

9.3.1. Подстановочные маски

Часть команды, касающаяся пункта назначения, существенно отличается от той, которую вы выполняли ранее. Последние два параметра, `172.31.70.0` и `0.0.0.255`, указывают на подсеть назначения. Но `0.0.0.255` – недопустимая маска подсети. Фактически это называется подстановочной маской (*wildcard mask*) и представляет собой обращенную маски подсети. В табл. 9.4 показано, что я имею в виду.

Таблица 9.4. Преобразование маски подсети в подстановочную маску

Маска подсети	Подстановочная маска
255.0.0.0	0.255.255.255
255.255.0.0	0.0.255.255
255.255.255.0	0.0.0.255
255.255.255.255	0.0.0.0

Я не собираюсь вдаваться в подробности, почему на оборудовании Cisco в списках ACL используются подстановочные маски вместо масок подсетей. Это связано с внушительным объемом вычислений, которые вы освоите, если решите сдать экзамен на получение сертификата Cisco. Сейчас же вам нужно знать, что при написании правила ACL вместо маски подсети вам нужно указать соответствующую обращенную маску. Сделайте закладку на этой странице, чтобы вернуться к данной таблице при необходимости.

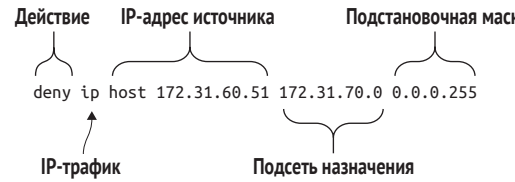

Рис. 9.4 ❖ Анализ правила ACL, запрещающего IP-трафик для всей подсети

Важно понимать, как работает команда, показанная на рис. 9.4. Обратите внимание, что перед адресом целевой подсети и подстановочной маской нет слова `host`. Это потому, что пункт назначения – не хост, а подсеть.

9.3.2. Замена списка ACL

Теперь, когда вы настроили список контроля доступа `101`, пришло время заменить им список `100`. Напоминаю, что вам необходимо применить список `101` к порту FastEthernet0/20.

Практикум

Используйте следующие команды, чтобы применить список контроля доступа `101` к порту FastEthernet0/20:

```
Switch1(config)#int fa0/20
Switch1(config-if)#ip access-group 101 in
```

Если вы все настроили правильно, у компьютера Кадры-ПК1 теперь нет доступа к устройствам в подсети `172.31.70.0`. Давайте проверим.

Практикум

Попробуйте пропинговать компьютер Начальство-ПК1 с компьютера Кадры-ПК1:

```
ping 172.31.70.51
```

В выводе должна отобразиться только ошибка:

```
C:\>ping 172.31.70.51

Pinging 172.31.70.51 with 32 bytes of data:
Request timed out.
```

Теперь время настоящего теста! У вас не должно быть возможности пропинговать любой IP-адрес в подсети `172.31.70.0`, включая SVI-интерфейс виртуальной сети `700` (`172.31.70.254`).

Практикум

Попробуйте пропинговать SVI-интерфейс виртуальной сети 700:

```
ping 172.31.70.254
```

Этот пинг также должен быть неудачным:

```
C:\>ping 172.31.70.254

Pinging 172.31.70.254 with 32 bytes of data:
Request timed out.
```

Отлично! Список ACL работает точно так, как вы его настроили. Но есть потенциальная проблема. Вы применили список контроля доступа 101 к порту FastEthernet0/20, к которому подключен компьютер Кадры-ПК1. Предположим, что кто-то (не вы, конечно) подключит кабели компьютера Кадры-ПК1 к другому порту, скажем FastEthernet0/19. Давайте взглянем на конфигурацию этого порта, чтобы выяснить, что может произойти:

```
Switch1#sh run int fa0/19

interface FastEthernet0/19
 switchport access vlan 600
 switchport mode access
end
```

Обратите внимание на два момента. Во-первых, этот порт является членом виртуальной сети 600, HR, поэтому компьютер Кадры-ПК1 сохраняет сетевое подключение. Но к этому интерфейсу список ACL не применяется, поэтому компьютер Кадры-ПК1 все равно может получить доступ к устройствам в виртуальной сети Executives! Команда show ip interface fa0/19 подтверждает это:

```
Switch1#show ip interface fa0/19
FastEthernet0/19 is up, line protocol is up
  Inbound  access list is not set
```

Практикум

Подключите компьютер Кадры-ПК1 к другому порту в виртуальной сети 600, а затем попробуйте пропинговать компьютер Начальство-ПК1 снова:

```
ping 172.31.70.51
```

Как вы видите, у компьютера Кадры-ПК1 теперь неограниченный доступ:

```
C:\> ping 172.31.70.51

Pinging 172.31.70.51 with 32 bytes of data:
Reply from 172.31.70.51: bytes=32 time=1ms TTL=127
```

Применение списка ACL на уровне порта работает нормально, пока устройство, с которого вы блокируете трафик, не будет подключено к другому пор-

ту. Но если это произойдет, использование списка ACL становится спорным моментом. Вы можете применить список ACL к каждому порту в виртуальной сети, но это довольно сложно. К счастью, у вас есть лучший вариант.

9.3.3. Применение списка управления доступом к коммутируемому виртуальному интерфейсу

Вместо того чтобы применять список контроля доступа `101` к каждому порту, относящемуся к виртуальной сети `600`, вы можете применить его к виртуальному интерфейсу этой сети.

Вам нужно заблокировать доступ с компьютера Кадры-ПК1 к подсети `Executives` независимо от того, к какому порту он подключен. Вы можете сделать это, применив список ACL непосредственно к SVI-интерфейсу виртуальной сети `600`.

Практикум

Войдите в режим конфигурирования SVI-интерфейса виртуальной сети `600`:

```
Switch1(config)#int vlan600
Switch1(config-if)#ip access-group 101 in
```

Проверьте конфигурацию, выполнив следующую команду:

```
show ip interface vlan 600 | i 600|101
```

Вы должны получить такой вывод:

```
Switch1#show ip interface vlan 600 | i 600|101
Vlan600 is up, line protocol is up
  Inbound  access list is 101
```

Поскольку вы применили список контроля доступа `101` к SVI-интерфейсу виртуальной сети `600`, система IOS применяет правила из этого списка ко всему трафику, поступающему на SVI-интерфейс сети `600`, независимо от исходящего физического порта. Это означает, что даже если компьютер Кадры-ПК1 подключен к порту без списка ACL, любой трафик от компьютера Кадры-ПК1, предназначенный сети `Executives`, будет заблокирован.

Практикум

С компьютера Кадры-ПК1 пропингуйте компьютер Начальство-ПК1:

```
ping 172.31.70.51
```

Пинг должен окончиться неудачей:

```
C:\>ping 172.31.70.51

Pinging 172.31.70.51 with 32 bytes of data:
Reply from 172.31.60.254: Destination net unreachable.
```

Пинг SVI-интерфейса виртуальной сети 700 должен иметь тот же результат:

Практикум

С компьютера Кадры-ПК1 пропингуйте SVI-интерфейс виртуальной сети 700:

```
ping 172.31.70.254
```

Вы должны увидеть следующее:

```
C:\>ping 172.31.70.254

Pinging 172.31.70.254 with 32 bytes of data:
Reply from 172.31.60.254: Destination net unreachable.
```

На данный момент конфигурация ACL довольно надежна. Независимо от того, к какому физическому порту коммутатора подключен компьютер Кадры-ПК1, данные с него не могут достичь любого устройства в виртуальной сети Executives. Но есть еще одна проблема. Поскольку компьютер Кадры-ПК1 получает свой IP-адрес с помощью DHCP-сервера, его IP-адрес может измениться, и список ACL станет бесполезным. Чтобы предусмотреть эту проблему, вам необходимо сделать список ACL полностью совместимым с DHCP-сервером.

9.4. Блокирование трафика «подсеть–подсеть»

При настройке списков ACL легко упустить какие-нибудь детали. Но, будучи сетевым администратором, вы всегда должны стремиться к некой цели, которую поставили изначально. В начале главы я изложил простое требование: не позволить компьютеру Кадры-ПК1 добраться до компьютера Начальство-ПК1. Позднее я изменил это требование, чтобы запретить доступ компьютера Кадры-ПК1 к любому устройству в виртуальной сети Executives. Возможно, вы уже поняли, что будет дальше!

Учитывая, что устройства в виртуальных локальных сетях HR и Executives получают свои IP-адреса от DHCP-сервера, ясно, что правила ACL для блокировки трафика на/с отдельных IP-адресов не смогут его предотвратить. Остается только один вариант: не допустить, чтобы весь IP-трафик из виртуальной сети 600 поступал в виртуальную сеть 700. Пришло время создать еще один список ACL!

Практикум

Создайте список контроля доступа 102, чтобы заблокировать весь трафик из подсети 172.31.60.0, направленный в подсеть 172.31.70.0:

```
Switch1(config)#ip access-list extended 102
Switch1(config-ext-nacl)#deny ip 172.31.60.0 0.0.0.255 172.31.70.0
0.0.0.255
Switch1(config-ext-nacl)#permit ip any any
```

Примените список ACL к SVI-интерфейсу виртуальной сети 600:

```
Switch1(config)#int vlan 600
Switch1(config-if)#ip access-group 102 in
```

Теперь любые устройства в подсети 172.31.60.0 не смогут подключаться к устройствам в подсети 172.31.70.0. Давайте поэкспериментируем, пропинговав пару IP-адресов.

Практикум

С компьютера Кадры-ПК1 пропингуйте компьютер Начальство-ПК1 и SVI-интерфейс виртуальной сети 700:

```
ping 172.31.70.51
ping 172.31.70.254
```

Оба пинга должны закончиться неудачей:

```
C:\>ping 172.31.70.51

Pinging 172.31.70.51 with 32 bytes of data:
Reply from 172.31.60.254: Destination net unreachable.

C:\>ping 172.31.70.254

Pinging 172.31.70.254 with 32 bytes of data:
Reply from 172.31.60.254: Destination net unreachable.
```

На первый взгляд, неплохо, но неубедительно, потому что это точно такие же результаты, что были получены раньше. Теперь попробуем изменить IP-адрес компьютера Кадры-ПК1 и вновь пропинговать адреса.

Практикум

Измените IP-адрес компьютера Кадры-ПК1 на статический вне диапазона, настроенного DHCP-сервером.
Установите шлюз по умолчанию – 172.31.60.254.

В предыдущей главе вы настроили диапазон адресов, выделяемых DHCP-сервером, для виртуальной сети 600 в пределах 172.31.60.50–172.31.60.250. На рис. 9.5 показано, как настроить на компьютере Кадры-ПК1 статический IP-адрес 172.31.60.49, который находится за пределами области адресов DHCP. Не забудьте указать шлюз по умолчанию!

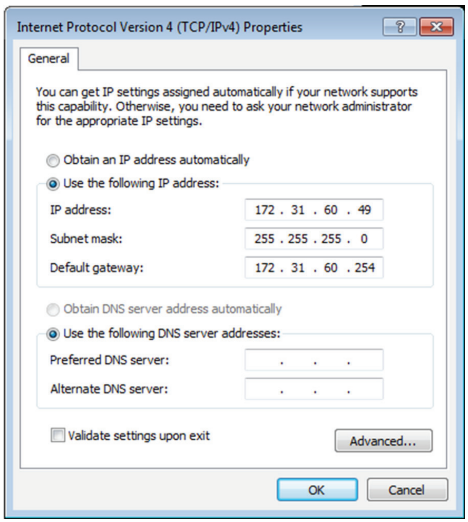

Рис. 9.5 ❖ Настройка на компьютере Кадры-ПК1 статического адреса и шлюза по умолчанию

Практикум

С компьютера Кадры-ПК1 снова пропингуйте адреса `172.31.70.51` и `172.31.70.254`.

Вы должны получить те же результаты, что и до смены IP-адреса:

```
C:\>ping 172.31.70.51

Pinging 172.31.70.51 with 32 bytes of data:
Reply from 172.31.60.254: Destination net unreachable.
C:\>ping 172.31.70.254

Pinging 172.31.70.254 with 32 bytes of data:
Reply from 172.31.60.254: Destination net unreachable.
```

Как видно, Коммутатор 1 предотвращает доступ любого устройства из подсети `172.31.60.0` к любому устройству в подсети `172.31.70.0`. Но вы можете подумать: «Как быть, если я не хочу этого делать? Что делать, если я просто хочу ограничить несколько IP-адресов тут и там?» Предположим, вы хотите, чтобы компьютер Кадры-ПК1 мог получить доступ только к интерфейсу `172.31.70.254` и ни к какому другому в этой подсети. Вы можете достичь этого с помощью списка ACL с одним незначительным изменением.

Практикум

Создайте новый список контроля доступа с именем 103:

```
Switch1(config)#ip access-list extended 103
```

Разрешите любому устройству из виртуальной сети HR связь только с SVI- интерфейсом виртуальной сети 700 по адресу 172.31.70.254,:

```
Switch1(config-ext-nacl)#permit ip 172.31.60.0 0.0.0.255 host
172.31.70.254
```

Запретите доступ с любых устройств в виртуальной сети HR к любым устройствам в подсети 172.31.70.0:

```
deny ip 172.31.60.0 0.0.0.255 172.31.70.0 0.0.0.255
```

Переопределите скрытое правило неявного запрета:

```
permit ip any any
```

Проверьте созданный список ACL командой show access-list 103.

Вы должны увидеть следующее:

```
Switch1#show access-lists 103
Extended IP access list 103
    10 permit ip 172.31.60.0 0.0.0.255 host 172.31.70.254
    20 deny ip 172.31.60.0 0.0.0.255 172.31.70.0 0.0.0.255
    30 permit ip any any
```

Теперь все, что вам нужно сделать, – это применить новый список ACL к SVI-интерфейсу виртуальной сети 600.

Практикум

Примените список контроля доступа 103 к SVI-интерфейсу виртуальной сети 600:

```
Switch1(config)#int vlan 600
Switch1(config-if)#ip access-group 103 in
```

Проверьте, что компьютер Кадры-ПК1 может пинговать исключительно адрес 172.31.70.254 в подсети Executives:

```
ping 172.31.70.254
ping 172.31.70.51
```

Компьютер Кадры-ПК1 может без проблем пинговать SVI-интерфейс виртуальной сети 700:

```
C:\>ping 172.31.70.254

Pinging 172.31.70.254 with 32 bytes of data:
Reply from 172.31.70.254: bytes=32 time=2ms TTL=255
```

Но не может пинговать компьютер Начальство-ПК1:

```
C:\>ping 172.31.70.51

Pinging 172.31.70.51 with 32 bytes of data:
Reply from 172.31.60.254: Destination net unreachable.
```

Цель достигнута.

9.5. Команды, использованные в этой главе

Порядок операций в списках контроля доступа нельзя путать, т. к. можно превратить работающий список ACL в совершенно бесполезный. См. табл. 9.5 при выполнении практического задания. Чтобы упростить задачу, я перечислил команды в том порядке, в котором вы будете их использовать.

Таблица 9.5. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`ip access-list extended 150`	Глобальный	Создает список контроля доступа `150` и входит в режим конфигурирования расширенного списка доступа IP
`deny ip host 1.2.3.4 host 5.6.7.8`	Расширенный список доступа IP	Запрещает IP-трафик с адреса `1.2.3.4` на адрес `5.6.7.8`
`permit ip 172.31.10.0 0.0.0.255 host 7.7.7.7`	Расширенный список доступа IP	Разрешает любой IP-трафик из подсети `172.31.10.0/255.255.255.0` к адресу `7.7.7.7`
`permit ip any any`	Расширенный список доступа IP	Разрешает любой IP-трафик
`ip access-group 150 in`	Интерфейс	Применяет список контроля доступа `150` к выбранному интерфейсу

9.6. Практическое задание

В этой главе вы использовали список ACL для управления IP-трафиком из виртуальной сети 600 к виртуальной сети `700`. В практическом задании вам нужно настроить список ACL для ограничения трафика в обратном направлении, из виртуальной сети `700` в виртуальную сеть `600`.

Выполните следующие задания:

1. Настройте список ACL для запрета доступа с компьютера Начальство-ПК1 (`172.31.70.51`) к любому устройству в подсети `172.31.60.0/255.255.255.0`.
2. Примените этот список ACL к соответствующему интерфейсу.
3. Настройте другой список ACL для запрета доступа всех устройств в подсети `172.31.70.0/255.255.255.0` к любому устройству в подсети `172.31.60.0/255.255.255.0`.
4. Замените список ACL, созданный на шаге 1, на список ACL, созданный на шаге 3.

Глава 10

Подключение коммутаторов с использованием транков

Коммутатор, который я описываю в этой книге, оборудован только 24 портами. У вашего устройства может быть 24 или 48 портов, но независимо от количества в организации любого размера, как вы можете себе представить, одного коммутатора будет недостаточно. По мере добавления в сеть новых устройств в конечном итоге вам не хватит портов коммутатора. Когда это произойдет, вам придется внедрить еще один коммутатор.

В этой главе вы подключите Коммутатор 2 к Коммутатору 1. Затем вы переключите компьютер Начальство-ПК1 с Коммутатора 1 на Коммутатор 2. Прежде чем приступать к этой главе, убедитесь, что вы настроили Коммутатор 2 согласно инструкциям, приведенным по ссылке **Source Code** на странице www.manning.com/books/learn-cisco-networkadministration-in-a-month-of-lunches. Если вы этого не сделали, то не сможете выполнить практических заданий из этой главы.

Недостаточно просто подключить второй коммутатор к существующему коммутатору и включить на нем питание. Это *почти* так же просто, но вам понадобится выполнить еще несколько ключевых шагов, чтобы сеть заработала:

1) физически подключите новый коммутатор;
2) настройте порты коммутатора для формирования транка виртуальной сети;
3) настройте виртуальные сети на новом коммутаторе.

На втором шаге появляется новый термин, *транк виртуальной сети*. Я расскажу подробнее о транках позже, но на данный момент запомните, что транк виртуальной сети – это особый тип логического соединения, через который проходит трафик для нескольких виртуальных сетей. В отличие от порта доступа, который передает трафик только для одной виртуальной сети, транковый порт одновременно передает трафик нескольким виртуальным сетям. Скорее всего, у вас уже есть транки виртуальной сети, работающие в вашей организации.

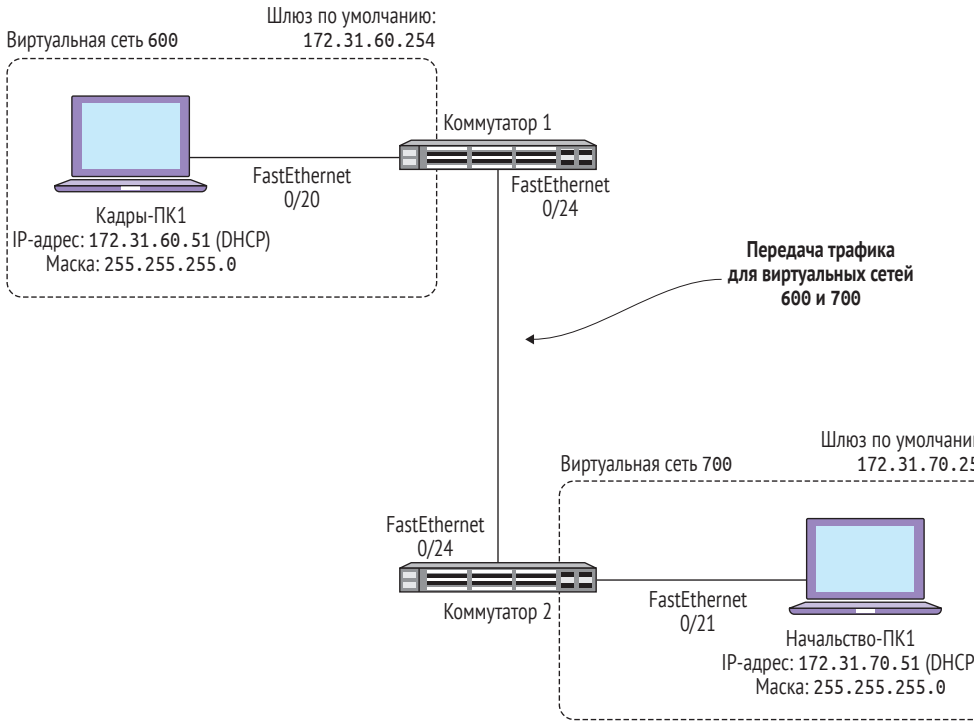

Рис. 10.1 ❖ Соединение Коммутатора 2 с Коммутатором 1 через транк виртуальной сети и переподключение компьютера Начальство-ПК1 к Коммутатору 2

Взгляните на рис. 10.1, чтобы получить четкое представление о том, что вы построите в этой главе. Обратите внимание на добавление Коммутатора 2 и миграцию компьютера Начальство-ПК1 с Коммутатора 1 на Коммутатор 2. Также обратите внимание, что транк между коммутаторами передает трафик для виртуальных сетей `600` и `700`.

10.1. Подключение дополнительного коммутатора

Еще раз, если вы еще не настроили Коммутатор 2 в соответствии с инструкциями на сайте книги, прервите чтение и сделайте это сейчас.

Важно отметить, что существует несколько способов физического подключения коммутаторов Cisco. Некоторые модели поддерживают Cisco StackWise – проприетарную технологию Cisco, которая позволяет подключать коммутаторы с использованием мощных (и дорогостоящих) кабелей StackWise. Но только некоторые модели поддерживают StackWise, и коммутатор Catalyst 3560, рекомендованный в этой книге, к ним не относится. Вот почему в этой главе вы подключите свои коммутаторы, используя простой Ethernet-кабель.

Даже если ваши коммутаторы поддерживают StackWise, вы все равно можете следовать инструкциям, изложенным в этой главе.

Практикум

Авторизуйтесь на Коммутаторе 1.
Возьмите кабель Ethernet и соедините порт FastEthernet0/24 Коммутатора 1 с портом FastEthernet0/24 Коммутатора 2.

На Коммутаторе 1 должна появиться связь:

```
*Mar  1 05:23:11.869: %LINK-3-UPDOWN: Interface FastEthernet0/24, changed
     state to up
*Mar  1 05:23:12.875: %LINEPROTO-5-UPDOWN: Line protocol on Interface
     FastEthernet0/24, changed state to up
```

Если связи нет, убедитесь, что ваш кабель работоспособен и что Коммутатор 2 настроен в соответствии с инструкциями на сайте книги. Если связь появилась, можно перейти к следующему шагу.

Дополнительно

Обычный прямой Ethernet-кабель прекрасно подойдет для подключения большинства коммутаторов Cisco, которыми вы, вероятно, будете пользоваться. Но если вы используете старую модель, которая не поддерживает функцию автосогласования (например, Catalyst 3550), есть вероятность, что прямой кабель не будет работать. Если это так, вам может потребоваться найти скрестный кабель для подключения коммутаторов.

10.2. Принципы транков виртуальной сети

Некоторое время назад я сказал, что *транк виртуальной сети* – это специальное соединение, которое передает трафик для нескольких виртуальных сетей. Также его называют *транковым соединением виртуальной сети*, или просто *транком*. Все эти термины означают одно и то же. Идея транка заключается в том, что вы можете использовать одно физическое соединение между коммутаторами для передачи трафика нескольких виртуальных сетей, не беспокоясь о перемещении трафика *между* виртуальными сетями. Рисунок 10.2 иллюстрирует транк виртуальной сети, который вы будете настраивать в этой главе.

Возможно, вам интересно, почему между коммутаторами 1 и 2 существует две связи, хотя используется только одно физическое соединение. Эти две линии представляют собой два отдельных *логических* соединения, по одному для каждой виртуальной сети. Несмотря на то что коммутаторы используют одно физическое соединение, внутри они разделяют трафик для каждой виртуальной сети. Позже в этой главе я немного углублюсь в технические подробности того, как они это делают.

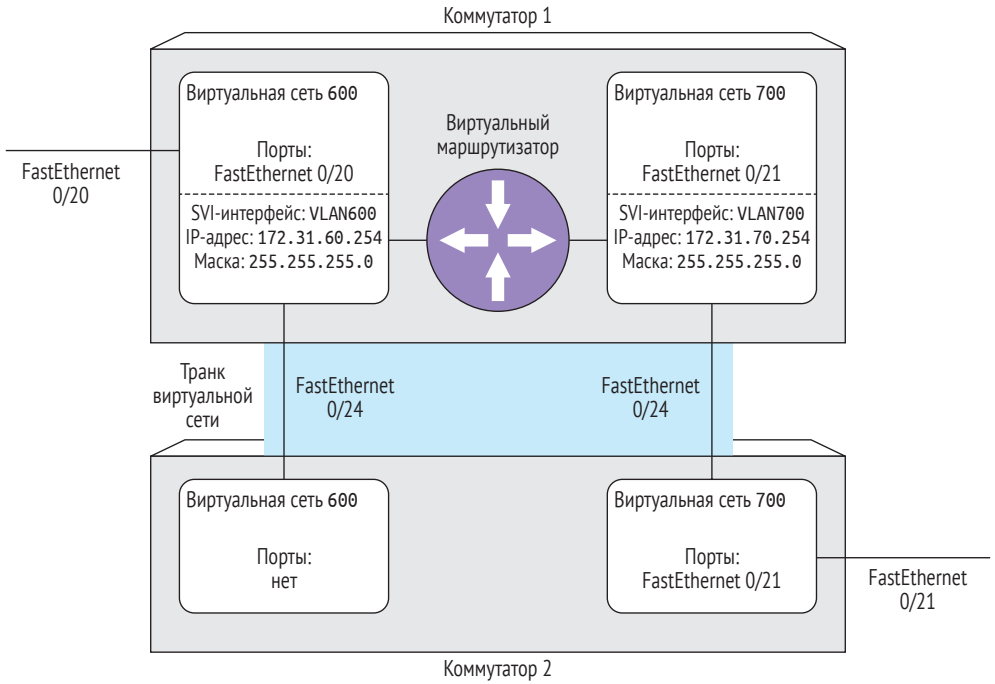

Рис. 10.2 ❖ «Вид сверху» на транк между Коммутаторами 1 и 2

10.2.1. Настройка транка виртуальной сети

Некоторое время назад компания Cisco существенно упростила процесс создания транка. В коммутаторах старых моделей все, что вам нужно было сделать, – это соединить их вместе, и они волшебным образом сформируют транк с помощью процесса, называемый *согласованием*. Однако в новых моделях это уже не так, и вам нужно выполнить небольшую ручную настройку, чтобы создать транк.

Протокол DTP

Протокол динамического транкинга (*Dynamic Trunking Protocol, DTP*) – это проприетарная функция Cisco, которая автоматически настраивает интерфейс как порт доступа виртуальной сети или транковый порт. DTP – **опциональная** функция, и вы можете настроить транк без нее, но она включена по умолчанию, и большинство сетей Cisco, с которыми вы столкнетесь, будут использовать этот протокол. Прежде чем вы сможете использовать протокол DTP для настройки транка, вам нужно увидеть, как настроена функция DTP на вашем коммутаторе.

Практикум

Введите следующую команду, чтобы увидеть, как настроен протокол DTP на порту FastEthernet0/24:

```
Switch1#show interfaces fa0/24 switchport
```

Поскольку я не знаю, какую конкретную модель коммутатора или версию системы IOS вы используете, важно, чтобы вы сделали это под себя и не рассчитывали, что моя конфигурация такая же, как ваша. Вы должны увидеть результат, аналогичный следующему:

```
Switch1#show interfaces fa0/24 switchport
Name: Fa0/24
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
[extraneous output truncated]
```

Протокол DTP настроен на создание либо порта доступа, либо транка на основе запросов другой стороны

По умолчанию протокол DTP настраивает интерфейс как порт доступа виртуальной сети

По умолчанию протокол DTP помещает порт в виртуальную сеть 1

Опция `Administrative Mode` имеет значение `dynamic auto` (слово dynamic относится к протоколу DTP). Этот параметр означает, что коммутатор будет ожидать другую сторону (в данном случае Коммутатор 2), чтобы сообщить, должен ли этот порт стать транковым или портом доступа виртуальная сеть.

Опция `Operational Mode` имеет значение `static access`. Напомню, в главе 5 вы использовали команду `switchport mode access`, чтобы настроить порт для доступа в специфическую виртуальную сеть. Технически вы можете не вводить эту команду, т. к. протокол DTP по умолчанию включает на порту этот статический режим и вводит порт в виртуальную сеть 1, о чем сообщает опция `Access Mode VLAN`.

Дополнительно

По-прежнему рекомендуется использовать команду `switchport mode access` при настройке порта доступа. Это устраняет функцию DTP как потенциальную точку отказа, позволяет избежать случайных транков и упрощает чтение конфигурации.

Как указано, интерфейс FastEthernet0/24 на Коммутаторе 1 настроен как порт доступа к виртуальной сети `1`. Это означает, что этот порт не транковый и, следовательно, не может передавать трафик для виртуальной сети `600` или `700`. Пришло время исправить это.

10.2.2. Настройка протокола DTP для автоматического согласования транка

Я только что говорил, что режим `dynamic auto` на порту FastEthernet0 / 24 Коммутатора 1 инструктирует протокол DTP ожидать, пока другая сторона – Комму-

татор 2 – сообщит ей, формировать ли транк. Поскольку коммутаторы еще не сформировали транк, вы можете сделать вывод, что Коммутатор 2, вероятно, также настроен в режиме `dynamic auto`. Другими словами, оба коммутатора ждут друг друга, кто сделает первый ход.

Чтобы протокол DTP согласовал транк виртуальной сети, необходимо сменить режим `dynamic auto` на Коммутаторе 1 на другой, `dynamic desirable`. Эта настройка инструктирует Коммутатор 1 сообщить Коммутатору 2: «Эй, я хочу создать транк!»

Практикум

Введите следующие команды, чтобы настроить порт FastEthernet0/24 на Коммутаторе 1 для согласования транка с Коммутатором 2:

```
Switch1(config)#interface fa0/24
Switch1(config-if)#switchport mode dynamic desirable
```

Интерфейс должен выключиться и включиться снова, примерно так:

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
     changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
     changed state to up
```

Кстати, это иногда называют *нестабильностью*, или *хлопаньем*. Вероятно, вы не получите какого-либо другого вывода, поэтому вам нужно вручную убедиться, что транк сформирован.

Практикум

Проверьте, что Коммутаторы 1 и 2 сформировали транк:

```
Switch1#show interfaces fa0/24 switchport
```

Вы должны увидеть нечто следующее:

```
Name: Fa0/24
Switchport: Enabled
Administrative Mode: dynamic desirable  <-- Протокол DTP настроен на попытку согласования транка
Operational Mode: trunk  <-- Транк виртуальной сети установлен между Коммутаторами 1 и 2
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: isl
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
[extraneous output truncated]
```

Обратите внимание, что административный режим `dynamic desirable` теперь отражает изменение конфигурации, которое вы только что внесли. Опция `Operational Mode` со значением `trunk` указывает, что этот порт теперь является портом транка. Хотя это неплохой способ проверки, в выводе отсутствуют некоторые важные сведения о транке.

Как я уже сказал, транк виртуальной сети может передавать трафик для нескольких виртуальных сетей. Но это *не* обязательно. В данном случае, поскольку вы собираетесь перевести компьютер Начальство-ПК1 на Коммутатор 2, вам необходимо убедиться, что трафик для виртуальной сети `Executives` (`700`) проходит через транк.

Практикум

Выполните команду `show interfaces trunk` на Коммутаторе 1, чтобы увидеть подробные сведения о созданном транке.

Вы должны увидеть следующее:

```
Switch1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/24      desirable        n-isl          trunking      1

Port        Vlans allowed on trunk
Fa0/24      1-4094

Port        Vlans allowed and active in management domain
Fa0/24      1,20,600,700  ◄── Виртуальные сети обмениваются данными через транк

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/24      1,20,600,700
```

Первые две строки указывают, что FastEthernet0/24 является транковым портом, образованным функцией DTP (на что указывает режим `desirable`).

Посмотрите на строку ниже – `Vlans allowed and active in management domain`. Она указывает на то, что Коммутатор 1 *отправляет* трафик для виртуальных сетей `1`, `20`, `600` и `700` через этот транк. Когда у вас есть транк, система IOS по умолчанию взаимодействует со всеми виртуальными сетями по этой магистрали. Опять же, это не означает, что все эти виртуальные сети находятся в одном широковещательном домене. Это лишь значит, что для передачи на Коммутатор 2 все они используют один и тот же физический порт.

Несмотря на то что вы собираетесь перевести компьютер Начальство-ПК1 на Коммутатор 2, нет никакого вреда в том, что все виртуальные сети могут взаимодействовать через транк. Важно то, что трафик виртуальной сети `700` передается с Коммутатора 1 к Коммутатору 2.

10.3. Настройка Коммутатора 2

Прежде чем перевести компьютер Начальство-ПК1 на Коммутатор 2, вам нужно настроить Коммутатор 2. Но прежде чем вы сможете это сделать, вам нужно авторизоваться на нем. Вы уже настроили SVI-интерфейс сети `VLAN1` Коммутатора 2, присвоив ему IP-адрес `192.168.1.102`. Напомню, что виртуальная сеть `1` предустановлена по умолчанию на всех коммутаторах Cisco, и вы не можете

удалить его или отключить. Следовательно, виртуальная сеть 1 всегда будет взаимодействовать через все транковые порты. Идеальный случай для тестирования созданного транка.

Практикум

Протестируйте транк к Коммутатору 2, пропинговав его SVI-интерфейс виртуальной сети 1:

```
Switch1#ping 192.168.1.102
```

Если вы настроили транк правильно, то должны получить ответ:

```
Switch1#ping 192.168.1.102
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.102, timeout is 2 seconds:
!!!!!
```

Если ваши пинги успешны, вы готовы к настройке Коммутатора 2. Вы займетесь этим, организовав Telnet-сеанс с Коммутатора 1. В системе IOS есть встроенный клиент Telnet, что часто очень удобно, особенно при настройке нового коммутатора!

Практикум

Организуйте Telnet-сеанс с Коммутатора 1 на Коммутатор 2:

```
Switch1#telnet 192.168.1.102
```

Авторизуйтесь, используя следующие данные:

```
Username: admin
Password: cisco
```

Вот что вы должны увидеть:

```
Switch1#telnet 192.168.1.102
Trying 192.168.1.102 ... Open

User Access Verification

Username: admin
Password:
Switch2#
```

Обратите внимание, что приглашение сообщает о том, что вы находитесь на Коммутаторе 2. Если вы мне не верите, то можете увидеть себя, выполнив команду `show ip interface vlan 1 | i up|Internet`:

```
Switch2#show ip interface vlan 1 | i up|Internet
Vlan1 is up, line protocol is up
  Internet address is 192.168.1.102/24
```

Тот факт, что вы подключены к этому SVI-интерфейсу, указывает на то, что трафик виртуальной сети 1 успешно проходит через транк. Как бы то ни было, у вас есть транк между Коммутаторами 1 и 2, и трафик виртуальной сети 1 свободно передается по нему. Следующий шаг – перевести компьютер Начальство-ПК1 с Коммутатора 1 на Коммутатор 2.

10.3.1. Настройка виртуальных сетей на дополнительном коммутаторе

Прежде чем физически перевести компьютер Начальство-ПК1 на Коммутатор 2, необходимо убедиться в наличии на нем виртуальной сети 700.

Практикум

На Коммутаторе 2 выполните команду `show vlan brief`, чтобы проверить наличие виртуальной сети 700.

Вы должны увидеть следующее:

```
Switch2#show vlan brief

VLAN Name                Status    Ports
---- ------------------- --------- -------------------------------
1    default             active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                   Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                   Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                   Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                   Fa0/21, Fa0/22, Fa0/23, Gi0/1
                                   Gi0/2
1002 fddi-default        act/unsup
1003 token-ring-default  act/unsup
1004 fddinet-default     act/unsup
1005 trnet-default       act/unsup
```

Виртуальная сеть 700 не существует! Фактически ни одна из виртуальных сетей, настроенных на Коммутаторе 1, – 20, 600 и 700, – не существует на Коммутаторе 2. Изначально это может показаться неважным. Но если вы посмотрите на транк виртуальной сети с точки зрения Коммутатора 2, то увидите большую проблему.

Практикум

На Коммутаторе 2 выполните команду `show interfaces trunk`.

Вы должны увидеть следующее:

```
Switch2#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/24      auto             n-isl          trunking      1
```

```
Port        Vlans allowed on trunk
Fa0/24      1-4094

Port        Vlans allowed and active in management domain
Fa0/24      1

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/24      1
```

Только VLAN1 находится в состоянии allowed and active, и никакие другие виртуальные сети не найдены. Это означает, что Коммутатор 2 может получать трафик лишь для виртуальной сети 1.

Напомню, что по умолчанию коммутаторы Cisco содержат только виртуальную сеть 1. Поскольку Коммутатор 2 был совсем недавно добавлен в сеть, виртуальные сети, настроенные на Коммутаторе 1, – 20, 600 и 700, – еще не настроены на нем. Чтобы Коммутатор 2 мог отправлять и получать трафик для этих виртуальных сетей через транк, вам нужно настроить их вручную.

Практикум

На Коммутаторе 2 настройте виртуальные сети 20, 600 и 700, выполнив следующие команды:

```
Switch2(config)#vlan 20,600,700
Switch2(config-vlan)#exit
```

Выполните еще раз команду show interfaces trunk для проверки.

Вывод команды show interfaces trunk демонстрирует, что настроенные виртуальные сети активны:

```
Switch2#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/24      auto             n-isl          trunking      1

Port        Vlans allowed on trunk
Fa0/24      1-4094

Port        Vlans allowed and active in management domain
Fa0/24      1,20,600,700

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/24      1
```

Суть в том, что если виртуальная сеть не настроена на коммутаторе, он не сможет отправлять или получать трафик для этой виртуальной сети. Каждый раз, когда вы добавляете новый коммутатор в среду, убедитесь, что вы настроили все необходимые виртуальные сети на нем!

10.4. Перемещение устройств на другой коммутатор

Теперь вы готовы перевести компьютер Начальство-ПК1 на Коммутатор 2. Это будет наипростейшая часть всего процесса!

Практикум

Отсоедините кабель компьютера Начальство-ПК1 от порта Коммутатора 1 и подключите его к порту FastEthernet0/21 Коммутатора 2.
Настройте этот интерфейс как порт доступа виртуальной сети 700, выполнив следующие команды:

```
Switch2(config)#interface fa0/21
Switch2(config-if)#switchport access vlan 700
```

Проверьте конфигурацию, выполнив команду `show interfaces fa0/21 switchport | i Mode`.

Вы должны увидеть следующий вывод:

```
Switch2# show interfaces fa0/21 switchport | i Mode
Administrative Mode: dynamic auto
Operational Mode: static access
Access Mode VLAN: 700 (VLAN0700)
```

Напомню, компьютер Начальство-ПК1 относится к виртуальной сети с номером 700 и именем Executives, поэтому вам нужно вручную настроить интерфейс в качестве порта доступа виртуальной сети 700. Но обратите внимание, что на этот раз вы не используете команду `switchport mode access`. Напомню, что ранее в главе я сказал, что если протокол DTP не может согласовать транк виртуальной сети на порту, он превратит этот интерфейс в *порт доступа виртуальной сети*. Это здесь и произошло. Все, что вам нужно сделать, – настроить доступ к виртуальной локальной сети 700, выполнив команду `switchport access vlan 700`, а протокол DTP позаботится об остальном.

Теперь вы закончили настройку Коммутатора 2. Следующим шагом будет переход на компьютер Начальство-ПК1 и проверка, может ли он взаимодействовать так же, как и до перемещения. Напомню, что компьютер Начальство-ПК1 настроен на автоматическое получение IP-адреса с помощью протокола DHCP. Каждый раз, когда вы переключаете компьютер с одного порта на другой, рекомендуется перезагрузить его или выполнить команды `ipconfig / release` и `ipconfig / renew`, чтобы убедиться, что он может получить IP-адрес, будучи подключенным к новому порту.

Практикум

На компьютере Начальство-ПК1 выполните команды `ipconfig /release` и `ipconfig /renew`, чтобы убедиться, что компьютер получает IP-адрес из подсети 172.31.70.0. Подтвердите установку соединения по протоколу IP пингом IP-адреса SVI-интерфейса VLAN700 Коммутатора 1:

```
ping 172.31.70.254
```

Вы должны увидеть успешные пинги:

```
PS C:\>ping 172.31.70.254

Pinging 172.31.70.254 with 32 bytes of data:
Reply from 172.31.70.254: bytes=32 time<1ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
Reply from 172.31.70.254: bytes=32 time<1ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255

Ping statistics for 172.31.70.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

10.5. Изменение инкапсуляции транка

Поскольку трафик для нескольких виртуальных сетей проходит через транк, обоим коммутаторам необходим способ, позволяющий определить, какие Ethernet-кадры к какой виртуальной сети принадлежат. Они достигают этого, используя *протокол инкапсуляции внешней линии* (*Inter-Switch Link, ISL*), который маркирует (помечает тегами) каждый Ethernet-кадр с идентификатором виртуальной сети, когда тот передается по транку.

Практикум

Вы можете проверить, что ваши коммутаторы используют инкапсуляцию ISL, поискав значение `isl` в соответствующем выводе. Введите следующие команды и просмотрите вывод:

```
show interfaces trunk | i Encap|isl
show interfaces fa0/24 switchport | i isl
```

Обратите внимание, что обе команды отобразили протокол ISL для инкапсуляции транкового соединения:

```
Switch1#show interfaces trunk | i isl|Encap
Port        Mode             Encapsulation  Status         Native vlan
Fa0/24      auto             n-isl          trunking       1

Switch1#show interfaces fa0/24 switchport | i isl
Operational Trunking Encapsulation: isl
```

Протокол ISL отлично работает на коммутаторах Cisco, и в этом случае нет причин для того, чтобы поменять его. Но это проприетарный протокол компании Cisco, поэтому не работает с коммутаторами сторонних производителей. Если вам нужно сформировать транк между коммутатором компании Cisco и другого производителя, существует еще один тип инкапсуляции, который вам придется использовать, – 802.1Q.

Гипервизоры, такие как VMware ESXi, Microsoft Hyper-V, Citrix XenServer и Linux KVM, также используют транки 802.1Q. Хотя это выходит за рамки данной книги, важно знать, как настроить инкапсуляцию 802.1Q на коммутаторах Cisco. Это действительно очень легко!

Практикум

Если вы все еще авторизованы на Коммутаторе 2, выполните команду `exit` в приглашении `Switch2#`, чтобы вернуться в оболочку командной строки Коммутатора 1. Войдите в режим конфигурирования интерфейса FastEthernet0/24 и введите следующую команду:

```
switchport trunk encapsulation dot1q
```

Команда, ссылающаяся на `dot1q` вместо `802.1q`, может показаться немного странной, но она выглядит именно так. Помните, что если вы забудете эту небольшую деталь, то всегда можете запросить справочную систему.
Проверьте результат, выполнив команду `show interfaces trunk`.

Вы должны увидеть значение `802.1q` в качестве типа инкапсуляции для транка:

```
Switch1#show interfaces trunk | i Encap|trunking
Port        Mode             Encapsulation  Status         Native vlan
Fa0/24      auto             802.1q         trunking       1
```

Функционально вы не увидите никакой разницы между производительностью транков в режиме инкапсуляции 802.1Q и ISL. Оба достигают той же цели, но за кадром используются разные технологии.

Выбор типа инкапсуляции транка вручную имеет интересный побочный эффект: он позволяет вам прекратить использовать протокол DTP. Вместо зависимости от протокола DTP для согласования транка теперь вы можете настроить интерфейс так, чтобы он *всегда* работал как безусловный транковый порт.

Практикум

На Коммутаторе 1 войдите в режим конфигурирования интерфейса FastEthernet0/24, чтобы сделать его безусловным транковым портом:

```
switchport mode trunk
```

Вывод не отобразится, поэтому введите команду `show run int fa0/24`, чтобы увидеть рабочую конфигурацию.

Вы должны увидеть достаточно интересный результат:

```
Switch1#show run | section FastEthernet0/24
interface FastEthernet0/24
 switchport trunk encapsulation dot1q
```

```
switchport mode trunk
switchport port-security maximum 3
switchport port-security aging time 10
switchport port-security violation restrict
```

Строки `dynamic desirable` нигде нет! Протокол DTP полностью отключен, как и ISL. Это означает, что при желании вы можете подключить к этому порту коммутатор другого производителя или хост виртуализации и легко создать транк с инкапсуляцией 802.1Q. Хотя, наиболее вероятно, в типичной офисной среде большинство транков, с которыми вы столкнетесь, будут настроены между коммутаторами Cisco.

Обратите внимание, что используются некоторые настройки функции Port Security из предыдущей главы. В практическом задании вы поэкспериментируете с ней и посмотрите, влияет ли функция Port Security на созданный транк.

10.6. Команды, использованные в этой главе

Обращайтесь к командам, представленным в табл. 10.1, по мере выполнения практического задания.

Таблица 10.1. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`show interfaces fa0/24 switchport`	–	Отображает подробные сведения о рабочем состоянии порта, в том числе о том, находится ли он в режиме транка или доступа
`switchport mode dynamic desirable`	Интерфейс	Настраивает порт, чтобы попытаться согласовать транк с устройством на другом конце
`show interfaces trunk`	–	Отображает информацию обо всех активных транках виртуальной сети
`telnet 192.168.1.102`	–	Запускает встроенный в IOS клиент Telnet и подключается к адресу `192.168.1.102`
`switchport trunk encapsulation dot1q`	Интерфейс	Устанавливает тип инкапсуляции транка 802.1Q (в отличие от ISL)
`switchport mode trunk`	Интерфейс	Настраивает интерфейс как статический транковый порт, эффективно отключая протокол DTP

10.7. Практическое задание

В этом задании вы разберетесь, как различные настройки могут влиять на созданный транк. Вам, сетевому администратору Cisco, нужно понять, как технологии, которые вы освоили в предыдущих главах, могут повлиять на ваши теперешние действия. Вы уже видели, как отсутствующие на Коммутаторе 2 виртуальные сети помешали передавать трафик из этих сетей через созданный транк. Следуйте приведенным ниже шагам и ответьте на следующие вопросы:

1. На Коммутаторе 2 настройте на интерфейсе FastEthernet0/24 режим `dynamic desirable`. Убедитесь, что транк все еще работает.
2. Повторно настройте интерфейс как безусловный транковый порт. Транк все еще работает?
3. Попытайтесь включить на интерфейсе функцию Port Security, выполнив команду `switchport port-security`. Что происходит?
4. С компьютера Начальство-ПК1 пропингуйте IP-адрес компьютера Кадры-ПК1, `172.31.60.51`. Удалось? Если нет, то почему?
5. Сохраните рабочие конфигурации как на Коммутаторе 1, так и на Коммутаторе 2.

Глава 11

Автоматическая настройка виртуальных сетей с помощью протокола VTP

Теперь, когда вы настроили транк между коммутаторами, пришло время представить вам еще одну интересную функцию системы IOS, которая может немного сэкономить время при настройке виртуальных сетей на новых коммутаторах.

В предыдущей главе вы настраивали Коммутатор 2 и вам пришлось вручную создавать на нем виртуальные сети `600` и `700`. Это было неважно, потому что вам пришлось делать это только на одном коммутаторе. Но как вы взглянете на ту же задачу, если вам пришлось бы добавить 10 новых коммутаторов и на каждом из них понадобилось настроить 100 виртуальных сетей? Хотя добавлять виртуальные сети вручную не так и сложно, выполнение задания займет колоссальное количество времени.

Чтобы решить эту проблему, Cisco предоставила классную функцию, называемую *протокол управления виртуальными сетями* (*VLAN Trunking Protocol, VTP*). Протокол VTP избавляет вас от необходимости вручную создавать виртуальные сети на каждом коммутаторе в вашей топологии. Вместо этого вы создаете их только на одном коммутаторе, а протокол VTP автоматически создает и именует те же самые виртуальные сети на всех других коммутаторах, если они подключены через транковые соединения и вы правильно настроили протокол VTP.

Рисунок 11.1 иллюстрирует цель этой главы. Протокол VTP основан на конфигурации клиент/сервер. VTP-сервер – это коммутатор (в данном случае Коммутатор 1), на котором вы настроите все ваши виртуальные сети. Процесс VTP, запущенный на Коммутаторе 1, отправит VTP-оповещения через транк, подключенный к Коммутатору 2. VTP-оповещения – это сообщения, содержащие список виртуальных сетей, настроенных на Коммутаторе 1. Коммутатор 2 получит VTP-оповещения и автоматически создаст виртуальные сети, настроенные на Коммутаторе 1.

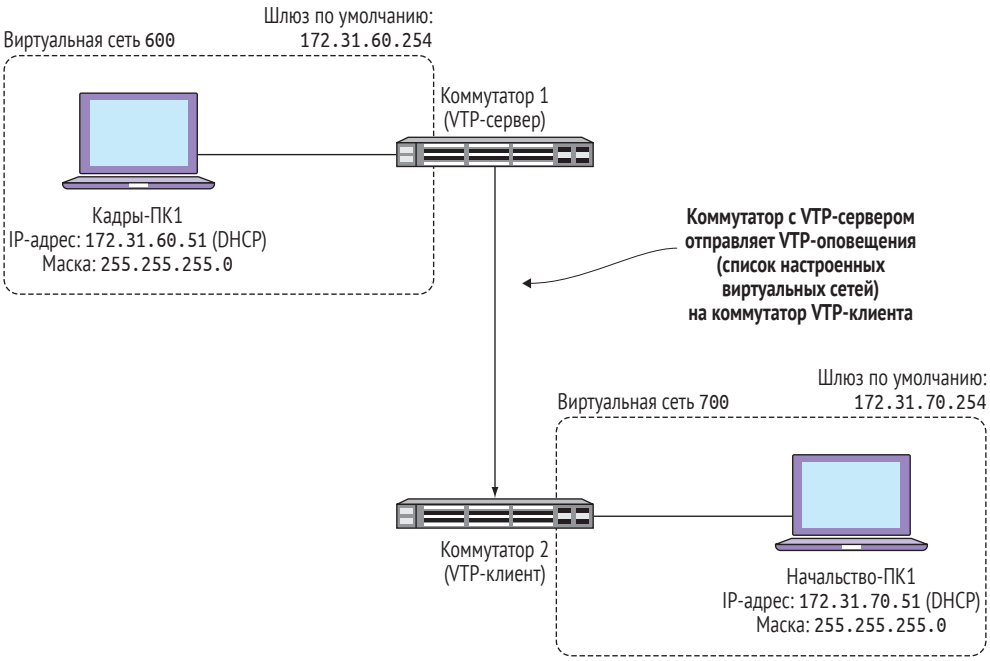

Рис. 11.1 ❖ Коммутатор 1 – это VTP-сервер, а Коммутатор 2 – его клиент. Коммутатор 1 отправляет VTP-оповещения, содержащие информацию о настроенных на нем виртуальных сетях

11.1. Пара слов в предостережение

Прежде чем начать, мне нужно дать вам два больших предупреждения о протоколе VTP. Во-первых, если протокол VTP неправильно настроен, сеть может лечь за считанные секунды. Поскольку протокол может автоматически создавать и удалять виртуальные сети, существует вероятность, что он удалит крайне важные виртуальные сети в рабочей среде. Ранее я отметил в книге, что буду предупреждать вас, что «если что-нибудь, чему я вас учу, может негативно повлиять на реальную сеть, не применяйте это». Это первое предупреждение. *Не настраивайте протокол VTP в рабочей сети, следуя каким-либо процессам, которые ваша организация считает необходимыми, но которые могут потенциально привести к катастрофическим последствиям!*

Во-вторых, из-за возможности протокола VTP делать очень неприятные вещи (например, удаление виртуальных сетей) многие организации его игнорируют. Лично я предпочитаю не использовать этот проект в большинстве случаев. Но многие организации используют его и, что более важно, вам, сетевому администратору Cisco, нужно понять, как его настроить, чтобы вы могли справиться с последствиями, если что-то пойдет не так!

Ниже приведены основные шаги, которые вы будете выполнять для запуска VTP-сервера.

1. Настройте Коммутатор 1 как VTP-сервер.
2. Настройте Коммутатор 2 как VTP-клиент.
3. Создайте виртуальные сети `900–999` на Коммутаторе 1.
4. Настройте функцию отсечения на VTP-сервере.

Я же говорил вам, что все просто! Поехали!

11.2. Настройка Коммутатора 1 в качестве VTP-сервера

Перед внесением изменений в конфигурацию всегда полезно получить представление о рабочей конфигурации. Таким образом, если что-то пойдет не так, у вас будет хотя бы некоторое представление о том, что изменилось.

Практикум

На Коммутаторе 1 выполните команду `show vtp status`, чтобы просмотреть рабочую конфигурацию протокола VTP.

Вы должны увидеть нечто подобное:

```
Switch1#show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 : cisco
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0023.ab40.8e00
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Feature VLAN:
--------------
VTP Operating Mode              : Transparent
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 8
Configuration Revision          : 0
MD5 digest                      : 0xF9 0xD6 0xB0 0x82 0x71 0x06 0x3B 0x92
                                  0x04 0xC3 0xCB 0xC8 0xFD 0x37 0x91 0xC7
```

Не беспокойтесь, если ваш результат непохож на мой. Вы все равно измените настройки. Я хочу обратить ваше внимание на конкретные моменты конфигурации, которые вы будете изменять на Коммутаторе 1.

`VTP Domain Name` – это строка, которую использует каждый участник протокола VTP, чтобы определить, следует прислушиваться или игнорировать информацию в VTP-оповещениях. Представьте эти сведения как фамилию. Если ваша фамилия Иванов и вы слышите, как кто-то кричит на Иванова, вы, вероятно, обратите внимание. Если ваша фамилия не Иванов, вы просто проигнорируете крики.

Параметру VTP Operating Mode присвоено значение Transparent, что означает, что протокол VTP игнорирует VTP-оповещения, но передает их подключенным коммутаторам. Прозрачный режим эффективно отключает VTP на коммутаторе, независимо от значения в строке VTP Domain Name.

В данном случае требуется, чтобы Коммутатор 1 работал в качестве VTP-сервера и чтобы он отправлял VTP-оповещения подключенным коммутаторам.

Практикум

На Коммутаторе 1 выполните следующую команду, чтобы настроить его в качестве VTP-сервера для VTP-домена Cisco:

```
Switch1(config)#vtp mode server
```

Вы должны увидеть единственную строку вывода:

```
Setting device to VTP Server mode for VLANS.
```

Следующим шагом будет установка пароля доступа к VTP-серверу. Целью пароля является предотвращение случайного сбоя сети из-за неправильной конфигурации протокола VTP.

Практикум

Установите пароль MoL для доступа к VTP-серверу на Коммутаторе 1:

```
Switch1(config)#vtp password MoL
```

Вы должны получить следующее подтверждение:

```
Setting device VTP password to MoL
```

На этом настройка Коммутатора 1 в качестве VTP-сервера завершена.

11.3. Настройка Коммутатора 2 в качестве VTP-клиента

Опять же, рекомендуется проверить существующую конфигурацию перед внесением изменений. Проверим конфигурацию протокола VTP на Коммутаторе 2.

Практикум

На Коммутаторе 2 выполните команду show vtp status.

Вы должны получить следующий вывод:

```
Switch2#show vtp status
VTP Version capable             : 1 to 3
```

```
VTP version running             : 1
VTP Domain Name                 : cisco      ◄— Коммутатор 2 автоматически заполняет доменное имя,
VTP Pruning Mode                : Disabled       полученное от Коммутатора 1
VTP Traps Generation            : Disabled
Device ID                       : 0024.5088.6d80
Configuration last modified by 192.168.1.102 at 3-1-93 00:58:04
Local updater ID is 192.168.1.102 on interface Vl1 (lowest numbered VLAN
     interface found)
Feature VLAN:
--------------
VTP Operating Mode              : Server
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 8
Configuration Revision          : 0
MD5 digest                      : 0xB2 0x50 0x7A 0x9B 0x04 0x97 0xBD 0x83
                                  0x5A 0xFD 0xC4 0x15 0xFB 0xF5 0x4C 0x6F
*** MD5 digest checksum mismatch on trunk: Fa0/24 ***   ◄— Пароли VTP на Коммутаторах 1 и 2
                                                           не совпадают
```

Обратите внимание на два момента.

Параметру `VTP Domain Name` присвоено значение `cisco`. Я не настраивал это, как и вы. Это произошло автоматически. По умолчанию VTP стартует с пустым доменным именем. Когда вы настроили Коммутатор 1 в качестве VTP-сервера, он отправил VTP-оповещение, включающее имя домена VTP. Коммутатор 2 получил это оповещение и автоматически установил его собственное доменное имя VTP для соответствия.

Следующее, на что следует обратить внимание, – это ошибка, указывающая на несоответствие контрольной суммы MD5 (`MD5 digest checksum mismatch`). Хотя это и неочевидно, это означает, что пароль VTP, настроенный на Коммутаторе 1, не соответствует несуществующему паролю на Коммутаторе 2. Это сообщение исчезнет после завершения настройки.

Практикум

Настройте Коммутатор 2 в качестве VTP-клиента, используя следующую команду:

```
Switch2(config)#vtp mode client
```

Вы должны увидеть такое сообщение:

```
Setting device to VTP Client mode for VLANS.
```

Затем установите пароль доступа к VTP-серверу – `MoL`:

```
Switch2(config)#vtp password MoL
```

Вы должны увидеть подтверждающее сообщение:

```
Setting device VTP password to MoL
```

11.4. Создание виртуальных сетей на Коммутаторе 1

Теперь вы готовы проверить настройки протокола VTP. Вы сделаете это, создав несколько новых виртуальных сетей на Коммутаторе 1, а затем проверите, что они распространяются на Коммутатор 2.

Практикум

На Коммутаторе 1 создайте виртуальные сети с номерами `900–999`:

```
Switch1(config)#vlan 900-999
Switch1(config-vlan)#exit
```

Перейдите на Коммутатор 2 и снова проверьте состояние VTP:

```
Switch2(config)#do sh vtp status
```

Вы должны увидеть следующее:

```
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 : cisco
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0024.5088.6d80                    IP-адрес SVI-интерфейса
Configuration last modified by 192.168.1.101 at 3-1-93 01:39:02  <-- VLAN1 Коммутатора 1

Feature VLAN:
--------------
VTP Operating Mode              : Client
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 108  <-- Количество виртуальных сетей на Коммутаторе 2
Configuration Revision          : 1
MD5 digest                      : 0x6B 0xCB 0x64 0xBE 0x29 0x9D 0xBF 0xB9
                                  0x8C 0x04 0xC0 0xE9 0x3E 0x7F 0x51 0x52
```

Обратите внимание на строку, в которой говорится, что `Configuration last modified by 192.168.1.101`. Это IP-адрес SVI-интерфейса `VLAN1` Коммутатора 1. Если вы забыли, какой коммутатор является VTP-сервером, выполните команду `show vtp status` и взгляните на эту строку в выводе.

Также обратите внимание, что количество существующих виртуальных сетей увеличилось с 8 до 108 благодаря добавлению сетей с номерами `900–999`. Вы только создали 100 новых виртуальных сетей на Коммутаторах 1 и 2, выполнив одно изменение конфигурации лишь на Коммутаторе 1! Чтобы по-настоящему оценить масштаб, взгляните на виртуальные сети на Коммутаторе 2.

Практикум

На Коммутаторе 2 выполните команду `show vlan`.

Здесь я не буду приводить полный вывод, но у вас должно быть несколько экранов вывода, перечисляющих все новые виртуальные сети:

```
Switch2#sh vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/22, Fa0/23, Gi0/1, Gi0/2
20   VOICE                            active
600  HR                               active
700  Executives                       active    Fa0/21
900  VLAN0900                         active
901  VLAN0901                         active
902  VLAN0902                         active
903  VLAN0903                         active
904  VLAN0904                         active
905  VLAN0905                         active
906  VLAN0906                         active
907  VLAN0907                         active
908  VLAN0908                         active
909  VLAN0909                         active
910  VLAN0910                         active
--More--
```

Шикарно, что можно создавать сотни виртуальных сетей на нескольких коммутаторах за секунды, но есть одно «но». Если вы хотите создать, удалить или переименовать виртуальную сеть на VTP-клиенте – в данном случае Коммутаторе 2, – вам нужно сделать это на сервере – Коммутаторе 1. Если вы попытаетесь изменить виртуальную сеть на Коммутаторе 2, протокол VTP остановит вас.

Практикум

На Коммутаторе 2 попробуйте войти в режим конфигурирования виртуальной сети:

```
Switch2(config)#vlan 900
```

Ваша попытка будет безуспешна. Как только вы попробуете войти в режим настройки виртуальной сети, протокол VTP остановит вас следующим сообщением:

```
VTP VLAN configuration not allowed when device is in CLIENT mode.
```

Такое поведение может показаться слегка неудобным, особенно если вы торопитесь создать новую виртуальную сеть. Решение состоит в том, чтобы перейти на VTP-сервер – Коммутатор 1 – и внести изменения на нем.

Практикум

На Коммутаторе 1 задайте любое имя для виртуальной сети `900`:

```
Switch1(config)#vlan 900
Switch1(config-vlan)#name Marketing
Switch1(config-vlan)#exit
```

В этот раз вы не должны получить никаких сообщений об ошибке.
Перейдите на Коммутатор 2 и проверьте распространение информации:

```
Switch2#show vlan brief | i Name|900
```

Вы должны увидеть виртуальную сеть `900` с новым именем:

```
VLAN Name                             Status    Ports
900  Marketing                        active
```

11.5. Включение VTP-отсечения

Название протокола *VLAN Trunking Protocol* немного вводит в заблуждение, поскольку он, как следует из имени, предлагает сыграть роль в создании транка виртуальной сети. Но, как вы уже знаете из прошлой главы, вы можете создавать транки виртуальной сети без участия протокола VTP. Итак, почему же протокол VTP носит такое название?

Когда вы используете протокол VTP для автоматического создания виртуальных сетей на нескольких коммутаторах, трафик из каждой из этих сетей автоматически проходит через ваши существующие транки. В этом случае протокол VTP не создает и не может создать транк виртуальной сети между Коммутаторами 1 и 2. Но поскольку эти коммутаторы уже с транками, создание виртуальной сети на обоих коммутаторах, например с номером `900`, приводит к прохождению трафика созданной сети по этому транку. Чтобы понять, что я имею в виду, взгляните на виртуальные сети, которые активны в транке.

Практикум

На Коммутаторе 1 выполните команду `show interfaces trunk`.

Вы должны увидеть следующий вывод:

```
Switch1#sh interfaces trunk

Port        Mode             Encapsulation  Status         Native vlan
Fa0/24      on               802.1q         trunking       1

Port        Vlans allowed on trunk
```

```
Fa0/24      1-4094

Port        Vlans allowed and active in management domain
Fa0/24      1,20,600,700,900-999

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/24      1,20,600,700,900-999
```

Все виртуальные локальные сети, которые существуют на Коммутаторе 1, передают трафик через транк с Коммутатором 2

Взгляните на список виртуальных сетей ниже строки `Vlans in spanning tree forwarding state and not pruned`. Каждая из только что созданных виртуальных сетей с `900` по `999`, а также существующие виртуальные сети передают трафик через транк, как это показано на рис. 11.2.

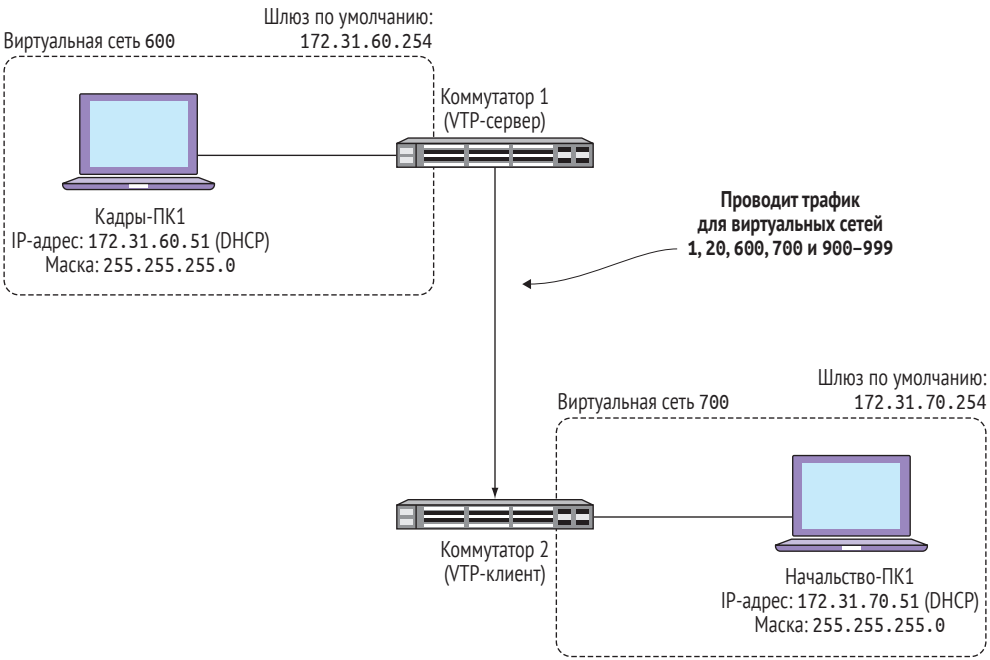

Рис. 11.2 ❖ Все виртуальные локальные сети передают трафик через транк между Коммутаторами 1 и 2

Как вы знаете, любая виртуальная сеть является отдельным широковещательным доменом. В рабочей сети, притом, что по каждой из этих виртуальных сетей проходит трафик, каждая передача трафика каждой виртуальной сети будет проходить через этот транк! Это может потреблять бóльшую часть, если не всю ширину полосы пропускания этого транка со скоростью 100 Мбит/с.

Решение состоит в том, чтобы использовать функцию протокола VTP, называемую *отсечением*, чтобы предотвратить появление в транке трафика неиспользуемых виртуальных сетей. Что представляет собой неиспользуемая виртуальная сеть, определить не так просто, как кажется на первый взгляд.

Взгляните еще раз на рис. 11.2. Обратите внимание, что в виртуальных сетях 900–999 нет устройств. Говоря техническим языком, нет коммутируемых виртуальных интерфейсов (SVI-интерфейсов) или портов доступа, относящихся к этим виртуальным сетям. Следовательно, эти виртуальные сети не используются. Вы можете убедиться в этом, проанализировав базу данных виртуальных сетей на Коммутаторах 1 и 2.

Практикум

На Коммутаторах 1 и 2 выполните команду `show vlan brief | i Fa`.

На Коммутаторе 1 вы должны увидеть только порты, принадлежащие к виртуальным сетям 1, 20, 600 и 700:

```
Switch1#show vlan brief | i Fa
1    default                       active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                             Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                             Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                             Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                             Fa0/17, Fa0/18, Fa0/22, Fa0/23
20   VOICE                         active    Fa0/19, Fa0/20
600  HR                            active    Fa0/19, Fa0/20
700  Executives                    active    Fa0/21
```

На Коммутаторе 2 вы должны увидеть членство только в виртуальных сетях 1 и 700:

```
Switch2#sh vlan brief | i Fa
1    default                       active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                             Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                             Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                             Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                             Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                             Fa0/22, Fa0/23, Gi0/1, Gi0/2
700  Executives                    active    Fa0/21
```

По методу исключения виртуальные сети 900–999 не используются на Коммутаторах 1 и 2. Функция отсечения протокола VTP может автоматически определять, какие виртуальные сети не используются, и отсекать их трафик от транка, но сначала вам нужно явно включить эту функцию.

Практикум

Включите VTP-отсечение на Коммутаторе 1, введя следующую команду в режиме глобальной конфигурации:

```
vtp pruning
```

Вы должны увидеть сообщение о том, что отсечение включено:

```
Switch1(config)#vtp pruning
Pruning switched on
```

Кстати, вам нужно ввести эту команду на VTP-сервере. Она не будет работать на любых VTP-клиентах. Чтобы подтвердить, что VTP-отсечение работает так, как ожидалось, рекомендую взглянуть на транк.

Практикум

На Коммутаторе 1 выполните команду `show interfaces trunk`.

Вы должны получить следующий вывод:

```
Switch1#show interfaces trunk

Port        Mode             Encapsulation  Status         Native vlan
Fa0/24      on               802.1q         trunking       1

Port        Vlans allowed on trunk
Fa0/24      1-4094

Port        Vlans allowed and active in management domain
Fa0/24      1,20,600,700,900-999

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/24      1,700   ◄── По транку проходит трафик только из виртуальных сетей 1 и 700
```

Взгляните еще раз на раздел `Vlans in spanning tree forwarding state and not pruned`. Функция отсечения протокола VTP *отсекла* все виртуальные сети, кроме `1` и `700`. Удаление виртуальных сетей `900–999` не должно удивлять, но почему отсечена виртуальная сеть `600`? Чтобы ответить на этот вопрос, взгляните на рис. 11.3.

Единственное устройство, которое является членом виртуальной сети `600`, – это компьютер Кадры-ПК1, подключенный к Коммутатору 1. Но на Коммутаторе 2 в виртуальной сети `600` не зарегистрировано устройств. Следовательно, виртуальная сеть `600` не используется на Коммутаторе 2, поэтому процесс VTP на Коммутаторе 1 отсекает его от транка.

Дополнительно

Протокол VTP никогда не отсекает от транка трафик виртуальной сети `1`. Я называю виртуальную сеть `1` неизменной, потому что вы не можете ее удалить или отключить. Это хорошо, поскольку протокол VTP использует виртуальную сеть `1` для отправки и приема VTP-оповещений.

Возникает вопрос: как Коммутатор 1 знает, какие виртуальные сети нужно отсечь? Об этом сообщает Коммутатор 2.

Практикум

На Коммутаторе 2 выполните команду `show interfaces pruning`.

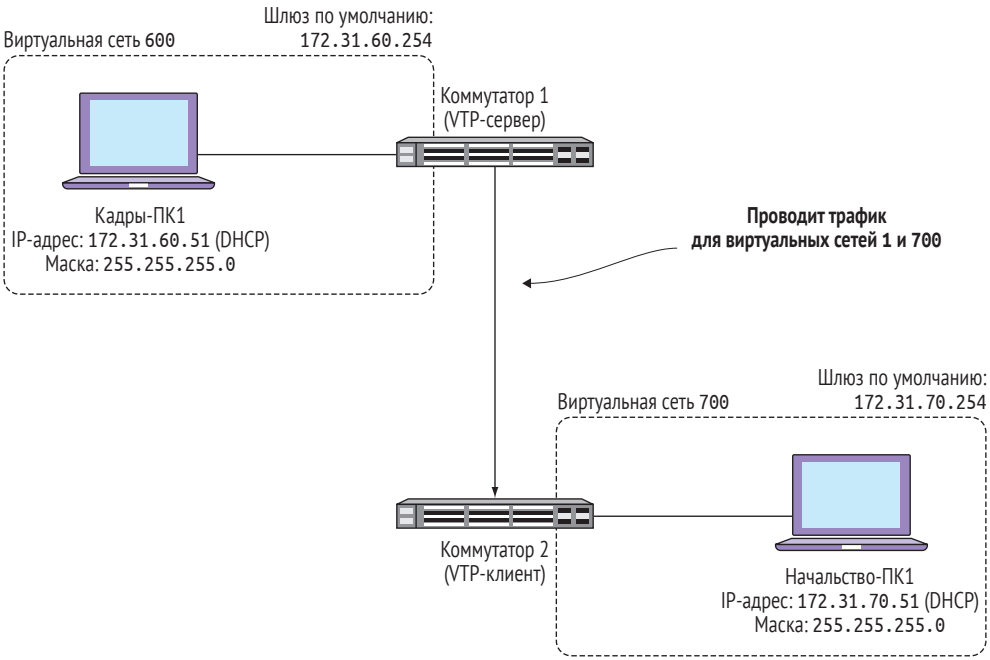

Рис. 11.3 ❖ Трафику виртуальных сетей 1 и 700 разрешено «прогуливаться» по транку между Коммутаторами 1 и 2

Вот что вы должны увидеть:

```
Switch2#show interfaces pruning

Port              Vlans pruned for lack of request by neighbor
Fa0/24            900-999

Port              Vlan traffic requested of neighbor
Fa0/24            1,700
```

Коммутатор 2 запросил у Коммутатора 1 разрешение на передачу трафика этих виртуальных сетей

Под строкой `Vlan traffic requested of neighbor` перечислены виртуальные сети, трафик которых Коммутатор 2 хочет получить от Коммутатора 1: это сети 1 и 700. Другими словами, Коммутатор 2 не хочет, чтобы Коммутатор 1 отсекал эти сети от транка. То, что это единственные виртуальные сети, трафик которых Коммутатор 1 допускает в транк, не случайно.

Следует иметь в виду, что отсечение виртуальных сетей необязательно симметрично, т. е. сети, отсекаемые Коммутатором 1, могут не совпадать с сетями, отсекаемыми Коммутатором 2. В практическом задании вы авторизуетесь на Коммутаторе 2 и разберетесь, что именно я имею в виду.

11.6. Команды, использованные в этой главе

Обращайтесь к табл. 11.1 по мере работы над практическим заданием.

Таблица 11.1. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`show vtp status`	–	Выводит информацию о рабочей конфигурации протокола VTP
`vtp mode server`	Глобальный	Настраивает коммутатор в качестве VTP-сервера
`vtp password MoL`	Глобальный	Устанавливает пароль VTP: `MoL`
`vtp mode client`	Глобальный	Настраивает коммутатор в качестве VTP-клиента
`vlan 900-999`	Глобальный	Создает виртуальные сети с номерами с `900` по `999` включительно
`show interfaces trunk`	–	Выводит информацию о транковых соединениях, включая список виртуальных сетей, которые передают трафик через эти транки
`vtp pruning`	Глобальный	Включает функцию VTP-отсечения
`show interfaces pruning`	–	Демонстрирует, какие виртуальные сети попали под действие функции VTP-отсечения

11.7. Практическое задание

В этом задании вы увидите, как изменения, внесенные вами в конфигурацию VTP-сервера, влияют на клиента – Коммутатор 2. Вы также увидите, как изменения, внесенные вами на Коммутаторе 2, влияют на функцию VTP-отсечения.

1. Удалите виртуальные сети `901–999` на Коммутаторе 1, выполнив команду `no vlan 901-999`. Что происходит с этими виртуальными сетями на Коммутаторе 2?
2. На Коммутаторе 2 создайте SVI-интерфейс для виртуальной сети `900`. Коммутатор 1 все еще отсекает трафик виртуальной сети `900` от транка?
3. Коммутатор 1 отсекает виртуальные сети `20` и `600` от транка с Коммутатором 2. Какие виртуальные сети от транка отсекает Коммутатор 2?
4. Какие виртуальные сети не отсекает Коммутатор 2?
5. Сохраните ваши конфигурации!

Глава 12

Защита от петель коммутации с помощью протокола STP

В предыдущей главе вы подключили два коммутатора – Коммутатор 1 и Коммутатор 2 – через одно соединение Ethernet. Представьте, что оба эти коммутатора находятся в рабочей сети и к ним подключены десятки пользователей. Если соединение между коммутаторами прервется, устройства на Коммутаторе 1 не смогут связаться с устройствами на Коммутаторе 2, и наоборот.

Решение – добавить избыточное соединение между FastEthernet0/23 на Коммутаторах 1 и 2, как показано на рис. 12.1. Если исходное соединение нарушено по какой-либо причине, коммутаторы могут связаться по дополнительной линии.

На первый взгляд, такая конфигурация превосходна. Кажется, что, добавив дополнительный канал, вы добавляете не только надежность, но также дополнительную полосу пропускания. Однако эта конфигурация не так хороша, как кажется.

Добавляя дополнительное соединение, вы по определению не можете получить трафик, проходящий по двум каналам одновременно. Чтобы понять, почему, рассмотрим следующий сценарий.

Предположим, компьютер Начальство-ПК1 генерирует широковещательный Ethernet-кадр и отправляет его в сеть. Как было описано в главе 2, Коммутатор 2 передаст этот кадр в каждый порт, включая оба порта, соединенных с Коммутатором 2, – FastEthernet0/23 и FastEthernet0/24. Это проиллюстрировано на рис. 12.2.

Коммутатор 1 получит кадр на оба порта. Так как оба идентичных кадра являются широковещательными, он перешлет их на все порты, включая те, что соединены с Коммутатором 2. Вкратце: пакеты будут курсировать между Коммутаторами 1 и 2 бесконечно. Возникнет *петля коммутации*.

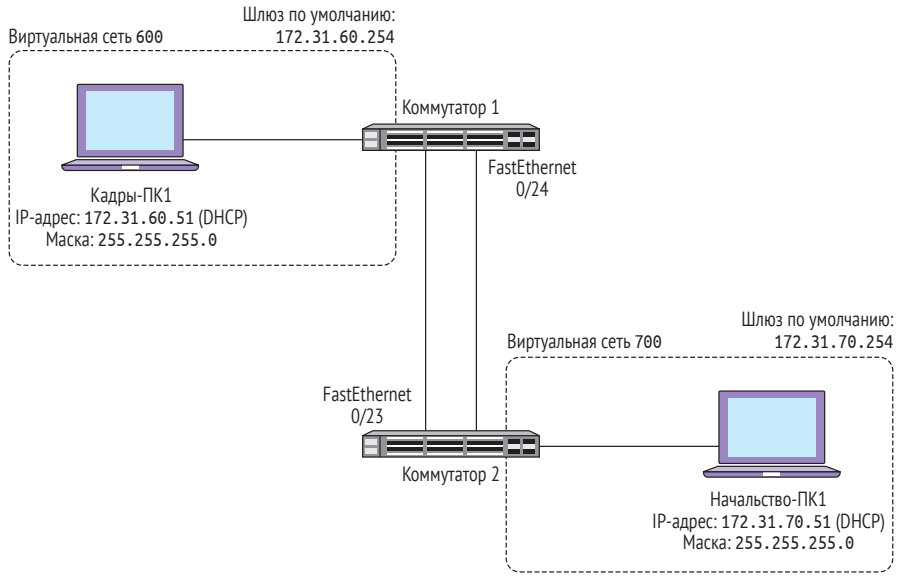

Рис. 12.1 ❖ Коммутаторы 1 и 2 с резервными соединениями. Обратите внимание, что номера интерфейсов одинаковы на каждом конце канала

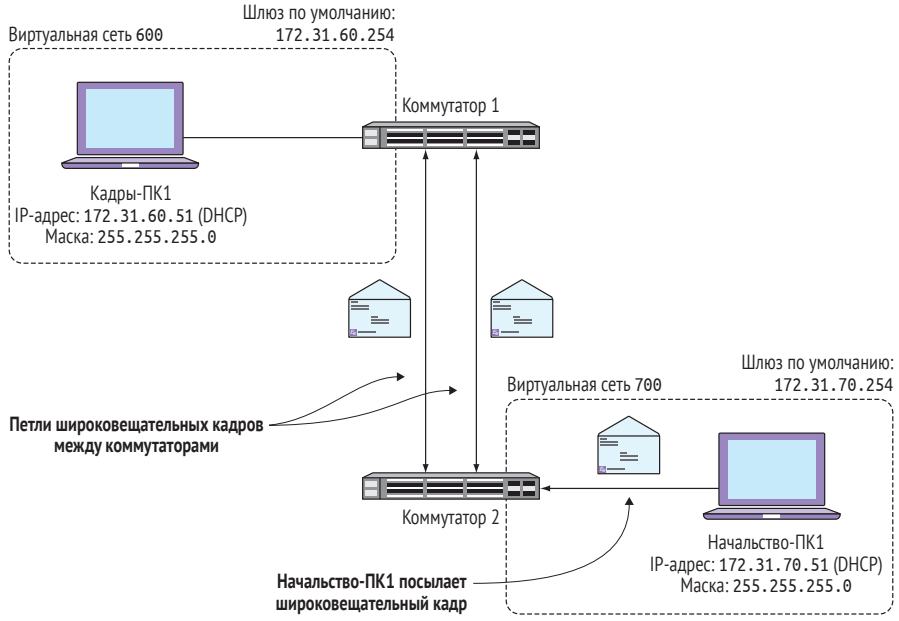

Рис. 12.2 ❖ Если Коммутатор 1 и Коммутатор 2 должны использовать оба соединения, широковещательный кадр от компьютера Начальство-ПК1 приведет к петле коммутации

Не вдаваясь в скучные детали, скажу, что петля коммутации приведет к тому, что исходный широковещательный кадр будет повторен много раз и послан по межкоммутаторным соединениям снова и снова, пока не будет исчерпана полоса пропускания или процессор каждого коммутатора не окажется перегруженным. Это приводит всегда к одному результату: устройства, подключенные к обоим коммутаторам, потеряют сетевое соединение. Смешно, но это именно та проблема, которую была призвана предотвратить избыточная связь.

12.1. Как работает протокол STP

Чтобы решить проблему петель коммутации, системный инженер Радиа Перлман предоставила нечто под названием *протокол остовного дерева (Spanning Tree Protocol, STP)*. Большинство сетевых администраторов называют его просто *Spanning Tree*, или более кратко *STP*. Протокол STP решает проблему петель коммутации, и на коммутаторах Cisco он включен по умолчанию. Но прежде чем вы сможете получить практическое представление о том, как работает этот протокол, вам нужно создать еще одно соединение между коммутаторами.

Практикум

Соедините с помощью Ethernet-кабеля порты FastEthernet0/23 обоих коммутаторов.

На Коммутаторе 1 настройте порт FastEthernet0/23 в качестве транкового для виртуальной сети:

```
Interface fa0/23
switchport trunk encapsulation dot1q
switchport mode trunk
```

Проверьте вашу конфигурацию, выполнив команду `show interfaces trunk`.

Не бойтесь создать петлю коммутации. Запомните, что протокол STP включен по умолчанию и предотвратит появление петель. Чтобы увидеть, как он это делает, взгляните на рис. 12.3.

Обратите внимание, что протокол STP блокирует порт FastEthernet0/24 на Коммутаторе 2. По умолчанию протокол STP блокирует порт с наивысшим номером в пользу порта с более низким. Важно понимать, что *блокировка* не означает выключение порта, его деактивацию или прерывание соединения. Это означает, что процесс STP на Коммутаторе 2 просто блокирует входящие/исходящие Ethernet-кадры на порту FastEthernet0/24. Вы можете проверить это в оболочке командной строки.

Практикум

На Коммутаторе 2 выполните команду `show spanning-tree vlan 700`.

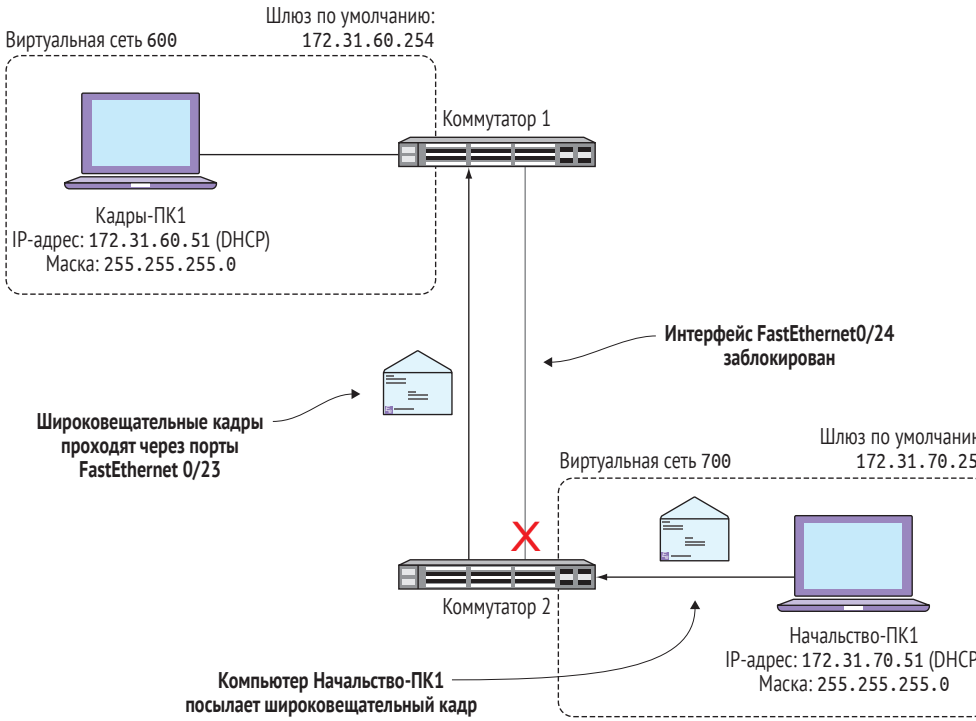

Рис. 12.3 ❖ Протокол STP предотвращает возникновение петель коммутации путем блокирования порта FastEthernet0/24 на Коммутаторе 2

Вы увидите следующий, несколько непонятный вывод:

```
Switch2#show spanning-tree vlan 700

VLAN0700
  Spanning Tree enabled protocol ieee
  Root ID     Priority    33468
              Address     0023.ab40.8e00
              Cost        19
              Port        25 (FastEthernet0/23)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    33468  (priority 32768 sys-id-ext 700)
              Address     0024.5088.6d80
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----
Fa0/21              Desg FWD 19        128.23   P2p
Fa0/23              Root FWD 19        128.25   P2p
Fa0/24              Altn BLK 19        128.26   P2p
```

Протокол STP заблокировал этот порт для предотвращения петель коммутации

Хотя это не очень очевидно, последняя строка показывает, что протокол STP перевел порт FastEthernet0/24 в состояние BLK, что означает *blocked – заблокирован*. Порт FastEthernet0/23, напротив, он оставил в состоянии FWD, что значит *forwarding*, т. е. *передача кадров*. Используя более общую терминологию, вы можете рассматривать порт FastEthernet0/24 как *запасной* или *резервный*, а порт FastEthernet0/23 – как *активный*.

Дополнительно

Протокол STP относится к разряду нетривиальных тем, заставляющих задуматься даже опытных сетевых администраторов. В этой книге не рассказывается о том, как протокол STP определяет, какой порт блокировать, здесь содержатся только подсказки, как настроить ваше окружение.

12.1.1. Как протокол STP действует в случае потери соединения

Смысл прокладки двух соединений состоит в том, чтобы избежать недоступности сети при прерывании одного из них. У вас есть два соединения, но протокол STP допускает трафик только по одному из них: FastEthernet0/23. Теперь симулируем потерю связи по этому соединению, чтобы увидеть, как будет реагировать протокол STP.

Практикум

На Коммутаторе 2 закройте интерфейс FastEthernet0/23:

```
interface fa0/23
shutdown
```

Выполните команду `show spanning-tree vlan 700`.

Вы должны увидеть следующее:

```
Switch2#sh spanning-tree vlan 700

VLAN0700
  Spanning Tree enabled protocol ieee
  Root ID    Priority    33468
             Address     0023.ab40.8e00
             Cost        19
             Port        26 (FastEthernet0/24)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    33468  (priority 32768 sys-id-ext 700)
             Address     0024.5088.6d80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----
Fa0/21              Desg FWD 19        128.23   P2p
Fa0/24              Root FWD 19        128.26   P2p
```

Протокол STP перевел порт FastEthernet0/24 в состояние передачи (FWD), т. к. FastEthernet0/23 более не доступен

Обратите внимание, что порт FastEthernet0/23, который вы только что закрыли, нигде не виден. Вместо этого можно увидеть порт FastEthernet0/24 в режиме передачи, что означает, что Коммутатор 2 теперь использует это соединение для передачи трафика на Коммутатор 1. Кстати, если в вашем выводе порт FastEthernet0/24 указан не в состоянии передачи, подождите около 30 секунд и попробуйте еще раз. Я вскоре объясню, почему.

Как бы ни был хорош протокол STP, он не идеален. Есть еще один недостаток, который проявляется, когда порт закрывается или открывается.

Практикум

С компьютера Начальство-ПК1 непрерывно пингуйте SVI-интерфейс виртуальной сети 700 на Коммутаторе 1:

```
Ping 172.31.70.254 –t
```

На Коммутаторе 2 включите порт FastEthernet0/23:

```
Interface fa0/23
No shut
```

Подождите 30 секунд.

По прошествии примерно 30 секунд вы должны увидеть следующие строки вывода:

```
Switch2(config-if)#no shut
Switch2(config-if)#
*Mar    1 03:07:47.234: %LINK-3-UPDOWN: Interface FastEthernet0/23, changed state to up
*Mar    1 03:07:50.883: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23,
      changed state to up
*Mar    1 03:08:19.899: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan900, changed
      state to up
*Mar    1 03:08:19.916: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed
      state to up
```

Особое внимание обратите на время. Интерфейс поднялся в 3:07:47, но SVI-интерфейсы виртуальных сетей 1 и 900 не заработали, пока не прошло 30 секунд. Причина этого – то, что протокол STP не переводит интерфейс в состояние передачи немедленно. В течение этих 30 секунд трафик между Коммутаторами 1 и 2 не циркулирует.

Непрерывный пинг на компьютере Начальство-ПК1 демонстрирует это со всей очевидностью:

```
PS C:\Users\Administrator> ping 172.31.70.254 -t

Pinging 172.31.70.254 with 32 bytes of data:
Reply from 172.31.70.254: bytes=32 time<1ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
Request timed out.
Request timed out.
```

```
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Reply from 172.31.70.254: bytes=32 time=2ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
```

Хотя шесть тайм-аутов пинга не смертельны для сети, их наверняка заметят пользователи. Что еще более важно, протокол STP выполняет свою работу, о чем свидетельствует возможный успех пингов. Вы также можете увидеть это, снова проверив статус протокола STP.

Практикум

Проверьте, что порт FastEthernet0/23 опять в состоянии передачи:

```
Show spanning-tree vlan 700
```

Вы должны увидеть следующее:

```
Switch2#sh spanning-tree vlan 700

VLAN0700
  Spanning Tree enabled protocol ieee
  Root ID    Priority    33468
             Address     0023.ab40.8e00
             Cost        19
             Port        25 (FastEthernet0/23)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    33468  (priority 32768 sys-id-ext 700)
             Address     0024.5088.6d80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----
Fa0/21              Desg FWD 19        128.23   P2p
Fa0/23              Root FWD 19        128.25   P2p
Fa0/24              Altn BLK 19        128.26   P2p
```

Теперь вы видите, что все порты снова находятся в состоянии, установленном изначально при избыточном подключении.

Дополнительно

Опять же, я не собираюсь вдаваться в различные подробности того, как работает протокол STP. Просто учитывайте, что по умолчанию протокол STP имеет 30-секундную задержку. Задержка в два раза превышает значение таймера задержки передачи. Можно сократить эту задержку до 8 секунд, изменив таймер задержки передачи на 4 секунды, используя команду `spanning-tree vlan 1-4094 forward-time 4`.

Вы или начальство в вашей организации могут счесть 30-секундную задержку неприемлемой. В таком случае у вас есть другая возможность.

12.2. Протокол RSTP

Быстрый протокол остовного дерева (Rapid Spanning Tree Protocol, RSTP) работает так же, что и протокол STP, только быстрее. Наибольшее неудобство – в том, что вам нужно вручную включить его на всех коммутаторах, на которых хотите его использовать. Как правило, я рекомендую использовать протокол RSTP, когда это возможно. Но если ваша организация не допускает этого или если вам он просто неудобен, нет никаких проблем в том, чтобы придерживаться обычного протокола STP.

Практикум

Включите протокол RSTP на Коммутаторах 1 и 2, выполнив следующую команду конфигурации:

```
spanning-tree mode rapid-pvst
```

Проверьте результат, выполнив команду `Show spanning-tree vlan 700`.

Вы должны увидеть следующий, уже знакомый результат:

```
Switch2#sh spanning-tree vlan 700
```

Обратите внимание, что вывод почти не изменился:

```
VLAN0700
  Spanning Tree enabled protocol rstp  ◄— Протокол RSTP включен
  Root ID     Priority    33468
              Address     0023.ab40.8e00
              Cost        19
              Port        25 (FastEthernet0/23)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    33468  (priority 32768 sys-id-ext 700)
              Address     0024.5088.6d80
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----
Fa0/21              Desg FWD 19        128.23   P2p
Fa0/23              Root FWD 19        128.25   P2p
Fa0/24              Altn BLK 19        128.26   P2p
```

Самое большое различие в деталях: STP задействовал протокол RSTP – Rapid Spanning Tree Protocol. Чтобы действительно оценить разницу, вам нужно посмотреть, как он себя ведет при сбое активного соединения.

Практикум

Выключите интерфейс FastEthernet0/23 на Коммутаторе 2:

```
Int fa0/23
Shut
```

Выполните команду `show spanning-tree vlan 700`.

Безо всякой заметной задержки вы увидите порт FastEthernet0/24 в состоянии передачи:

```
Switch2#sh spanning-tree vlan 700

VLAN0700
  Spanning Tree enabled protocol rstp
  Root ID    Priority    33468
             Address     0023.ab40.8e00
             Cost        19
             Port        26 (FastEthernet0/24)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    33468  (priority 32768 sys-id-ext 700)
             Address     0024.5088.6d80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----
Fa0/21              Desg FWD 19        128.23   P2p
Fa0/24              Root FWD 19        128.26   P2p
```

Если вы опять включите порт FastEthernet0/23, то увидите, как быстро среагирует протокол RSTP.

Практикум

Включите интерфейс FastEthernet0/23 снова:

```
Int fa0/23
No shut
```

Как можно быстрее выполните команду `show spanning-tree vlan 700`.

Вы увидите, что ситуация опять нормализовалась:

```
Switch2#sh spanning-tree vlan 700

VLAN0700
  Spanning Tree enabled protocol rstp
  Root ID    Priority    33468
             Address     0023.ab40.8e00
             Cost        19
```

```
            Port        25 (FastEthernet0/23)
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    33468  (priority 32768 sys-id-ext 700)
             Address     0024.5088.6d80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----
Fa0/21              Desg FWD 19        128.23   P2p
Fa0/23              Root FWD 19        128.25   P2p
Fa0/24              Altn BLK 19        128.26   P2p
```

Порт FastEthernet0/23 снова находится в состоянии передачи. Обратите внимание, что протокол STP предпочитает использовать порт FastEthernet0/23, если тот доступен. По определению все версии протокола STP – RSTP и т. д. – выбирают порт с наименьшим номером.

Вы можете подумать, что это все еще не совсем оптимальная конфигурация. У вас есть резервные соединения, но за счет использования дополнительных портов на каждом коммутаторе. Было бы хорошо использовать оба соединения одновременно, не создавая петель коммутации. В следующей главе я покажу вам, как этого достичь.

12.3. Режим PortFast

Так как протокол STP старается не допустить петель коммутации в сети, он работает со всеми портами, а не только с подсоединенными к коммутаторам. Это производит интересный эффект на конечные устройства, такие как компьютеры или IP-телефоны. В структуре организации довольно часто пользователь перезагружает компьютер и не может войти в сеть около 30 секунд. Перезагрузка компьютера (так же как его выключение и включение) приводит к сбросу сетевой карты, что для коммутатора выглядит так, как если бы кто-то выдернул и подключил обратно сетевой кабель.

Практикум

С компьютера Начальство-ПК1 непрерывно пингуйте SVI-интерфейс виртуальной сети 700:

```
Ping 172.31.70.254 -t
```

На Коммутаторе 2 выключите и включите порт FastEthernet0/21, к которому подключен компьютер Начальство-ПК1:

```
Int fa0/21
Shut
No shut
```

На компьютере Начальство-ПК1 вы должны увидеть следующий вывод:

```
PS C:\Users\Administrator> ping 172.31.70.254 -t

Pinging 172.31.70.254 with 32 bytes of data:
Reply from 172.31.70.254: bytes=32 time<1ms TTL=255
Request timed out.
Request timed out.
Request timed out.
Reply from 172.31.70.50: Destination host unreachable.
Reply from 172.31.70.50: Destination host unreachable.
Reply from 172.31.70.50: Destination host unreachable.
Request timed out.
Reply from 172.31.70.50: Destination host unreachable.
Request timed out.
Reply from 172.31.70.50: Destination host unreachable.
Reply from 172.31.70.254: bytes=32 time=2ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
```

Виден единственный успешный пинг и за ним ряд неудач, после которых опять следуют успешные пинги. Протокол STP не знает, какое устройство подсоединено к этому порту, поэтому выжидает длительное время (30 секунд), чтобы убедиться, что трафик через этот порт не приведет к петле коммутации.

Как бы ни был умен протокол STP, это не человек, и он недостаточно сообразителен, чтобы быстро понять, что единственное присоединенное к данному порту устройство – это компьютер. К счастью, вы, сообразительный сетевой администратор, можете приказать протоколу STP не ждать, чтобы немедленно перевести порт в состояние передачи.

Коммутаторы Cisco обладают некоей особенностью в их реализации протокола STP, называемой *PortFast*. Это вполне соответствующее название, т. к. *PortFast* действительно обеспечивает связку порт–скорость.

Практикум

Включите режим PortFast на порту FastEthernet0/21:

```
Int fa0/21
Spanning-tree portfast
```

Вы должны получить страшное предупреждение:

```
Switch2(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single host. Connecting hubs,
concentrators, switches, bridges, etc... to this interface when portfast is enabled, can cause
temporary bridging loops.
Use with CAUTION

%Portfast has been configured on FastEthernet0/21 but will only have effect when the interface
is in a non-trunking mode.
```

В переводе на нормальный язык это означает, что вы никогда не должны включать режим PortFast на интерфейсах, соединенных с коммутаторами. Если это сделать, вы запустите петлю коммутации, т. к. протокол STP на этом порту деактивируется.

Кстати говоря, включение режима PortFast на интерфейсе одинаково действует на протоколы STP и RSTP. Теперь, когда вы включили режим PortFast, время протестировать его.

Практикум

С компьютера Начальство-ПК1 непрерывно пингуйте SVI-интерфейс виртуальной сети 700:

```
Ping 172.31.70.254 –t
```

На Коммутаторе 2 выключите и включите порт FastEthernet0/21, к которому подключен компьютер Начальство-ПК1:

```
Int fa0/21
Shut
No shut
```

Вы должны увидеть, что неудачей закончилось не более одного пинга:

```
PS C:\Users\Administrator> ping 172.31.70.254 -t

Pinging 172.31.70.254 with 32 bytes of data:
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
Request timed out.
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
Reply from 172.31.70.254: bytes=32 time<1ms TTL=255
```

Большинство пользователей даже не заметит этого мимолетного пропадания связи с сетью. Хотя вам, разумеется, не нужно включать режим PortFast на портах, подключенных к устройствам конечного пользователя, это может потенциально избавить вас от случайных проблем пользователя, который перезагрузил устройство или случайно задел сетевой кабель.

12.4. Команды, использованные в этой главе

Хотя вы и не изучили огромного количества команд в этой главе, те несколько, которые вы узнали, очень мощные, поэтому важно точно знать, что они позволяют сделать. Обратитесь к табл. 12.1, как к памятке.

Таблица 12.1. Команды, использованные в этой главе

Команда	В режиме конфигурации	Описание
`show spanning-tree vlan 700`	–	Выводит информацию о текущей конфигурации протокола (R)STP
`spanning-tree mode rapid-pvst`	Глобальный	Глобально включает протокол RSTP
`spanning-tree portfast`	Интерфейс	Включает режим PortFast, что заставляет протокол (R)STP поместить порт в состояние передачи немедленно

Напомню, обычный протокол STP включен по умолчанию. Он стабилен и прекрасно работает, но при изменении топологии сети (например, если соединение между коммутаторами нарушено) ему требуется около 30 секунд, чтобы возобновить связь. Протокол RSTP, как следует из названия (Rapid – быстрый), не требует так много времени, поэтому я рекомендую использовать его, когда это возможно.

12.5. Практическое задание

1. Включите режим PortFast на порту, к которому подсоединен компьютер Кадры-ПК1.
2. Отключите Ethernet-кабель, подсоединенный к одному из портов FastEthernet0/23, не важно, от какого. Можно ли с компьютера Начальство-ПК1 пропинговать адрес `172.31.70.254`?
3. Физически отсоедините кабель, подключенный к порту FastEthernet0/24. Можно ли все еще с компьютера Начальство-ПК1 пропинговать адрес `172.31.70.254`?
4. Подсоедините оба кабеля и сохраните ваши конфигурации!

Глава 13

Оптимизация сети с использованием каналов порта

Напомню, в предыдущей главе вы использовали два соединения между Коммутаторами 1 и 2, а протокол остовного дерева (STP) допускал трафик только по одному из этих соединений. Это позволяет предотвратить петли коммутации, но ограничивается в том, что порты не используются полностью. Таким образом, мы не можем пользоваться всей возможной полосой пропускания задействованных портов.

Технология *агрегации каналов*, также известная как *EtherChannel*, позволяет обойти это ограничение. Она позволяет трафику проходить одновременно по обоим соединениям, не создавая петель коммуникации.

По мере чтения этой главы вы узнаете, как настроить агрегацию каналов, чтобы реализовать конфигурацию, показанную на рис. 13.1.

Обратите внимание, что полученный в итоге агрегированный канал состоит из соединений FastEthernet0/23 и 0/24 и называется Port-channel1. Когда вы настраиваете агрегированный канал на коммутаторе, коммутатор создает виртуальный или логический интерфейс для представления агрегированного канала.

Следуя традиции Cisco, агрегированный канал может называться также *EtherChannel*, *Port channel*, *bundle* или *port group*. В основном я буду использовать термин *агрегированный канал*, но в командах конфигурации и выводе вы увидите и другие термины.

По мере погружения в администрирование сетей Cisco вы сможете заметить, что существует несколько путей достижения желаемой цели. Какой конкретный метод вы используете, зависит не столько от результата, которого вы хотите достичь, сколько от того, сколько времени потребуется и насколько он будет масштабироваться. Конфигурация агрегированного канала ничем не отличается.

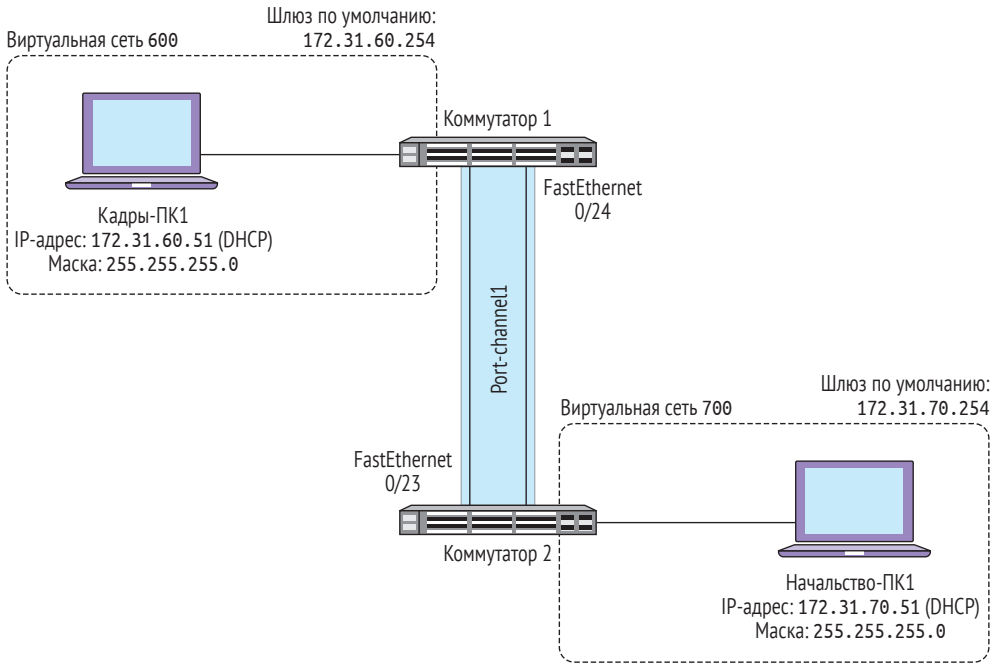

Рис. 13.1 ❖ Агрегированный канал между Коммутаторами 1 и 2

Вспомните главу 10, в которой вы настраивали транки виртуальной сети между вашими коммутаторами. Протокол Dynamic Trunking Protocol (DTP), который включен по умолчанию на коммутаторах Catalyst, позволил вам настроить магистраль путем конфигурации коммутатора только на одном конце соединения.

Вы не можете поступить так с агрегированными каналами. В отличие от транков виртуальной сети, единственный способ получить работающий агрегированный канал – настроить оба коммутатора.

13.1. Статический или динамический агрегированный канал?

Прежде чем настраивать агрегированный канал, вам нужно решить, каким он должен быть. При настройке агрегированного канала между коммутаторами Cisco у вас есть два варианта: статическая или динамическая конфигурация.

13.1.1. Статический агрегированный канал

Статический агрегированный канал аналогичен безусловному транковому порту виртуальной сети. Коммутатор, настроенный с безусловным транковым

портом, не учитывает, как настроен коммутатор на другом конце. Точно так же статический агрегированный канал связывает два или более физических портов в одном логическом интерфейсе агрегированного канала, независимо от того, чем занимается коммутатор на другом конце.

Существенный недостаток этого метода заключается в том, что он создает огромный потенциал для петель коммутации и потери трафика либо его попадания в «черную дыру». Я расскажу об этом позже в данной главе, но когда вы настраиваете агрегированный канал, он скрывает физические интерфейсы от протокола STP, так что STP больше не может блокировать какие-либо отдельные интерфейсы.

Преимущество создания статического агрегированного канала заключается в том, что быстрый взгляд на текущую конфигурацию дает совершенно ясное представление о том, что есть агрегированный канал, а также какие физические порты в него включены.

13.1.2. Динамический агрегированный канал

Динамический агрегированный канал несколько аналогичен согласованному транку виртуальной сети с использованием протокола DTP. Но есть два существенных отличия.

Первое отличие в том, что есть два протокола, с помощью которых можно установить агрегированный канал: *Link Aggregation Control Protocol (LACP)* и *Port Aggregation Protocol (PAgP)*. В этой главе я буду рассматривать только LACP, т. к. он используется наиболее часто.

Второе отличие состоит в том, что протокол DTP включен по умолчанию, а протоколы LACP или PAgP – нет. Вам все еще нужно настраивать оба коммутатора и указывать, какой протокол вы хотите использовать.

Большое преимущество протоколов LACP или PAgP – в том, что они выполняют проверку работоспособности, чтобы гарантировать, что агрегированный канал будет работать должным образом, прежде чем создавать его. Это похоже на то, как протокол DTP выполняет собственную проверку перед созданием транка виртуальной сети. Обеспечивая проверку работоспособности, эти динамические протоколы гарантируют, что ваш агрегированный канал не будет черной дырой для трафика и не создаст петель коммутации.

Относительно небольшой недостаток динамических протоколов заключается в том, что они влекут за собой небольшую начальную задержку, о которой вы сможете узнать в тот момент, когда настраиваете агрегированный канал с использованием протокола LACP.

13.2 Настройка динамического агрегированного канала с помощью протокола LACP

Создание динамического агрегированного канала с использованием протокола LACP – наиболее распространенный метод, поэтому вы начнете с него. Он

настолько распространен, что для многих термины *LACP* и *агрегированный канал* являются синонимами. Понимая тонкое различие между агрегированным каналом и одним из протоколов, используемых для его согласования – LACP, – вы намного опередите остальных администраторов.

Повторяю, Link Aggregation Control Protocol (протокол управления агрегацией каналов) – это только протокол. Вы настраиваете его на двух коммутаторах, и он работает, чтобы установить агрегированный канал между этими коммутаторами. Простая настройка протокола LACP не гарантирует, что вы получите рабочий агрегированный канал.

Практикум

Выполните следующие команды на Коммутаторах 1 и 2:

```
interface range fa0/23-24
channel-group 1 mode active
```

Ключевое слово `active` означает, что вы используете протокол LACP вместо PAgP или статического агрегированного канала.

Среди множества выходных данных вы должны увидеть следующее на обоих коммутаторах:

```
*Mar  2 04:26:38.669: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
*Mar  2 04:26:39.676: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel1, changed
state to up
```

Обратите внимание, что интерфейс Port-channel1 не появляется до тех пор, пока вы не выполните команды на обоих коммутаторах. Это связано с тем, что все, что вы сделали, – это настроить протокол LACP, чтобы попытаться согласовать агрегированный канал. Чтобы протокол LACP смог это сделать, вам нужно настроить и Коммутатор 1, и Коммутатор 2 так, чтобы они общались друг с другом с помощью протокола LACP. Прежде чем один из коммутаторов создаст интерфейс Port-channel1, оба коммутатора должны явно договориться друг с другом через протокол LACP насчет создания агрегированного канала.

Как только агрегированный канал появится, вы можете проверить, какие конкретные порты относятся к нему.

Практикум

На Коммутаторах 1 и 2 выполните команду `show etherchannel summary`.

Вы должны увидеть следующее:

```
Switch2#show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
```

```
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+--------------------
1      Po1(SU)       LACP        Fa0/23(P) Fa0/24(P)
```

Обратите внимание, что используется протокол LACP. Оба физических порта, FastEthernet0/23 и 0/24, являются частью агрегированного канала.

Другой надежный способ проверки наличия агрегированного канала заключается в проверке протокола STP.

Практикум

На Коммутаторах 1 и 2 выполните команду `show spanning-tree vlan 700`.

Вывод будет слегка отличаться от того, что вы видели ранее:

```
Switch2#show spanning-tree vlan 700

VLAN0700
  Spanning Tree enabled protocol ieee
  Root ID    Priority    33468
             Address     0023.ab40.8e00
             Cost        19
             Port        64 (Port-channel1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    33468  (priority 32768 sys-id-ext 700)
             Address     0024.5088.6d80
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- -----
Fa0/21              Desg FWD 19        128.23   P2p Edge
Po1                 Root FWD 12        128.64   P2p
```

Обратите внимание, что порты FastEthernet0/23 и 0/24 здесь не отображаются. Отображается только значение Po1– сокращение для Port-channel1. Как я уже говорил ранее, агрегированный канал скрывает физические порты от протокола STP.

Если эта конфигурация кажется очень простой, то это потому, что она такая и есть. Протокол LACP делает всю грязную работу, чтобы убедиться, что все

в порядке, прежде чем пытаться установить агрегированный канал. Чтобы по-настоящему оценить метод, сравните его со статической конфигурацией. Но прежде чем вы сможете это сделать, вам нужно вернуть все обратно, удалив существующий агрегированный канал.

Практикум

На Коммутаторах 1 и 2 выполните следующую команду, чтобы удалить интерфейс Port-channel1:

```
no interface port-channel 1
```

Эта команда может показаться немного странной, поскольку она удаляет интерфейс. Но помните, что Port-channel1 является (или был) *виртуальным* интерфейсом. Это абстракция двух физических интерфейсов: FastEthernet0/23 и 0/24.

Когда вы удаляете интерфейс агрегированного канала, система IOS делает то, чего вы, вероятно, не ожидали. Она отключает эти интерфейсы!

Практикум

На Коммутаторах 1 и 2 просмотрите текущую конфигурацию интерфейсов FastEthernet0/23 и 0/24:

```
Show run int fa0/23
Show run int fa0/24
```

Вы должны увидеть следующее:

```
Switch2#show run int fa0/23
interface FastEthernet0/23
 shutdown
end

Switch2#show run int fa0/24
interface FastEthernet0/24
 shutdown
end
```

Само собой разумеется, что агрегированный канал после отключения обоих интерфейсов не будет работать. Чтобы убедиться, что агрегированный канал действительно удален, вам нужно снова включить физические интерфейсы.

Практикум

На Коммутаторах 1 и 2 снова включите интерфейсы FastEthernet0/23 и 0/24:

```
Int range fa0/23-24
No shutdown
```

Агрегированный канал больше не должен отображаться:

```
Switch2#show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 0
Number of aggregators:           0

Group  Port-channel  Protocol    Ports
------+-------------+-----------+--------------------
```

Теперь, когда агрегированный канал полностью удален, вы готовы его настроить в статическом режиме.

13.3. Создание статического агрегированного канала

Настройка статического агрегированного канала столь же проста, что и использование протокола LACP, но он не предоставляет вам те же проверки и защиту, что LACP. Учитывая это, вы можете задать вопрос, почему кто-то даже задумывается о создании статического агрегированного канала.

Как правило, есть две основные причины. Во-первых, некоторым организациям не нравится идея использования динамических протоколов. По возможности они избегают протоколов DTP, LACP и всего остального, что может динамически изменять сеть без прямого вмешательства человека. Вторая причина заключается в том, что если вам нужно создать канал между коммутатором Cisco и более старым коммутатором, который не поддерживает протокола LACP или PAgP, то единственным способом остается статический агрегированный канал.

Если вы хотите (или должны) использовать статический маршрут, функция *EtherChannel Misconfiguration Guard* обеспечивает некоторую защиту от петель коммутации, вызванных неправильно настроенным агрегированным каналом. Она включена по умолчанию.

Практикум

Убедитесь, что функция EtherChannel Misconfiguration Guard включена на Коммутаторах 1 и 2:

```
spanning-tree etherchannel guard misconfig
```

Настройте статический агрегированный канал:

```
interface range fa0/23-24
channel-group 1 mode on
```

Проверьте результат, выполнив команду `show etherchannel summary`.

Вы должны получить знакомый вывод:

```
Switch2#show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+--------------------
1      Po1(SU)          -        Fa0/23(P) Fa0/24(P)
```

Он почти идентичен тому, что вы видели раньше, но на этот раз колонка `Protocol` пуста, потому что вы не используете протокола LACP/PAgP. Агрегированный канал, однако, все еще работает, и интерфейсы FastEthernet0/23 и 0/24 являются его членами.

Это может показаться совершенным. После моего ужасного предупреждения о возможности возникновения петель коммутации пока все, кажется, идет плавно. Так что ради интереса давайте что-нибудь сломаем!

Практикум

На Коммутаторе 1 удалите агрегированный канал:

```
No interface po1
```

Помните, что при удалении интерфейса агрегированного канала система IOS отключает физические интерфейсы портов FastEthernet0/23 и 0/24. Включите их снова:

```
Int range fa0/23-24
No shut
```

На Коммутаторе 1 вы не получите никаких результатов, указывающих на проблему. И на Коммутаторе 2 сначала ничего очевидного не произойдет. Но

примерно через минуту вы увидите ошибку, указывающую на возникновение петли коммутации:

```
*Mar 1 00:50:03.809: %SW_MATM-4-MACFLAP_NOTIF: Host 2c27.d737.9ad1 in vlan700 is flapping
between port Fa0/21 and port Po1
```

Почти через минуту функция EtherChannel Misconfiguration Guard запустится и выключит агрегированный канал:

```
*Mar 1 00:50:31.265: %PM-4-ERR_DISABLE: channel-misconfig (STP) error detected on Po1, putting
Po1 in err-disable state
```

Сообщение `channel-misconfig (STP) error detected` указывает, что функция Guard Misconfiguration Guard Ether-Channel обнаружила петлю коммутации и отключила интерфейс агрегированного канала.

Для определения более четких различий между созданием статического агрегированного канала и динамического с использованием LACP обратите внимание на то, что функции EtherChannel Misconfiguration Guard потребовалось около двух минут, чтобы отключить агрегированный канал. В реальной сети это могло привести к двум минутам простоя или, по крайней мере, к очень низкой производительности сети. Протокол LACP, с другой стороны, обнаружил бы проблему и немедленно отключил агрегированный канал.

Чтобы восстановить агрегированный канал, вам нужно заново создать его на Коммутаторе 1 и вывести порты FastEthernet0/23 и 0/24 из состояния `err-disable` (отключен из-за ошибки) на Коммутаторе 2.

Практикум

На Коммутаторе 1 выполните следующие команды конфигурации, чтобы пересоздать агрегированный канал:

```
interface range fa0/23-24
channel-group 1 mode on
```

На Коммутаторе 2 перегрузите интерфейсы FastEthernet0/23 и 0/24, чтобы вывести их из состояния `err-disable`:

```
Int po1
Shutdown
No shutdown
```

Порты должны заработать на обоих коммутаторах.

13.4. Методы балансировки нагрузки

В начале главы я сказал, что агрегированный канал позволяет трафику проходить одновременно по обоим каналам. Только что настроенный агрегированный канал позволит трафику одновременно передаваться через порты FastEthernet0/23 и 0/24. Но способы, которыми трафик проходит по этим соединениям, ничем не отличаются.

Чтобы понять, как агрегированный канал решает, через какое конкретное соединение отправлять трафик, взгляните на рис. 13.2.

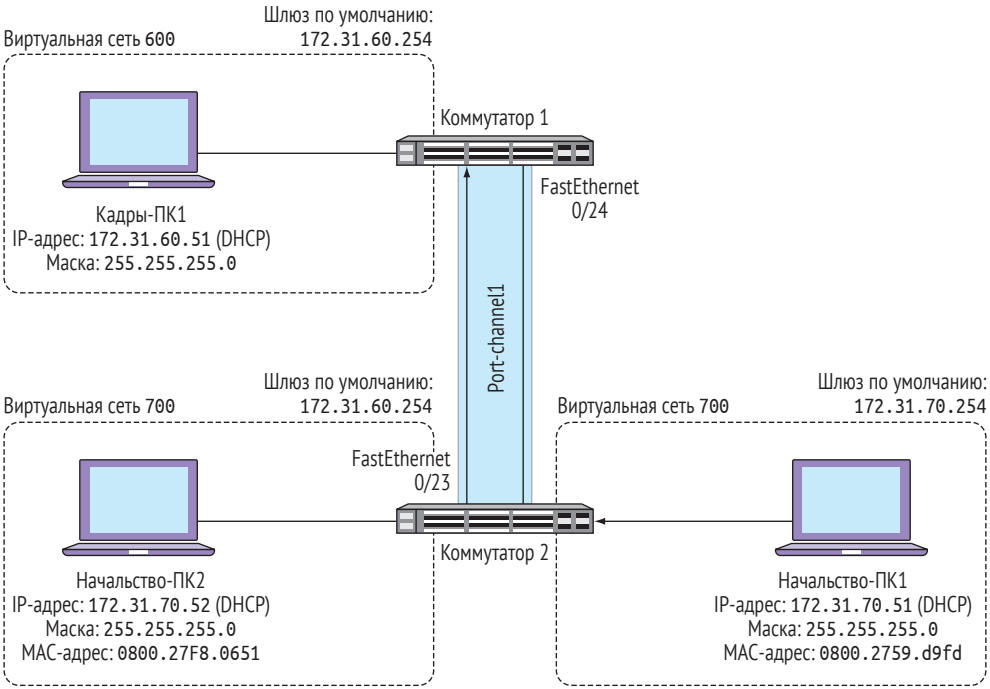

Рис. 13.2 ❖ По умолчанию метод балансировки нагрузки агрегированного канала отправляет трафик с компьютера Начальство-ПК1 через интерфейс FastEthernet0/23

Обратите внимание, что я добавил дополнительный компьютер в виртуальную сеть 700. Не беспокойтесь о добавлении еще одного компьютера в свою топологию. Цель здесь состоит в том, чтобы просто показать вам, как агрегированные каналы пересылают трафик.

По умолчанию все порты передают трафик на основе MAC-адреса источника кадра. Обратите внимание, что трафик от компьютера Начальство-ПК1 с исходным MAC-адресом 0800.2759.d9fd идет через порт FastEthernet0/23. Вам не нужно понимать, какой алгоритм использует система IOS для принятия этого решения, но вам нужно знать, как определить, через какой порт она отправит трафик от заданного конкретного MAC-адреса источника.

Практикум

На Коммутаторе 2 выполните следующую команду, чтобы определить, по какому физическому порту будет проходить трафик от компьютера Начальство-ПК1 (0800.2759.d9fd):

```
test etherchannel load-balance interface port-channel 1 mac
0800.2759.d9fd ffff.ffff.ffff
```

Вы должны получить следующий результат:

```
Switch2#test etherchannel load-balance interface port-channel 1 mac
     0800.2759.d9fd ffff.ffff.ffff
Would select Fa0/23 of Po1
```

Вывод показывает, что трафик от компьютера Начальство-ПК1 будет выходить через порт FastEthernet0/23 *каждый раз*. Может показаться, что это противоречит самой цели агрегированного канала, который должен позволить вам использовать пропускную способность обоих каналов одновременно. Но при ближайшем рассмотрении можно увидеть, что агрегированный канал использует оба соединения.

Практикум

На Коммутаторе 2 выполните следующую команду, чтобы определить, по какому физическому порту будет проходить трафик от компьютера Начальство-ПК2 (0800.27F8.0651):

```
test etherchannel load-balance interface port-channel 1 mac
0800.27F8.0651 ffff.ffff.ffff
```

Вы должны получить следующий результат:

```
Switch2#test etherchannel load-balance interface port-channel 1 mac
     0800.27F8.0651 ffff.ffff.ffff
Would select Fa0/24 of Po1
```

Трафик от компьютера Начальство-ПК2 будет проходить по другому соединению, FastEthernet0/24, как показано на рис. 13.3.

Именно так агрегированный канал использует оба соединения одновременно: путем балансировки нагрузки (или, точнее, распределения нагрузки) на основе MAC-адреса источника.

Как правило, вы не будете изменять метод балансировки нагрузки по умолчанию, хотя и можете. Если вы когда-либо столкнетесь с каким-либо неожиданным поведением или получите неожиданный вывод, можно проверить настроенный метод балансировки нагрузки.

Практикум

На Коммутаторах 1 и 2 проверьте текущий метод балансировки нагрузки, выполнив следующую команду:

```
Show etherchannel load-balance
```

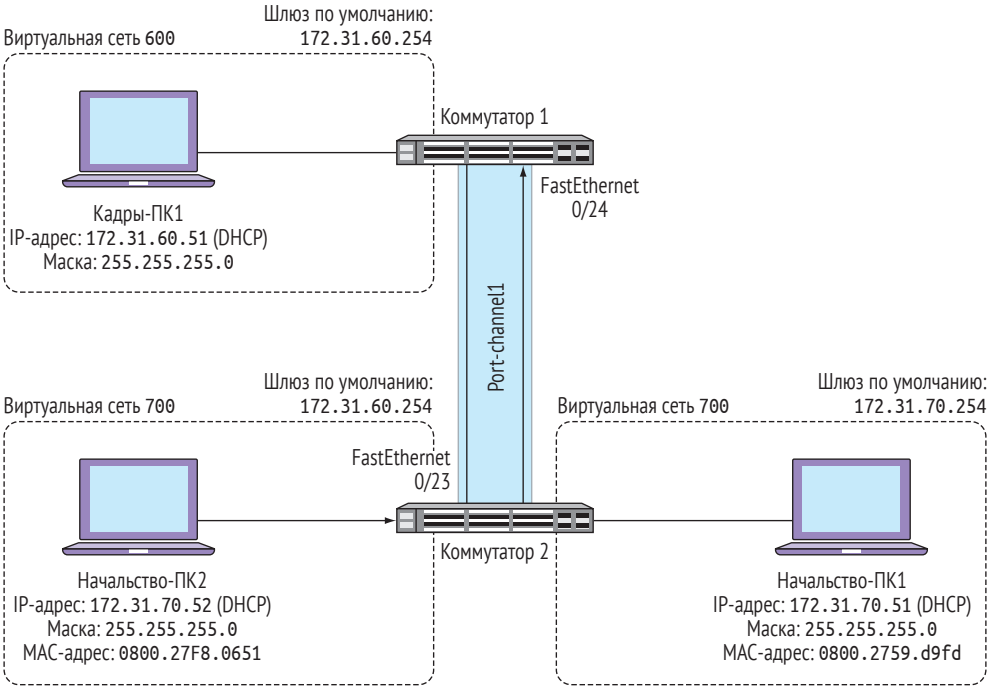

Рис. 13.3 ❖ По умолчанию метод балансировки нагрузки агрегированного канала отправляет трафик от компьютера Начальство-ПК2 по соединению FastEthernet0/24

Вы должны увидеть следующее:

```
Switch2#show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-mac

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source MAC address
  IPv4: Source MAC address
  IPv6: Source MAC address
```

Одинокое значение `src-mac` в выводе является сокращением от «source MAC – исходный MAC-адрес» – текущего настроенного метода балансировки нагрузки. Имейте в виду, что методы не обязательно должны совпадать на обоих коммутаторах. Несовпадение методов балансировки нагрузки не обязательно вызовет какие-либо проблемы, но может, в зависимости от особенностей сети. В худшем случае вы не сможете использовать полную пропускную способность агрегированного канала, а производительность сети снизится.

Опять же, хочу подчеркнуть, что при обычных обстоятельствах вы не должны менять метод балансировки нагрузки. Но я также признаю, что кто-то другой может его изменить, поэтому вам нужно знать, как изменить его на значение по умолчанию.

Практикум

На Коммутаторе 1 установите метод балансировки нагрузки `src-mac`:

```
port-channel load-balance src-mac
```

Эта команда может вас удивить, потому что в нее не включены ключевые слова `etherchannel` или `channel-group`. Обратитесь к встроенной справочной системе, которая поможет вам, если вы не можете вспомнить синтаксис команды.

13.5. Команды в этой главе

Обратитесь к табл. 13.1, чтобы вспомнить команды этой главы.

Таблица 13.1. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`interface range fa0/23-24`	Глобальный	Выбирает порты FastEthernet0/23 и 0/24 для конфигурации
`channel-group 1 mode active`	Интерфейс	Включает протокол LACP на выбранном порту (портах)
`show etherchannel summary`	–	Отображает информацию о настроенных агрегированных каналах
`no interface port-channel 1`	Глобальный	Удаляет интерфейс Port-channel1
`spanning-tree etherchannel guard misconfig`	Глобальный	Включает функцию EtherChannel Misconfiguration Guard
`channel-group 1 mode on`	Интерфейс	Создает статический агрегированный канал, используя выбранный(е) порт(ы) в качестве членов
`test etherchannel load-balance interface port-channel 1 mac 0800.2759.d9fd ffff.ffff.ffff`	–	Отображает, по какому интерфейсу будет выходить трафик с MAC-адреса `0800.2759.d9fd` на широковещательный адрес
`show etherchannel load-balance`	–	Отображает настроенный метод балансировки нагрузки
`port-channel load-balance src-mac`	Глобальный	Устанавливает метод балансировки нагрузки `src-mac`

13.6. Практическое задание

Сейчас вы снова создадите агрегированный канал с помощью протокола LACP, но на этот раз добавите дополнительное соединение. Проделайте следующие шаги, чтобы выполнить задание.

1. Удалите существующий агрегированный канал на Коммутаторах 1 и 2.
2. Соедините порты FastEthernet0/22 обоих коммутаторов.
3. Используйте протокол LACP для создания агрегированного канала, состоящего из интерфейсов FastEthernet0/22, 0/23 и 0/24.

Глава 14

Обеспечение масштабируемости сети путем совместного использования маршрутизаторов и коммутаторов

На данный момент Коммутатор 1 маршрутизирует IP-трафик между двумя подсетями: `172.31.60.0/24` в виртуальной сети `600` и `172.31.70.0/24` в виртуальной сети `700`. Это обычная и совершенно приемлемая конфигурация. Но возникают ситуации, когда нужно маршрутизировать трафик между виртуальными сетями с помощью дополнительного устройства – *маршрутизатора*.

Вы можете столкнуться с сетью, в которой нет маршрутизаторов или коммутаторов уровня 3. Начнем с того, что маловероятно, что в такой сети будут использоваться виртуальные сети, потому что невозможен способ маршрутизации между ними. Следовательно, если организации нужны виртуальные сети, но все, что у них есть, – это коммутаторы уровня 2 и маршрутизатор, то единственный способ маршрутизации трафика между виртуальными сетями заключается в использовании маршрутизатора.

Этот специфический вариант использования маршрутизатора встречается все реже, поскольку коммутаторы уровня 3 получают все большее распространение. Но организации всех размеров широко используют маршрутизаторы для других важных функций, включая сетевое взаимодействие между офисами и подключение IP-телефонов к коммутируемой телефонной сети общего

пользования (PSTN). Вот почему сетевому администратору важно иметь четкое и правильное представление о том, как работают эти устройства.

В данной главе я подробно расскажу вам о подключении и настройке маршрутизаторов для выполнения *маршрутизации между виртуальными сетями*. Позже в этой книге вы узнаете о более продвинутых темах, включая маршрутизацию по *глобальной сети* (Wide area network, WAN) и настройку протоколов динамической маршрутизации.

Ниже представлены базовые шаги, которые вы будете выполнять в этой главе.

1. Подключение маршрутизатора к Коммутатору 1.
2. Создание транка виртуальной сети между Коммутатором 1 и Маршрутизатором 1.
3. Настройка Маршрутизатора 1 для маршрутизации между виртуальными сетями.

14.1. Конфигурация «маршрутизатор-на-палочке»

В этой главе вы собираетесь создать то, что сетевые администраторы называют *конфигурацией «маршрутизатора-на-палочке» (router-on-a-stick)*. Звучит как вкусное лакомство. Но, как вы можете видеть на рис. 14.1, это всего лишь описание способа физического подключения маршрутизатора к коммутатору.

Маршрутизатор 1 физически соединен с Коммутатором 1 с помощью одного Ethernet-кабеля. Это единственное соединение и есть *палочка* в названии «*маршрутизатор-на-палочке*». Также обратите внимание, что между Коммутатором 1 и Маршрутизатором 1 настроен транк виртуальной сети. Этот транк позволяет трафику как из виртуальной сети `600`, так и из виртуальной сети `700` достичь маршрутизатора, так что он может обеспечить маршрутизацию между виртуальными сетями.

Дополнительно

Вы можете услышать, что кто-то говорит о физическом Ethernet-соединении между устройствами как соединении уровня 1, ссылаясь на *физический* уровень модели OSI (Open Systems Interconnect). Транк виртуальной сети работает на уровне 2 – *канальном* уровне передачи данных (Data Link Layer), т. е. там же, где функционируют MAC-адреса и протокол определения адреса (ARP). Я не собираюсь подробно описывать модель OSI в этой книге, но предлагаю обратиться к веб-сайту www.manning.com/books/learn-cisco-network-administration-in-a-month-of-lunches для получения дополнительной информации.

Напомню, что Коммутатор 1 имеет два коммутируемых виртуальных интерфейса (SVI-интерфейса): один – для виртуальной сети `600` с IP-адресом `172.31.60.254/24` и другой – для виртуальной сети `700` с IP-адресом `172.31.70.254/24`. Поскольку Маршрутизатор 1 эффективно заменит Коммутатор 1 в качестве

устройства, которое выполняет маршрутизацию между виртуальными сетями 600 и 700, вам необходимо удалить эти SVI-интерфейсы с Коммутатора 1.

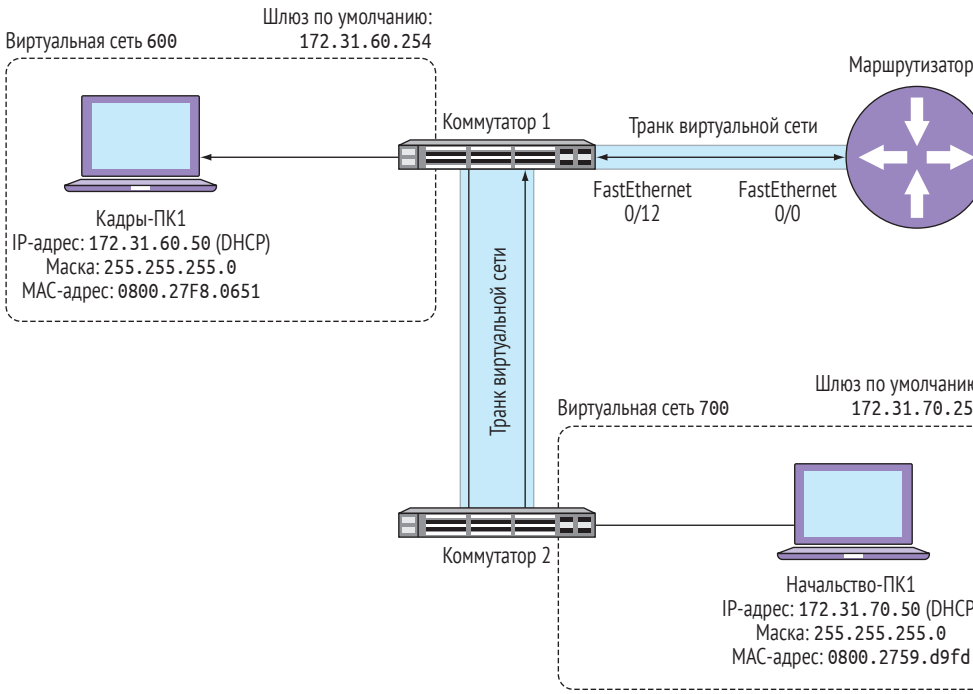

Рис. 14.1 ❖ Конфигурация «маршрутизатор-на-палочке»

Практикум

На Коммутаторе 1 удалите SVI-интерфейсы виртуальных сетей 600 и 700:

```
no int vlan 600
no int vlan 700
```

Эта операция не только удаляет виртуальные интерфейсы; также при этом соответствующие IP-адреса прекращают свое существование. Напомню, что компьютер Начальство-ПК1 настроен на использование адреса 172.31.70.254 в качестве шлюза по умолчанию, тогда как на компьютере Кадры-ПК1 используется шлюз по умолчанию 172.31.60.254. Но как только вы удаляете SVI-интерфейсы виртуальных сетей 600 и 700, эти IP-адреса перестают существовать. Результат для сети заключается в том, что устройства в виртуальных сетях 600 и 700 больше не могут взаимодействовать друг с другом. Чтобы исправить это, вы будете подключать и настраивать Маршрутизатор 1 для выполнения той работы, которую до этого момента выполнял Коммутатор 1.

14.2. Подключение Маршрутизатора 1

Вы уже должны были настроить Маршрутизатор 1 в соответствии с инструкциями на сайте книги. Если вы это сделали, то интерфейс FastEthernet0/0 Маршрутизатора 1 будет иметь IP-адрес `192.168.1.201/24`. Вы будете использовать этот IP-адрес для авторизации на Маршрутизаторе 1, чтобы настроить его (чуть позже).

Практикум

Подключите порт FastEthernet0/0 Маршрутизатора 1 к порту FastEthernet0/12 Коммутатора 1. Подайте питание на Маршрутизатор 1.

Пока вы ожидаете загрузки Маршрутизатора 1, настройте интерфейс FastEthernet0/12 на Коммутаторе 1 в качестве транкового порта 802.1Q:

```
Interface f0/12
Switchport trunk encapsulation dot1q
Switchport mode trunk
```

Повторюсь, интерфейс FastEthernet0/12 должен быть транковым портом, чтобы трафик виртуальных сетей `600` и `700` мог проходить по одному физическому каналу, направляясь на Маршрутизатор 1.

Практикум

Как только Маршрутизатор 1 загрузится, организуйте `telnet`-сеанс на Коммутаторе 1:

```
telnet 192.168.1.201
```

Авторизуйтесь, указав логин `admin` и пароль `cisco`.

Если вы авторизовались в системе успешно, то должны увидеть приглашение `Router1#`. Возможно, вам будет интересно узнать, как можно пользоваться `telnet`-сеансом связи с Коммутатора 1 на Маршрутизатор 1 до настройки транка на Маршрутизаторе 1. Ответ заключается в выводе команды `show interfaces fa0/12 trunk` на Коммутаторе 1:

```
Switch1#show interfaces fa0/12 trunk

Port        Mode          Encapsulation     Status       Native vlan
Fa0/12      on            802.1q            trunking     1
```

Обратите внимание, что параметр `Native vlan` имеет значение `1`. Не вдаваясь в подробности инкапсуляции 802.1Q, это означает, что Маршрутизатор 1 передает трафик виртуальной сети `1`, как если бы интерфейс FastEthernet0 /12 был обычным портом доступа в виртуальной сети `1`. Чтобы понять, почему это важно, взгляните на IP-адрес SVI-интерфейса `VLAN1` Коммутатора 1:

```
Switch1#show ip interface Vlan 1 | i Internet
  Internet address is 192.168.1.101/24   192.168.1.101   YES   NVRAM   up    up
```

Он находится в той же подсети, что и IP-адрес интерфейса FastEthernet0/0 Маршрутизатора 1 (`192.168.1.201/24`). Интерфейс Маршрутизатора 1 фактически аналогичен сетевому интерфейсу на компьютере. Он не транковый и имеет только один IP-адрес.

Практикум

На Маршрутизаторе 1 просмотрите IP-адреса всех интерфейсов:

```
show ip interface brief
```

Вы должны увидеть только один IP-адрес:

```
Router1#show ip interface brief
Interface         IP-Address      OK? Method Status                Protocol
FastEthernet0/0   192.168.1.201   YES NVRAM  up                    up
FastEthernet0/1   unassigned      YES NVRAM  administratively down  down
```

Чего здесь не хватает? Чтобы Маршрутизатор 1 выступал в качестве шлюза по умолчанию для подсетей `172.31.60.0/24` (виртуальная сеть `600`) и `172.31.70.0/24` (виртуальная сеть `700`), ему нужен IP-адрес в каждой из этих подсетей. Но вы не можете назначить три разных IP-адреса для интерфейса FastEthernet0/0. Вы можете использовать интерфейс FastEthernet0/1, но это потребует применения другого порта на Коммутаторе 1. И вместо того чтобы быть «маршрутизатором-на-палочке», Маршрутизатор 1 станет «маршрутизатором-на-двух-палочках». Решение заключается в настройке субинтерфейса на Маршрутизаторе 1 для каждой виртуальной сети и подсети.

14.3. Настройка субинтерфейсов

Концептуально *субинтерфейс* на маршрутизаторе похож на SVI-интерфейс на коммутаторе, поскольку представляет собой виртуальный интерфейс, который находится в одной виртуальной сети и может иметь собственный IP-адрес. Но на этом схожесть заканчивается. В табл. 14.1 перечислены субинтерфейсы, которые вы настроите на Маршрутизаторе 1.

Таблица 14.1. (Суб)интерфейсы Маршрутизатора 1, их IP-адреса и соответствующие виртуальные сети

(Суб)интерфейсы	IP-адреса	Виртуальная сеть
FastEthernet0/0	`192.168.1.201/24`	`1`
FastEthernet0/0.600	`172.31.60.254/24`	`600`
FastEthernet0/0.700	`172.31.70.254/24`	`700`

Обратите внимание, что субинтерфейсы совместно используют имена физических родительских интерфейсов. Имя каждого субинтерфейса должно

быть именем физического родительского интерфейса, за которым следуют точка (.) и уникальный номер от 0 до 4294967295. Важно отметить, что номер субинтерфейса не имеет непосредственного отношения к виртуальной сети (для удобства задается обычно по номеру виртуальной сети). Вместо этого вам нужно указать виртуальную сеть вручную в конфигурации субинтерфейса.

Практикум

Создайте субинтерфейс FastEthernet0/0.600:

```
Interface FastEthernet0/0.600
```

Настройте субинтерфейс как члена виртуальной сети 600:

```
encapsulation dot1Q 600
```

Присвойте ему IP-адрес 172.31.60.254/24...

```
ip address 172.31.60.254 255.255.255.0
```

...и понятное описание:

```
description VLAN 600 subinterface to Switch1 fa0/12
```

Проверьте результат, выполнив команду show interface fa0/0.600.

Вы должны увидеть информацию об IP-адресе и виртуальной сети, согласно выполненным настройкам:

```
Router1#show int fa0/0.600
FastEthernet0/0.600 is up, line protocol is up
  Hardware is Gt96k FE, address is 0015.fa64.76d2 (bia 0015.fa64.76d2)
  Description: VLAN 600 subinterface to Switch1 fa0/12
  Internet address is 172.31.60.254/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec, reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID 600.
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
```

Обратите внимание, что также отображается описание, добавленное в конфигурацию. Описание опционально, но может оказаться полезным, если вы когда-нибудь забудете, для чего предназначен субинтерфейс. Вы также можете использовать ключевое слово description в интерфейсах коммутатора!

Создание субинтерфейса для виртуальной сети 600 и назначение ему IP-адреса обеспечивают для устройств в виртуальной сети 600 способ маршрутизации IP-трафика в разные подсети. Но для выполнения маршрутизации между виртуальными сетями Маршрутизатор 1 также нуждается в субинтерфейсе в виртуальной сети 700 вместе с соответствующим IP-адресом для этой подсети.

Практикум

Настройте субинтерфейс для виртуальной сети 700:

```
interface FastEthernet0/0.700
Encapsulation dot1q 700
Ip address 172.31.70.254 255.255.255.0
Description VLAN 700 subinterface to Switch1 fa0/12
```

Проверьте результат командой show interface fa0/0.700.

Вы должны увидеть следующее:

```
Router1#show int fa0/0.700
FastEthernet0/0.700 is up, line protocol is up
  Hardware is Gt96k FE, address is 0015.fa64.76d2 (bia 0015.fa64.76d2)
  Description: VLAN 700 subinterface to Switch1 fa0/12
  Internet address is 172.31.70.254/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec, reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID 700.
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
```

Следующий шаг – проверка, что на самом деле представляет собой транк виртуальной сети между Маршрутизатором 1 и Коммутатором 1.

Практикум

Выполните следующие команды на Коммутаторе 1, чтобы проверить, что трафик виртуальных сетей 600 и 700 передается через интерфейс FastEthernet0/0:

```
Show vlans 600
Show vlans 700
```

Это еще одна область, где маршрутизаторы и коммутаторы немного отличаются друг от друга. Обратите внимание, что вывод выглядит очень не похоже на то, что вы видите при выполнении команды show vlan на коммутаторе. Вот что должно отобразиться:

```
Router1#show vlans 600

Virtual LAN ID:  600 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface:   FastEthernet0/0.600

   Protocols Configured:   Address:          Received:   Transmitted:
           IP              172.31.60.254          21              0
         Other                                     0              2

Router1#show vlans 700

Virtual LAN ID:  700 (IEEE 802.1Q Encapsulation)

   vLAN Trunk Interface:   FastEthernet0/0.700
```

```
Protocols Configured:    Address:          Received:    Transmitted:
        IP               172.31.70.254           21               0
      Other                                       0              16
```

Я сократил часть вывода для этой книги, но для каждой команды вы должны увидеть идентификатор виртуальной сети, имя субинтерфейса и IP-адрес. Не беспокойтесь, если вы видите значение `0` в колонке `Received` или `Transmitted`. Вам нужно будет произвести некоторый трафик в виртуальной сети, прежде чем эти значения начнут увеличиваться. Напомню, что в главе 8 вы настроили две области адресов DHCP на Коммутаторе 1, как показано в табл. 14.2.

Таблица 14.2. Области адресов, опции и время аренды DHCP для каждой виртуальной сети

VLAN	Подсеть	Маска	Шлюз по умолчанию
600	172.31.60.0	255.255.255.0	172.31.60.254
700	172.31.70.0	255.255.255.0	172.31.70.254

Вам нужно узнать IP-адреса шлюза по умолчанию для каждой подсети. Это те же адреса, которые вы только что присвоили субинтерфейсам на Маршрутизаторе 1. Следовательно, вам нужно уметь определять, что компьютер Начальство-ПК1 использует адрес `172.31.70.254` в качестве шлюза по умолчанию.

Практикум

Выполните команду `ipconfig` на компьютере Начальство-ПК1 (если на нем используется операционная система Windows).

IP-адрес может отличаться, но IP-адрес шлюза по умолчанию должен быть таким же, как показано ниже:

```
PS C:\> ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : benpiper.com
   Link-local IPv6 Address . . . . . : fe80::d8ae:58d6:2dc0:9450%11
   IPv4 Address. . . . . . . . . . . : 172.31.70.50
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.31.70.254
```

Хотя это и не абсолютно необходимо, вы можете сэкономить некоторое время на устранении неполадок, убедившись, что можно использовать интерфейс FastEthernet0/0.700 Маршрутизатора 1.

Практикум

На компьютере Начальство-ПК1 пропингуйте адрес `172.31.70.254`.

Если все настроено правильно, вы должны получить следующий ответ:

```
PS C:\> ping 172.31.70.254

Pinging 172.31.70.254 with 32 bytes of data:
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
Reply from 172.31.70.254: bytes=32 time=1ms TTL=255
```

Это доказывает, что виртуальное соединение 700 между Коммутатором 1 и Маршрутизатором 1 работает. Но целью маршрутизатора является выполнение маршрутизации между виртуальными локальными сетями 600 и 700. Простой способ проверить, работает ли маршрутизация между виртуальными сетями, – пропинговать хост в одной виртуальной сети с хоста в другой.

Практикум

С компьютера Начальство-ПК1 пропингуйте компьютер Кадры-ПК1 (172.31.60.50).

Вы должны получить следующий ответ от хоста Кадры-ПК1:

```
PS C:\> ping 172.31.60.50

Pinging 172.31.60.50 with 32 bytes of data:
Reply from 172.31.60.50: bytes=32 time=1ms TTL=127
Reply from 172.31.60.50: bytes=32 time=1ms TTL=127
Reply from 172.31.60.50: bytes=32 time=2ms TTL=127
Reply from 172.31.60.50: bytes=32 time=1ms TTL=127
```

Пинг между хостами в разных виртуальных сетях не особенно увлекателен. Хуже того, огромное количество текстового вывода, который вы должны увидеть, может затмить глаза и заставит забыть обо всех деталях, которые происходят за кадром. Рисунок 14.2 иллюстрирует в деталях то, что происходит, когда вы пингуете компьютер Кадры-ПК1 с компьютера Начальство-ПК1.

На шаге 1 компьютер Начальство-ПК1 создает IP-пакет с запросом ping внутри него. Он инкапсулирует этот пакет внутрь Ethernet-кадра, адресованного MAC-адресу маршрутизатора – 0015.fa64.76d2, и отправляет его в виртуальную сеть 700. Поскольку субинтерфейс FastEthernet0/0.700 Маршрутизатора 1 находится в виртуальной сети 700, он получает Ethernet-кадр.

На шаге 2 Маршрутизатор 1 принимает IP-пакет и загружает его в новый Ethernet-кадр, адресованный MAC-адресу компьютера Кадры-ПК1 – 0800.27f8.0651. Он отправляет этот кадр из интерфейса FastEthernet0/0.600 в виртуальную сеть 600, где компьютер Кадры-ПК1 получает его.

На шаге 3 компьютер Кадры-ПК1 создает IP-пакет с ответом ping. Он инкапсулирует пакет внутрь Ethernet-кадра, адресованного MAC-адресу Маршрутизатора 1, и отправляет его в виртуальную сеть 600.

На шаге 4 Маршрутизатор 1 принимает IP-пакет, содержащий ответ ping, и инкапсулирует его в новый Ethernet-кадр, адресованный MAC-адресу компьютера Начальство-ПК1 – 0800.2759.d9fd.

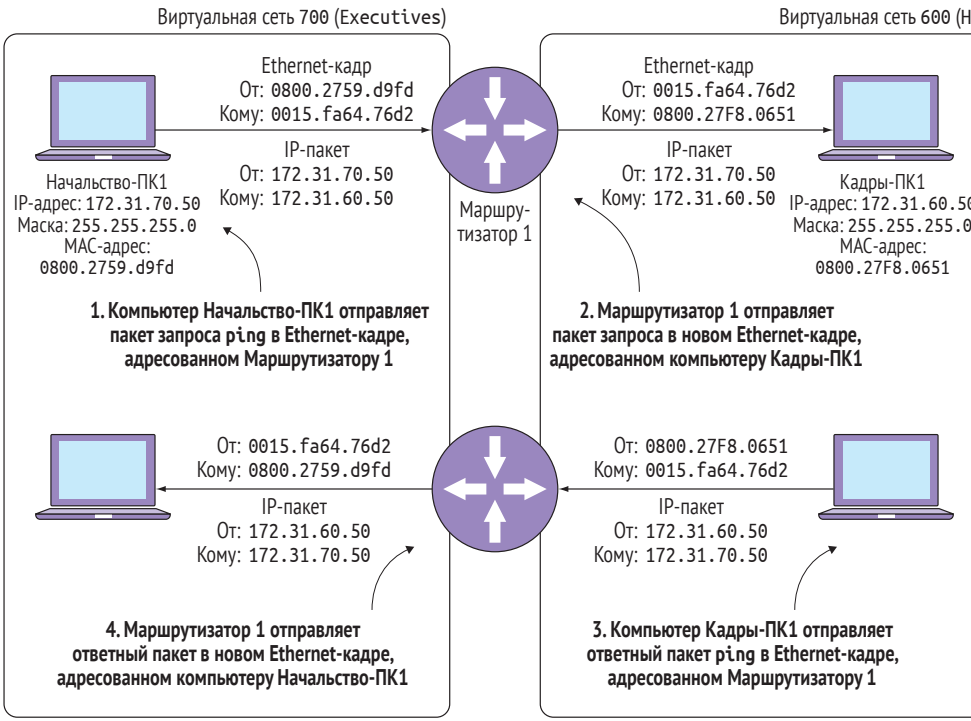

Рис. 14.2 ❖ Маршрутизатор 1 занимается маршрутизацией между виртуальными сетями 600 и 700

Как же много всего нужно для маршрутизации IP-пакета между двумя виртуальными сетями! Хотя рис. 14.2 поможет вам понять примеры в этой главе, в реальном мире вы не всегда будете иметь такую понятную диаграмму, чтобы визуально представлять, как IP-трафик проходит из одной виртуальной сети в другую. Вот почему важно, чтобы вы могли быстро определить на основе скучного вывода текста, как маршрутизатор будет маршрутизировать трафик.

14.4. Таблица IP-маршрутизации

В главе 7 вы вкратце рассмотрели таблицу IP-маршрутизации на Коммутаторе 1. Теперь пришло время узнать, как ее интерпретировать.

Практикум

На Маршрутизаторе 1 выполните команду `show ip route`.

Вы должны увидеть следующее:

```
Router1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
```

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    172.31.0.0/24 is subnetted, 2 subnets
C       172.31.60.0 is directly connected, FastEthernet0/0.600
C       172.31.70.0 is directly connected, FastEthernet0/0.700
C    192.168.1.0/24 is directly connected, FastEthernet0/0
```

Таблица IP-маршрутизации описывает каждую подсеть IP, о которой «знает» маршрутизатор, и какой интерфейс он должен использовать для достижения этой подсети. Информация в таблице маршрутизации Маршрутизатора 1 должна казаться очевидной. Например, Маршрутизатор 1 «знает», что он может достичь подсети `172.31.60.0/24` с субинтерфейса FastEthernet0/0.600, потому что IP-адрес, который вы настроили на этом субинтерфейсе, находится в подсети `172.31.60.0/24`. Это так называемый *подключенный маршрут*, поскольку Маршрутизатор 1 напрямую подключен к этой подсети.

Очевидно, но что делать, если Маршрутизатор 1 получает пакет для подсети, о котором он ничего не знает? Например, если он получает пакет для `1.2.3.4`, ему некуда отправить пакет, потому что для него нет соответствующего маршрута.

Практикум

На компьютере Начальство-ПК1 пропингуйте IP-адрес `1.2.3.4`.

Вы должны увидеть ряд неудачных пингов:

```
PS C:\> ping 1.2.3.4

Pinging 1.2.3.4 with 32 bytes of data:
Reply from 172.31.70.254: Destination host unreachable.
Reply from 172.31.70.254: Destination host unreachable.
Reply from 172.31.70.254: Destination host unreachable.
Reply from 172.31.70.254: Destination host unreachable.

Ping statistics for 1.2.3.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

Когда маршрутизатор получает пакет, предназначенный для подсети, для которой у него нет маршрута, он передаст отправителю сообщение `Destination host unreachable`.

До сих пор вы могли пинговать компьютер Кадры-ПК1 с компьютера Начальство-ПК1. Теперь пришло время попробовать пинг в противоположном направлении, от компьютера Кадры-ПК1 до компьютера Начальство-ПК1.

Практикум

С компьютера Кадры-ПК1 попробуйте пропинговать компьютер Начальство-ПК1.

Вы должны получить сообщения об успешных ответах:

```
PS C:\> ping 172.31.70.50

Pinging 172.31.70.50 with 32 bytes of data:
Reply from 172.31.70.50: bytes=32 time=1ms TTL=127
Reply from 172.31.70.50: bytes=32 time=1ms TTL=127
Reply from 172.31.70.50: bytes=32 time=2ms TTL=127
Reply from 172.31.70.50: bytes=32 time=1ms TTL=127

Ping statistics for 172.31.70.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

Напомню, что в главе 9 вы внедрили список ACL (Access Control List – список доступа) для блокировки всего IP-трафика из подсети `172.31.60.0/24` в подсеть `172.31.70.0/24`. Вы применили этот список ACL к SVI-интерфейсу виртуальной сети `600` на Коммутаторе 1, но в начале этой главы вы удалили этот SVI-интерфейс и назначили его IP-адрес (`172.31.60.254`) интерфейсу FastEthernet0/0.600 Маршрутизатора 1. Негативный эффект заключается в том, что компьютер Кадры-ПК1 теперь использует Маршрутизатор 1, у которого нет списка доступа в нужном расположении. Давайте это исправим!

14.5. Применение списка доступа на субинтерфейсе

Способ создания списка ACL и применения его к интерфейсу почти точно такой же на маршрутизаторе, как и на коммутаторе. Вы начинаете с создания списка ACL, чтобы запретить весь IP-трафик из подсети `172.31.60.0/24` в подсеть `172.31.70.0/24`.

Практикум

Используя следующую команду, создайте список доступа `100` на Маршрутизаторе 1:

```
access-list 100 deny ip 172.31.60.0 0.0.0.255 172.31.70.0 0.0.0.255
access-list 100 permit ip any any
```

Примените список доступа `100` к субинтерфейсу FastEthernet0/0.600:

```
int fa0/0.600
ip access-group 100 in
```

Проверьте результат, выполнив команду `show access-list`.

Вы должны увидеть новый список ACL:

```
Extended IP access list 100
    10 deny ip 172.31.60.0 0.0.0.255 172.31.70.0 0.0.0.255
    20 permit ip any any (23 matches)
```

Теперь, когда вы применили список ACL, вам больше не удастся пропинговать компьютер Начальство-ПК1 с компьютера Кадры-ПК1.

Практикум

С компьютера Кадры-ПК1 попробуйте снова пропинговать компьютер Начальство-ПК1.

Вы должны увидеть следующее:

```
PS C:\> ping 172.31.70.50

Pinging 172.31.70.50 with 32 bytes of data:
Reply from 172.31.60.254: Destination net unreachable.
Reply from 172.31.60.254: Destination net unreachable.
Reply from 172.31.60.254: Destination net unreachable.
Reply from 172.31.60.254: Destination net unreachable.
```

Обратите внимание, что сообщение `Destination net unreachable` похоже на то, что вы получали, пытаясь пропинговать адрес `1.2.3.4`. Эта ошибка – способ маршрутизатора сообщить, что он не может или, в этом случае, не отправит IP-пакет на указанный пункт назначения.

Вам не нужно быть экспертом расшифровки этих сообщений об ошибках. Просто помните, что когда они появляются, это может указывать на отсутствующий маршрут в таблице IP-маршрутизации или на блокировку трафика в списке доступа.

14.6. Команды в этой главе

Обратитесь к табл. 14.3, содержащей список всех команд, используемых в этой главе.

Таблица 14.3. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`interface fastethernet0/0.600`	Глобальный	Создает субинтерфейс FastEthernet0/0.600 в составе физического интерфейса FastEthernet0/0
`encapsulation dot1Q 600`	Интерфейс	Помещает выбранный субинтерфейс в виртуальную сеть `600`
`ip address 172.31.70.254 255.255.255.0`	Интерфейс	Назначает выбранному интерфейсу адрес `172.31.70.254/24`
`show vlans 600`	Глобальный	Показывает интерфейсы в виртуальной сети `600` и их соответствующие IP-адреса
`show ip route`	Глобальный	Показывает таблицу IP-маршрутизации

14.7. Практическое задание

Сейчас вы попрактикуетесь в создании и применении списков ACL на маршрутизаторе.

1. Создайте новый список ACL, чтобы запретить устройствам в подсети `Executives` доступ к подсети `HR`.
2. Примените список ACL к субинтерфейсу FastEthernet0/0.700.
3. Попробуйте пропинговать хост Кадры-ПК1 с компьютера Начальство-ПК1.
4. Удалите список ACL с интерфейса FastEthernet0/0.700.
5. Сохраните конфигурацию.

Глава 15

Направление трафика вручную с использованием таблицы IP-маршрутизации

Подключить два коммутатора, находящихся в одном офисе, очень просто. Все, что вам нужно сделать, – это проложить пару кабелей Ethernet, настроить интерфейсы, и дело сделано. Но теперь представьте, что ваши коммутаторы находятся в разных офисах на расстоянии в сотни километров друг от друга. Как вы их свяжете?

В главе 8 я говорил, что у вас есть несколько вариантов, когда дело доходит до соединения географически разделенных расположений. Два популярных метода – частные линии T1/E1 и виртуальные частные сети MPLS (VPN). Используя оба этих метода, оператор связи обеспечивает физическую связь между вашими расположениями. Но вам все равно придется настроить IP-маршрутизацию между этими расположениями.

В прошлой главе вы получили представление о том, как работает IP-маршрутизация, когда настраивали топологию «маршрутизатор-на-палочке». На рис. 15.1 показана топология, которую вы создали на нынешний момент.

Напомню, что вам не нужно было явно указывать Маршрутизатору 1, как маршрутизировать трафик между IP-подсетями. Все, что вам нужно было сделать, – это создать пару субинтерфейсов, настроить IP-адрес на каждом из них, и маршрутизатор позаботился обо всем остальном.

Но это только с одним маршрутизатором. Когда вы подключаете географически разделенные расположения, нужно настроить несколько маршрутизаторов. Взгляните на рис. 15.2; вы будете перенастраивать свою сеть так, как показано в этой главе.

Вы собираетесь разместить Маршрутизатор 1 между двумя коммутаторами и настроить его для маршрутизации IP-трафика между ними. Однако из диаграммы не очевидно, что Коммутаторы 1 и 2 также будут функционировать как маршрутизаторы. Вспомните из главы 7, что каждый коммутатор уровня 3 име-

ет в себе виртуальный маршрутизатор. Еще кое-что, что может быть неясным на данный момент, заключается в добавлении двух новых транзитных подсетей `10.0.12.0/30` между Коммутатором 1 и Маршрутизатором 1 и `10.0.21.0/30` между Маршрутизатором 1 и Коммутатором 2. В следующей главе я расскажу о транзитных подсетях.

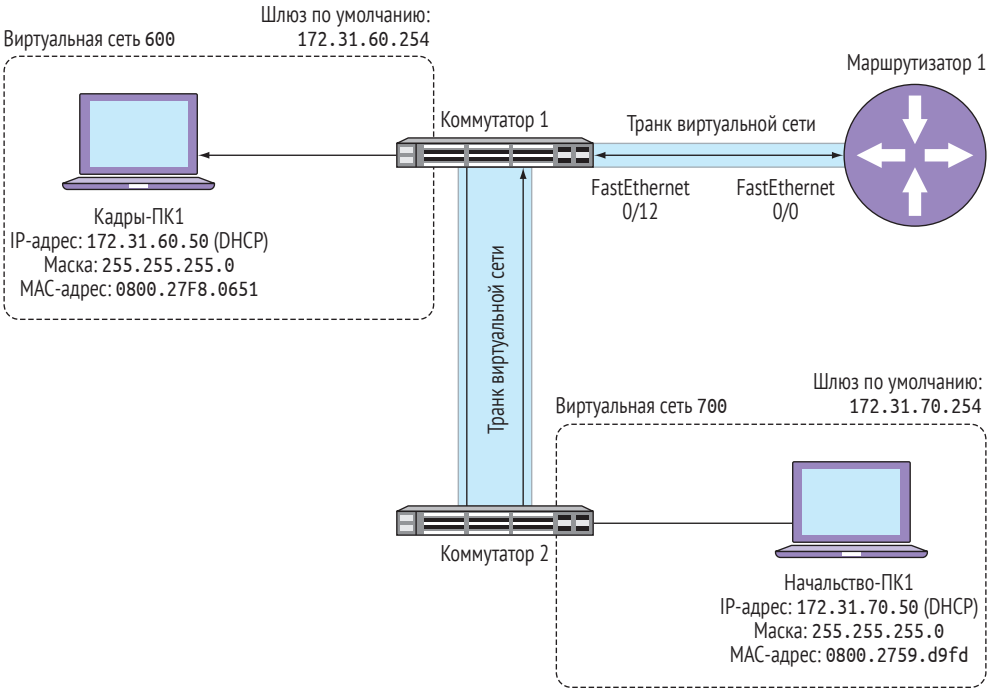

Рис. 15.1 ❖ Физическая топология «маршрутизатор-на-палочке»

Ниже представлены основные шаги, которым вы будете следовать в этой главе.

1. Подключите Маршрутизатор 1 к Коммутатору 2 и удалите настроенные субинтерфейсы на Маршрутизаторе 1.
2. Настройте интерфейсы между Маршрутизатором 1 и Коммутатором 2.
3. Создайте новый субинтерфейс между Маршрутизатором 1 и Коммутатором 1.
4. Настройте шлюзы по умолчанию для подсетей `Executives` и `HR`.
5. Создайте пул DHCP для подсети `Executives` на Коммутаторе 2.
6. Настройте статические IP-маршруты на Коммутаторах 1 и 2, а также на Маршрутизаторе 1.

Приступим!

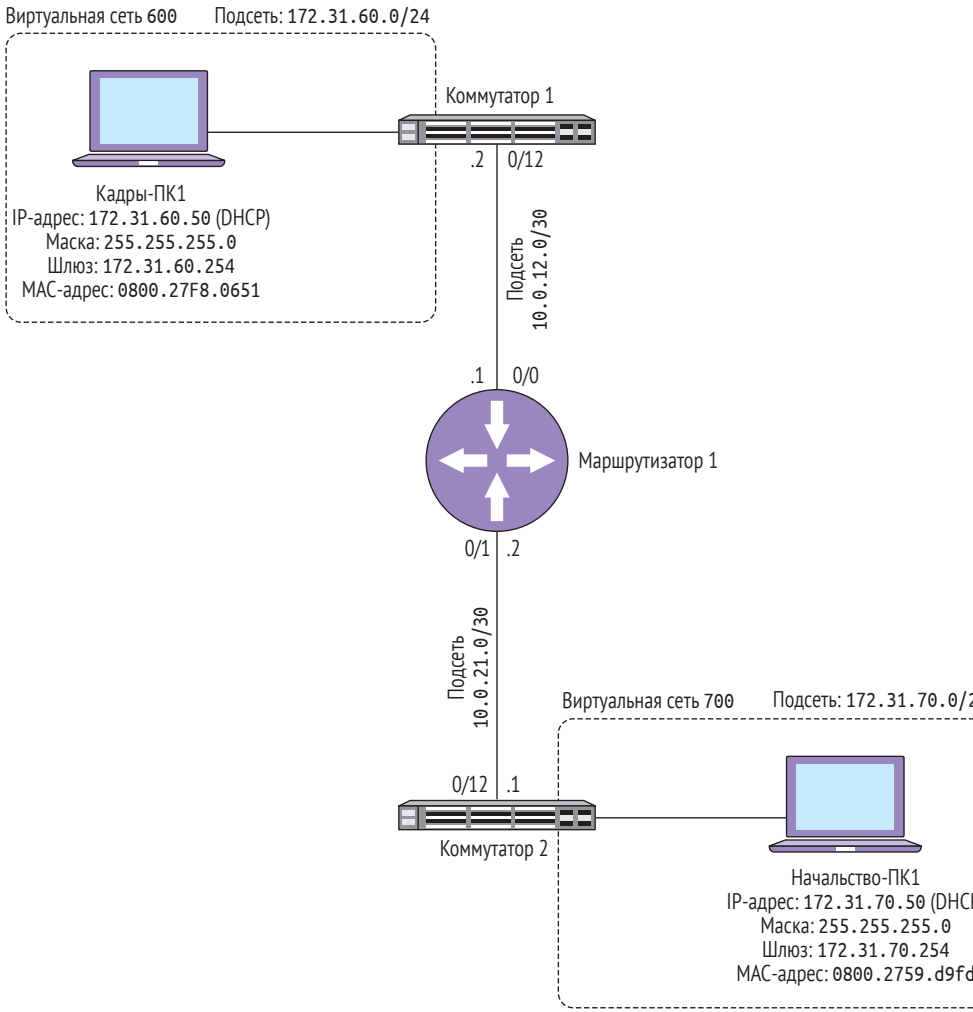

Рис. 15.2 ❖ Коммутаторы 1 и 2 подключены через Маршрутизатор 1

15.1. Подключение Маршрутизатора 1 к Коммутатору 2

Напомню, что в главе 14 вы настроили два субинтерфейса на Маршрутизаторе 1: FastEthernet0/0.600 и FastEthernet0/0.700. Поскольку вы больше не собираетесь использовать конфигурацию «маршрутизатор-на-палочке», вам нужно удалить их.

Практикум

Подключим интерфейс FastEthernet0/1 Маршрутизатора 1 к интерфейсу FastEthernet0/12 Коммутатора 2.
Выполните следующие команды, чтобы удалить оба субинтерфейса на Маршрутизаторе 1:

```
no interface fa0/0.600
no interface fa0/0.700
```

Как только вы выполните эти команды, появится сообщение: `Not all config may be removed and may reappear after reactivating the sub-interface`. Так и должно быть.
Проверьте, что оба субинтерфейса находятся в состоянии `deleted`, выполнив команду `show ip interface brief`.

Вы должны увидеть оба субинтерфейса в состоянии `deleted`:

```
Router1#show ip interface brief
Interface              IP-Address      OK?  Method  Status                Protocol
FastEthernet0/0        192.168.1.201   YES  NVRAM   up up
FastEthernet0/0.600    unassigned      YES  NVRAM   deleted               down
FastEthernet0/0.700    unassigned      YES  NVRAM   deleted               down
FastEthernet0/1        unassigned      YES  NVRAM   administratively down down
```

Обратите внимание, что интерфейс FastEthernet0/1, который теперь подключен к Коммутатору 2, не имеет IP-адреса и отключен администратором. Чтобы Маршрутизатор 1 мог управлять трафиком с Коммутатора 2 и обратно, вам нужно настроить транзитную подсеть между ними.

15.2. Настройка транзитных подсетей

Единственная цель транзитной подсети – передавать IP-трафик между двумя и только двумя устройствами в одном широковещательном домене. Отличие транзитной подсети – в относительно небольшом размере. Напомню из главы 7, что маска подсети определяет, насколько велика подсеть. Вам нужны две транзитные подсети: одна – между Коммутатором 1 и Маршрутизатором 1, а другая – между Маршрутизатором 1 и Коммутатором 2. Для каждой подсети требуется достаточно адресов только для двух устройств. Эти подсети перечислены в табл. 15.1.

Таблица 15.1. Транзитные подсети. Используя маску подсети 255.255.255.252, каждая транзитная подсеть достаточно велика, чтобы содержать только два устройства

Подсеть	Маска подсети	Количество доступных адресов
10.0.12.0	255.255.255.252	2
10.0.21.0	255.255.255.252	2

Вы можете использовать два разных способа настройки транзитной подсети между маршрутизатором и коммутатором. Например, назначить транзитные IP-адреса непосредственно физическим интерфейсам маршрутизатора и коммутатора. Или вы можете назначить транзитные IP-адреса субинтерфейсу маршрутизатора и транзитному SVI-интерфейсу виртуальной сети коммутатора. Я собираюсь продемонстрировать вам оба пути, начиная с первого.

15.2.1. Назначение транзитных IP-адресов непосредственно физическим интерфейсам

Процесс назначения транзитного IP-адреса физическому интерфейсу маршрутизатора несколько отличается от аналогичного процесса на коммутаторе. По мере выполнения упражнений я покажу отличия.

Практикум

Настройте в интерфейсе FastEthernet0/1 Маршрутизатора 1 IP-адрес `10.0.21.2` и маску подсети `255.255.255.252`:

```
int fa0/1
description To Switch1 Fa0/12
ip address 10.0.21.2 255.255.255.252
no shutdown
```

Вы должны увидеть, что интерфейс заработал:

```
*Jun 14 21:36:47.571: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Jun 14 21:36:48.571: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up
```

Затем вам нужно настроить транзитную подсеть на Коммутаторе 2. По умолчанию вы не можете назначить IP-адрес непосредственно порту коммутатора. Вам нужно выполнить команду `no switchport`, чтобы превратить порт в так называемый *маршрутизируемый интерфейс*. Как предполагает этот термин, маршрутизируемый интерфейс ведет себя точно так же, как интерфейс на маршрутизаторе.

Практикум

Включите IP-маршрутизацию на Коммутаторе 2 (вы уже включили ее на Коммутаторе 1 ранее):

```
ip routing
```

Настройте порт FastEthernet0/12 как маршрутизируемый интерфейс:

```
int fa0/12
no switchport
```

Присвойте ему IP-адрес 10.0.21.1 с маской подсети 255.255.255.252:

```
ip address 10.0.21.1 255.255.255.252
```

Проверьте результат, выполнив команду show ip int fa0/12.

Вы должны увидеть следующее:

```
Switch2#show ip int fa0/12
FastEthernet0/12 is up, line protocol is up
  Internet address is 10.0.21.1/30
  Broadcast address is 255.255.255.255
```

Обратите внимание на то, что система IOS показывает адрес в виде 10.0.21.1/30, вместо того чтобы предоставить вам маску подсети в виде 255.255.255.252. Объяснение взаимосвязи между маской подсети и обозначением косой черты связано с большим количеством двоичных вычислений и выходит за рамки этой главы. Просто знайте, что если вы видите адрес, указанный здесь, вы правильно настроили IP-адрес и маску подсети.

Следующее, что нужно сделать, – проверить, сможет ли Коммутатор 2 пропинговать новый IP-адрес Маршрутизатора 1.

Практикум

С Коммутатора 2 попробуйте пропинговать адрес 10.0.21.2:

```
Switch2#ping 10.0.21.2
```

Вы должны получить следующий ответ:

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.21.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1007 ms
```

Одно из преимуществ этого метода заключается в том, что он простой и не требует создания виртуальной сети для каждой подсети. Однако недостаток в том, что человек, не посвященный в технологии Cisco, может столкнуться с трудностями в понимании конфигурации.

15.2.2. Назначение транзитных IP-адресов субинтерфейсам и SVI-интерфейсам

Теперь вы создадите транзитную подсеть между Коммутатором 1 и Маршрутизатором 1, но на этот раз сделаете это немного по-другому. На Коммутаторе 1 вы создадите новую виртуальную сеть и SVI-интерфейс, а затем назначите ему один из транзитных IP-адресов.

Практикум

Настройте транзитную виртуальную сеть 999 и присвойте транзитный IP-адрес 10.0.12.1 его SVI-интерфейсу:

```
Switch1(config)#vlan 999
Switch1(config-vlan)#name Transit
Switch1(config-vlan)#exit
Switch1(config)#interface vlan999
Switch1(config-if)#ip address 10.0.12.1 255.255.255.252
```

Убедитесь, что интерфейс FastEthernet0/12 настроен как транковый порт:

```
Switch1(config)#interface fa0/12
Switch1(config-if)#description To Router1 Fa0/1
Switch1(config-if)#switchport trunk encapsulation dot1q
Switch1(config-if)#switchport mode trunk
```

Проверьте конфигурацию, выполнив команду show ip interface vlan999.

Вы должны увидеть следующее:

```
Show ip interface vlan999
Switch1#show ip interface vlan999
Vlan999 is up, line protocol is up
  Internet address is 10.0.12.1/30
  Broadcast address is 255.255.255.255
```

Следующим шагом будет настройка субинтерфейса в виртуальной сети 999 на Маршрутизаторе 1.

Практикум

На Маршрутизаторе 1 создайте субинтерфейс FastEthernet0/0.999 и поместите его в виртуальную сеть 999:

```
Router1(config)#int fa0/0.999
Router1(config-subif)#encapsulation dot1Q 999
```

Присвойте субинтерфейсу IP-адрес 10.0.12.2:

```
Router1(config-subif)#ip address 10.0.12.2 255.255.255.252
```

Проверьте результат, выполнив команду show ip int brief.

Вы должны увидеть созданный субинтерфейс и его новый IP-адрес:

```
Router1#sh ip int brief
Interface              IP-Address      OK?  Method  Status   Protocol
FastEthernet0/0        192.168.1.201   YES  manual  up       up
```

```
FastEthernet0/0.600    unassigned      YES  manual  deleted  down
FastEthernet0/0.700    unassigned      YES  manual  deleted  down
FastEthernet0/0.999    10.0.12.2       YES  manual  up       up
FastEthernet0/1        10.0.21.2       YES  manual  up       up
```

Повторюсь, рекомендуется пропинговать Коммутатор 1 для проверки IP-подключения:

```
Router1#ping 10.0.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.12.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/3/4 ms
```

Как я уже упоминал, это хороший способ, чтобы конфигурация стала более понятной тем, кто незнаком с системой IOS. Присвоение виртуальной сети имени `Transit` или в этом духе упрощает поиск.

15.3. Удаление транка между коммутаторами

Чтобы смоделировать дистанцию между коммутаторами, вам нужно удалить транк виртуальной сети между ними, который представляет собой агрегированный канал. Для этого вам не нужно физически отключать кабели Ethernet от портов FastEthernet0/22, 0/23 и 0/24, хотя вы можете, если хотите. Вместо этого вы можете закрыть интерфейс агрегированного канала.

Практикум

Выключите интерфейс Port-channel1 на Коммутаторе 1:

```
interface Port-channel1
shutdown
```

15.4. Настройка шлюзов по умолчанию

В предыдущей главе вы настроили два субинтерфейса на Маршрутизаторе 1 с IP-адресами шлюза по умолчанию для подсетей `HR` и `Executives`. В начале этой главы вы удалили оба этих субинтерфейса. В настоящее время устройства в подсетях `Executives` (`172.31.70.0/24`) и `HR` (`172.31.60.0/24`) не имеют шлюза по умолчанию.

Согласно рис. 15.3, Коммутатор 1 будет действовать как шлюз по умолчанию для подсети `HR`, тогда как Коммутатор 2 будет действовать как шлюз по умолчанию для подсети `Executives`.

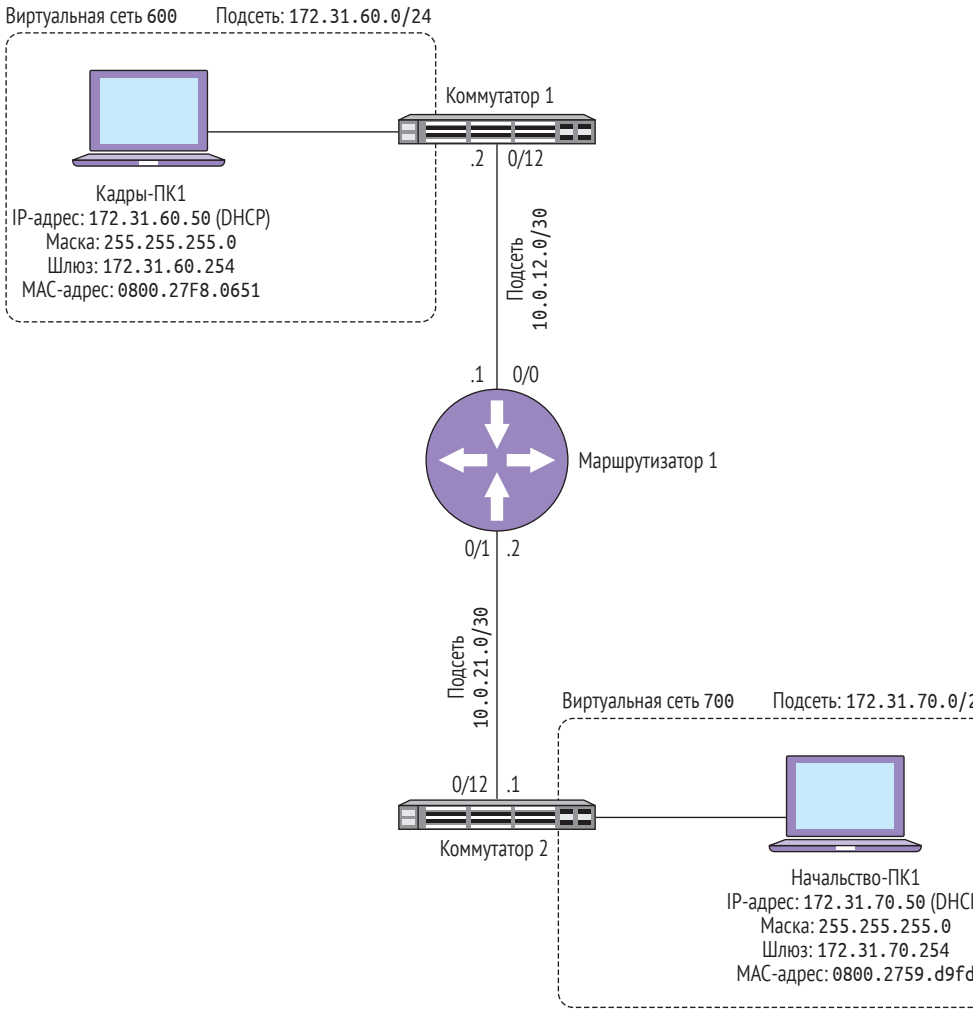

Рис. 15.3 ❖ Коммутаторы 1 и 2, соединенные через Маршрутизатор 1

Практикум

На Коммутаторе 2 создайте SVI-интерфейс виртуальной сети 700 и присвойте ему IP-адрес 172.31.70.254/24:

```
Switch2(config)#int vlan700
Switch2(config-if)#ip address 172.31.70.254 255.255.255.0
```

На Коммутаторе 1 пересоздайте SVI-интерфейс виртуальной сети 600 и присвойте ему IP-адрес 172.31.60.254/24:

```
Switch1(config)#int vlan600
Switch1(config-if)#ip address 172.31.60.254 255.255.255.0
```

15.5. Создание пула DHCP для подсети Executives

Ранее вы настроили две области адресов DHCP на Коммутаторе 1: одну – для подсети HR и одну для подсети Executives. Но теперь, когда виртуальная сеть Executives не имеет транка на Коммутаторе 1, устройства в этой виртуальной сети не смогут получить от нее DHCP-адреса. Вам нужно настроить DCHP-сервер на Коммутаторе 2.

Практикум

Создайте DHCP-пул для подсети Executives:

```
ip dhcp pool Executives
 network 172.31.70.0 255.255.255.0
 dns-server 192.168.100.10 192.168.100.11
 default-router 172.31.70.254
 domain-name benpiper.com
 lease 7
```

Исключите некоторые IP-адреса, чтобы сохранить их для статического присваивания:

```
ip dhcp excluded-address 172.31.70.251 172.31.70.254
ip dhcp excluded-address 172.31.70.1 172.31.70.49
```

Авторизуйтесь на компьютере Начальство-ПК1 и убедитесь, что он получил IP-адрес. Выполните команду ipconfig /renew (в оболочке командной строки Windows), если потребуется.

Вы должны увидеть обновленную информацию об IP-адресах:

```
PS C:\> ipconfig /renew

Windows IP Configuration

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : benpiper.com
   Link-local IPv6 Address . . . . . : fe80::d8ae:58d6:2dc0:9450%11
   IPv4 Address. . . . . . . . . . . : 172.31.70.50
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.31.70.254
```

Хотя компьютер Начальство-ПК1 имеет действительный IP-адрес, он все еще не может пропинговать компьютер Кадры-ПК1:

```
PS C:\> ping 172.31.60.51

Pinging 172.31.60.51 with 32 bytes of data:
Reply from 172.31.70.254: Destination host unreachable.
Reply from 172.31.70.254: Destination host unreachable.
Reply from 172.31.70.254: Destination host unreachable.
Reply from 172.31.70.254: Destination host unreachable.
```

Причина, по которой компьютер не пингуется, заключается в том, что шлюз для компьютера Начальство-ПК1 – Коммутатор 2 ничего не знает о подсети 172.31.60.0/24.

Практикум

Взгляните сами! На Коммутаторе 2 выполните команду `show ip route`.

Вы должны увидеть следующее:

```
Switch2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.21.0/30 is directly connected, FastEthernet0/12
L        10.0.21.1/32 is directly connected, FastEthernet0/12
      172.31.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.31.70.0/24 is directly connected, Vlan700
L        172.31.70.254/32 is directly connected, Vlan700
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Vlan1
L        192.168.1.102/32 is directly connected, Vlan1
```

Вы видите записи о транзитной подсети (`10.0.21.0/30`) и подсети `Executives` (`172.31.70.0/24`), но без подсети `HR` (`172.31.60.0/24`). Коммутатор 2 не знает, куда отправить пакет пинга с компьютера Начальство-ПК1! Чтобы решить проблему, вам необходимо создать статический IP-маршрут, состоящий из трех частей: целевой подсети, маски подсети и следующего перехода.

Следующий переход (next hop) – это IP-адрес устройства, на которое Коммутатор 2 должен отправить пакет. В данном случае это будет IP-адрес, назначенный ранее интерфейсу FastEthernet0/1 Маршрутизатора 1 (`10.0.21.2`).

Практикум

Добавьте статический маршрут подсети `172.31.60.0/24` с адресом `10.0.21.2` (Маршрутизатор 1) в качестве следующего перехода:

```
Switch2(config)#ip route 172.31.60.0 255.255.255.0 10.0.21.2
```

Проверьте результат, выполнив команду `show ip route`.

Вы должны увидеть новую запись для подсети HR:

```
Switch2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.21.0/30 is directly connected, FastEthernet0/12
L        10.0.21.1/32 is directly connected, FastEthernet0/12
      172.31.0.0/16 is variably subnetted, 3 subnets, 2 masks
S        172.31.60.0/24 [1/0] via 10.0.21.2
C        172.31.70.0/24 is directly connected, Vlan700
L        172.31.70.254/32 is directly connected, Vlan700
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Vlan1
L        192.168.1.102/32 is directly connected, Vlan1
```

Буква S указывает на то, что это статический маршрут. Попробуем пропинговать снова.

Практикум

С компьютера Начальство-ПК1 попробуйте снова пропинговать адрес 172.31.60.51.

```
PS C:\> ping 172.31.60.51

Pinging 172.31.60.51 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

Пинг все еще не проходит. Обратите внимание, что на Коммутаторе 1 команда show ip route не показывает путь к подсети Executives (172.31.70.0/24):

```
Switch2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set
```

```
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.12.0/30 is directly connected, Vlan999
L        10.0.12.1/32 is directly connected, Vlan999
      172.31.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.31.60.0/24 is directly connected, Vlan600
L        172.31.60.254/32 is directly connected, Vlan600
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Vlan1
L        192.168.1.101/32 is directly connected, Vlan1
```

Чтобы исправить это, вам нужно будет добавить статический маршрут к подсети Executives.

Практикум

Добавьте статический маршрут подсети Executives (172.31.70.0/24):

```
Switch1(config)#ip route 172.31.70.0 255.255.255.0 10.0.12.2
```

Проверьте результат, выполнив команду show ip route.

Теперь вы должны увидеть подсеть, закрепленную за статическим маршрутом:

```
Switch2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.12.0/30 is directly connected, Vlan999
L        10.0.12.1/32 is directly connected, Vlan999
      172.31.0.0/16 is variably subnetted, 3 subnets, 2 masks
C        172.31.60.0/24 is directly connected, Vlan600
L        172.31.60.254/32 is directly connected, Vlan600
S        172.31.70.0/24 [1/0] via 10.0.12.2
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Vlan1
L        192.168.1.101/32 is directly connected, Vlan1
```

Практикум

С компьютера Начальство-ПК1 попробуйте снова пропинговать компьютер Кадры-ПК1:

```
ping 172.31.60.51
```

Пинг все еще не работает:

```
PS C:\> ping 172.31.60.51

Pinging 172.31.60.51 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

Причина, по которой пропинговать компьютер все еще не удается, заключается в том, что Маршрутизатор 1 не знает о подсетях Executives и HR. Запомните, что весь трафик между подсетями HR и Executives должен проходить через Маршрутизатор 1. Чтобы исправить это, вам нужно сообщить Маршрутизатору 1 о наличии обеих подсетей.

Практикум

На Маршрутизаторе 1 выполните команду `show ip route`.

Вы должны увидеть, что Маршрутизатору 1 ничего не известно о каких-либо подсетях:

```
Switch2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/30 is subnetted, 2 subnets
C        10.0.12.0 is directly connected, FastEthernet0/0.999
C        10.0.21.0 is directly connected, FastEthernet0/1
C     192.168.1.0/24 is directly connected, FastEthernet0/0
```

Маршрутизатор знает о транзитных подсетях, это видно из его таблицы маршрутизации, потому что это *связанные с ним сети*, то есть он имеет IP-адреса в обеих подсетях. Но у него нет IP-адресов в подсетях Executives и HR, поэтому он не знает о них. Чтобы решить проблему, вам нужно добавить статические маршруты для обеих подсетей. Запомните, что Маршрутизатор 1 должен знать обо всех подсетях, поэтому он может не только передавать запросы пинга от компьютера Начальство-ПК1 на компьютер Кадры-ПК1, но также передавать ответ в противоположном направлении.

Практикум

На Маршрутизаторе 1 добавьте статический маршрут для сети `172.31.60.0/24`, используя транзитный адрес Коммутатора 1 в качестве следующего перехода:

```
Router1(config)#ip route 172.31.60.0 255.255.255.0 10.0.12.1
```

Добавьте другой статический маршрут для сети 172.31.70.0/24, используя транзитный IP-адрес Коммутатора 2 (10.0.21.1) в качестве следующего перехода:

```
Router1(config)#ip route 172.31.70.0 255.255.255.0 10.0.21.1
```

Проверьте результат, выполнив команду show ip route.

Теперь вы должны увидеть обе подсети, обозначенные в таблице маршрутизации как статические маршруты:

```
Switch2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      172.31.0.0/24 is subnetted, 2 subnets
S         172.31.60.0 [1/0] via 10.0.12.1
S         172.31.70.0 [1/0] via 10.0.21.1
      10.0.0.0/30 is subnetted, 2 subnets
C         10.0.12.0 is directly connected, FastEthernet0/0.999
C         10.0.21.0 is directly connected, FastEthernet0/1
C         192.168.1.0/24 is directly connected, FastEthernet0/0
```

Теперь вам должен быть доступен пинг обеих подсетей с Маршрутизатора 1.

Практикум

С Маршрутизатора 1 выполните пинг компьютеров Кадры-ПК1 (172.31.60.51) и Начальство-ПК1 (172.31.70.50):

```
Router1#ping 172.31.60.51
Router1#ping 172.31.70.50
```

Вы должны получить ответ от обоих компьютеров:

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.31.60.51, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Router1#ping 172.31.70.50

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.31.70.50, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Прекрасно! Теперь у вас маршрутизация между компьютерами Начальство-ПК1 и Кадры-ПК1 полностью настроена. Далее следует снова пропинговать компьютер Кадры-ПК1 с компьютера Начальство-ПК1.

Практикум

С компьютера Кадры-ПК1 попробуйте пропинговать компьютер Начальство-ПК1:

```
ping 172.31.60.51
```

```
PS C:\> ping 172.31.60.51
Pinging 172.31.60.51 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

Ох! Все равно не работает! Несмотря на то что IP-маршрутизация работает, пинг терпит неудачу из-за списка ACL, который вы настроили в главе 9. Пришло время избавиться от него.

Практикум

На Коммутаторе 1 удалите входящий список контроля доступа `101` с FastEthernet0/20, представляющий собой интерфейс, присоединенный к компьютеру Кадры-ПК1:

```
Switch1(config)#int fa0/20
Switch1(config-if)#no ip access-group 101 in
```

С компьютера Начальство-ПК1 попробуйте снова пропинговать адрес `172.31.60.51`.

Теперь все работает:

```
PS C:\> ping 172.31.60.51

Pinging 172.31.60.51 with 32 bytes of data:
Reply from 172.31.60.51: bytes=32 time=1ms TTL=125
Reply from 172.31.60.51: bytes=32 time=1ms TTL=125
Reply from 172.31.60.51: bytes=32 time=1ms TTL=125
Reply from 172.31.60.51: bytes=32 time=1ms TTL=125
```

Как же много работы, чтобы проложить маршрут между двумя подсетями! Хотя это и заняло много времени, статическая маршрутизация обычно настраивается однократно. Этот способ необязателен для организаций, в которых требуется подключение нескольких офисов в малом количестве расположений.

Но статическая маршрутизация не лишена недостатков. Компания, с которой я работал, использовала статическую маршрутизацию для подключения четырех офисов по всей стране. Но когда компания была приобретена более крупной корпорацией, имевшей сотни подсетей, статические маршруты боль-

ше не имели смысла. К счастью, у меня был другой вариант: *протоколы динамической маршрутизации*.

Протокол динамической маршрутизации позволяет каждому маршрутизатору в сети автоматически оповещать обо всех подсетях, о которых он знает. Каждый маршрутизатор распространяет информацию об этих подсетях по всей сети, пока каждый другой маршрутизатор не узнает о каждой подсети. Короче говоря, он автоматизирует процесс добавления маршрутов в таблицу IP-маршрутизации, так что вам этого не нужно.

Хотя концепция проста, настройка протоколов динамической маршрутизации смущает многих людей. В следующей главе я расскажу вам о самых популярных протоколах и покажу вам, как заменить статическую конфигурацию, выполненную в этой главе, на надежную динамическую – точно так же, как вы использовали бы ее в рабочей сети!

15.6. Команды, использованные в этой главе

Обратитесь к табл. 15.2 для просмотра списка всех команд, использованных в этой главе.

Таблица 15.2. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`ip routing`	Глобальный	Включает IP-маршрутизацию
`no switchport`	Интерфейс	Меняет режим работы интерфейса с switch port на routed interface
`show ip interface fa0/12`	–	Показывает IP-информацию по интерфейсу FastEthernet0/12

15.7. Практическое задание

В этом задании вам нужно удалить статические маршруты на Маршрутизаторе 1, а также Коммутаторах 1 и 2. Так вы подготовитесь к следующей главе, где будете настраивать динамическую маршрутизацию.

Выполните следующие команды в режиме глобальной конфигурации:

1. На Коммутаторе 1:

```
no ip route 172.31.70.0 255.255.255.0 10.0.12.2
```

2. На Коммутаторе 2:

```
no ip route 172.31.60.0 255.255.255.0 10.0.21.2
```

3. На Маршрутизаторе 1:

```
no ip route 172.31.60.0 255.255.255.0 10.0.12.1
```

и

```
no ip route 172.31.70.0 255.255.255.0 10.0.21.1
```

Глава 16

Интенсивный курс по протоколам динамической маршрутизации

В предыдущей главе вам пришлось вручную настроить статические маршруты для IP-соединений в вашей сети. В этой главе мы собираемся автоматизировать этот процесс, настроив два самых популярных протокола динамической маршрутизации: проприетарный протокол Cisco – Enhanced Interior Gateway Routing Protocol (EIGRP) и протокол Open Shortest Path First (OSPF). Но прежде чем вы узнаете, как работают эти два протокола и как их настроить, расскажу, зачем они нужны.

Сетевые администраторы иногда называют протоколы динамической маршрутизации *внутренними протоколами шлюза* (interior gateway protocols, IGP). Протоколы динамической маршрутизации автоматизируют процесс оповещения о подсетях других маршрутизаторов в сети. Хотя это может показаться не очень интересным, но поможет сэкономить вам много рабочего времени.

Протоколы динамической маршрутизации также отвечают за многое другое. Если в подсети есть несколько путей или маршрутов, протокол динамической маршрутизации будет выбирать лучший из них – как правило, самый короткий и самый быстрый путь. Протокол динамической маршрутизации также может автоматически сменить маршрутизацию при сбое каналов, если доступен другой путь. Недостаток протоколов динамической маршрутизации – в том, что их сложно настраивать, как вы узнаете в этой главе.

Все это будет проиллюстрировано, когда вы будете настраивать конфигурацию, показанную на рис. 16.1.

Обратите внимание, что в итоге у вас получатся избыточные соединения между Коммутаторами 1 и 2. У вас уже есть соединение через Маршрутизатор 1. Позже в этой главе вы добавите дополнительное соединение непосредственно между Коммутаторами 1 и 2. В отличие от предыдущих разделов, прямое соединение между коммутаторами не будет являться транком виртуальной сети.

Вместо этого вы настроите соответствующие интерфейсы на Коммутаторах 1 и 2 в качестве маршрутизируемых интерфейсов, чтобы эти коммутаторы действовали как напрямую подключенные маршрутизаторы. Оба протокола динамической маршрутизации, которые вы настроите, – EIGRP и OSPF – будут принимать решения, как маршрутизировать IP-трафик между виртуальными сетями 600 и 700.

Хотя протоколы EIGRP и OSPF могут работать одновременно, это сложная конфигурация, выходящая за рамки данной книги. Итак, сначала вы настроите протокол EIGRP, посмотрите, как он работает, а затем сбросите эту конфигурацию и настроите протокол OSPF. Давайте начнем!

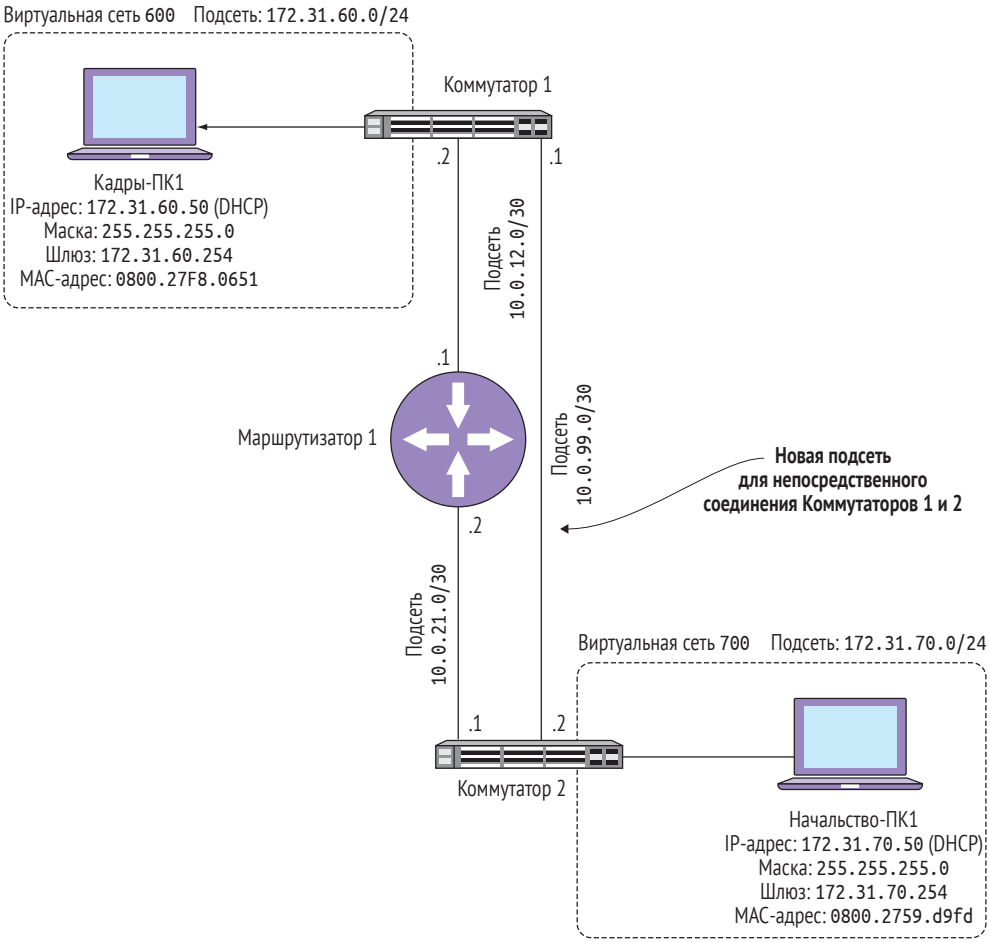

Рис. 16.1 ❖ Коммутаторы 1 и 2 подключены напрямую и через Маршрутизатор 1. Обратите внимание на добавленную подсеть 10.0.99.0/30 между Коммутаторами 1 и 2

16.1. Идентификаторы маршрутизаторов

Когда вы добавляете маршрутизаторы и подсети в сеть, становится все труднее обслуживать сетевое окружение. Вам нужен способ однозначно идентифицировать каждое устройство в сети. Один из способов сделать это – назначить соответствующие имена хостов, такие как `Router` и `Switch2`. Но протоколы динамической маршрутизации не идентифицируют маршрутизаторы на основе имен хостов. Вместо этого они используют *идентификаторы маршрутизатора* или *RID*.

Идентификатор RID выглядит так же, как IP-адрес, хотя технически это не так. Когда вы настраиваете протокол динамической маршрутизации на маршрутизаторе, он берет самый большой IP-адрес маршрутизатора (слева направо) и назначает его в качестве идентификатора RID. Взгляните на IP-адреса в Маршрутизаторе 1:

```
Router1#show ip interface brief
Interface              IP-Address      OK?   Method  Status     Protocol
FastEthernet0/0        192.168.1.201   YES   NVRAM   up         up
FastEthernet0/0.999    10.0.12.2       YES   NVRAM   up         up
FastEthernet0/1        10.0.21.2       YES   NVRAM   up         up
```

Идентификатор маршрутизатора будет `192.168.1.201`, потому что это самый большой идентификатор RID на любом из вышеперечисленных интерфейсов. Проблема заключается в том, что вы можете не помнить, что `192.168.1.201` – это Маршрутизатор 1. Это интуитивно не понятно. Но изменение IP-адреса на Маршрутизаторе 1, не говоря уже обо всех других устройствах в вашей сети, – то еще удовольствие. К счастью, вам это не нужно. Вместо этого вы можете назначить новый IP-адрес специальному типу интерфейса, называемому loopback-интерфейсом.

16.1.1. Настройка loopback-интерфейсов

До этого момента каждый настроенный вами IP-адрес относился к интерфейсу, будь то SVI- или физический интерфейс. Но есть еще один тип виртуального интерфейса, называемый *loopback-интерфейсом*. Вы можете назначить IP-адрес loopback-интерфейсу, но, в отличие от физического или SVI-интерфейса, loopback-интерфейс не привязан к определенному физическому интерфейсу или виртуальной сети, и он всегда работает (если вы не заблокируете его с административными правами). Если вы знакомы с loopback-интерфейсом на компьютере, это в целом одно и то же. Вы можете настроить столько loopback-интерфейсов, сколько вам нужно.

Самое значимое преимущество loopback-интерфейсов для протоколов маршрутизации заключается в том, что при включении протокола EIGRP или OSPF он будет использовать IP-адрес loopback-интерфейса в качестве идентификатора маршрутизатора, даже если есть больший IP-адрес на физическом или SVI-интерфейсе. Если у вас несколько loopback-интерфейсов, он будет использовать самый большой IP-адрес из всех loopback-интерфейсов. Таким

образом, вы можете неявно установить идентификатор маршрутизатора, настроив loopback-интерфейс.

Перед настройкой протокола динамической маршрутизации вы можете настроить все свои loopback-интерфейсы. После запуска протокола динамической маршрутизации он будет блокировать идентификатор маршрутизатора до тех пор, пока устройство не перезагрузится или не перезапустится протокол динамической маршрутизации, даже если вы измените или удалите IP-адреса своих loopback-интерфейсов.

Практикум

Поочередно авторизуйтесь на Коммутаторах 1 и 2 и Маршрутизаторе 1. Настройте интерфейс Loopback1 на каждом из них, как показано ниже.
Коммутатор 1:

```
interface loopback1
ip address 1.1.1.1 255.255.255.255
```

Коммутатор 2:

```
interface loopback1
ip address 2.2.2.2 255.255.255.255
```

Маршрутизатор 1:

```
interface loopback1
ip address 12.12.12.12 255.255.255.255
```

На всех устройствах проверьте IP-адреса loopback-интерфейсов, выполнив команду `show ip interface brief`. Запомните, что имена таких интерфейсов всегда начинаются со слова `loopback`.

Маска подсети `255.255.255.255` может несколько запутать. Напомню, что маска подсети определяет границы каждой подсети. В этом случае маска подсети указывает, что каждый IP-адрес loopback-интерфейса является единственным IP-адресом в подсети. Не волнуйтесь, если это звучит странно. Дело в том, что, используя столь странную маску, вы защищаете IP-адреса.

И последнее замечание о loopback-интерфейсах: они необязательны, но я рекомендую их настроить в любом случае, особенно если в вашей организации принято назначать неоднозначные имена устройствам Cisco. Loopback-интерфейсы могут служить хорошим уникальным идентификатором, даже если это нужно в целях персонального использования. Как только вы сконфигурируете свои loopback-интерфейсы, вы готовы начать настройку протокола EIGRP.

16.2. Настройка протокола EIGRP

Настройка протокола EIGRP проста, но команды, которые вы будете выполнять, не так интуитивно понятны, как хотелось бы. Многие люди находят их совершенно запутанными, поэтому я потрачу некоторое время на объяснения.

Есть две важные вещи, которые необходимо выполнить для настройки протокола EIGRP на каждом маршрутизаторе в вашей сети. Во-первых, вам нужно сообщить протоколу EIGRP, какие подсети следует использовать для связи с другими маршрутизаторами (с *соседями*), на которых запущен протокол EIGRP. Во-вторых, вы также должны сообщить протоколу EIGRP, о каких подсетях вы хотите извещать этих соседей.

Взгляните на рис. 16.2. Коммутаторы 1 и 2 и Маршрутизатор 1 будут работать по протоколу EIGRP друг с другом. Для начала вы включите протокол EIGRP только в двух подсетях: `10.0.12.0/30` между Коммутатором 1 и Маршрутизатором 1 и `10.0.21.0/30` между Маршрутизатором 1 и Коммутатором 2.

Практикум

На Коммутаторе 1 выполните следующие команды, чтобы включить протокол EIGRP:

```
Switch1(config)# router eigrp 7
Switch1(config-router)# network 10.0.12.1 0.0.0.0
```

Команда `router eigrp 7` переводит маршрутизатор в режим настройки протокола EIGRP. Число `7` указывает на номер *автономной системы* протокола EIGRP. Номер автономной системы произволен и не имеет значения сам по себе, но все маршрутизаторы с протоколом EIGRP, которые вы хотите включить в общую маршрутизацию, должны иметь один и тот же номер автономной системы. Так же, как идентификатор маршрутизатора однозначно идентифицирует маршрутизатор, номер автономной системы однозначно идентифицирует группу маршрутизаторов с протоколом EIGRP, которые используют одни и те же маршруты.

Последняя часть команды `network 10.0.12.1 0.0.0.0` – это подстановочная маска. Команда `network` смущает многих людей, потому что она выполняет два разных действия. Во-первых, она включает протокол EIGRP на интерфейсе коммутатора `10.0.12.1`, заставляя его взаимодействовать по протоколу EIGRP с Маршрутизатором 1. Во-вторых, это приводит к тому, что протокол EIGRP извещает о подсети, частью которой является его IP-адрес. Поскольку `10.0.12.1` – часть подсети `10.0.12.0/30`, протокол EIGRP оповещает об этой подсети. Дальше вы настроите протокол EIGRP оповещения о сети `172.31.60.0/24`.

Практикум

Не выходя из режима конфигурации маршрутизации протокола EIGRP на Коммутаторе 1, выполните следующую команду:

```
Switch1(config-router)# network 172.31.60.0 0.0.0.255
```

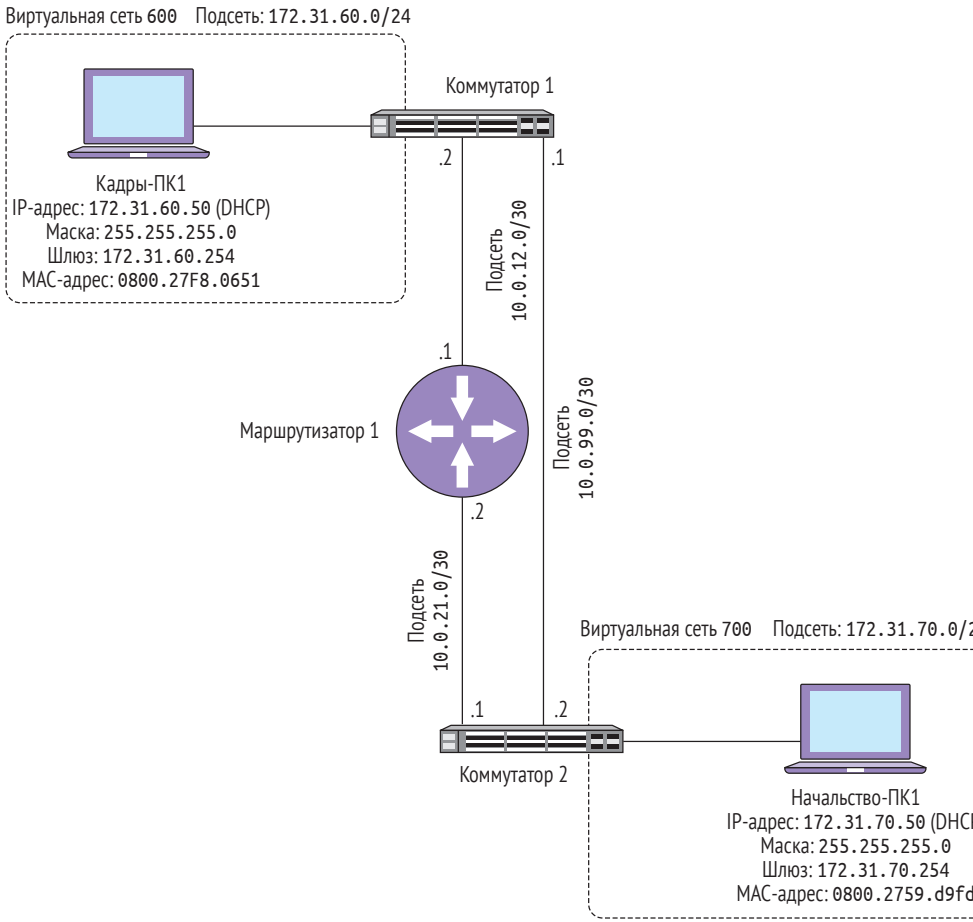

Рис. 16.2 ❖ Включение протокола EIGRP для подсетей `10.0.12.0/30`, `10.0.99.0/30`, `10.0.21.0/30`, `172.31.60.0/24` и `172.31.70.0/24`

Эта команда включает протокол EIGRP на всех интерфейсах, имеющих IP-адрес в диапазоне от `172.31.60.0` до `172.31.60.255`. Вспомним из главы 9, что подстановочная маска представляет собой обратную маску подсети. Подстановочная маска `0.0.0.255` эквивалентна маске подсети `255.255.255.0`. Следовательно, первые три октета в инструкции должны соответствовать IP-адресу интерфейса, но последним октетом может быть любое число от `0` до `255`. SVI-интерфейс виртуальной сети `600` на Коммутаторе 1 имеет IP-адрес `172.31.60.254`, поэтому IOS разрешает протокол EIGRP на SVI-интерфейсе виртуальной сети `600`. В рамках соглашения система IOS также извещает о подсети `172.31.60.0/24` другие маршрутизаторы с протоколом EIGRP. На данный момент вы прочитали намного больше, чем сделали, поэтому пришло время увидеть протокол EIGRP в действии!

Практикум

Проверьте, что на Коммутаторе 1 включен протокол EIGRP, выполнив указанную ниже команду:

```
show ip protocols
```

Ваш экран должен заполниться довольно длинным выводом:

```
Switch1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 7"
  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-IPv4 Protocol for AS(7)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 1.1.1.1
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    10.0.12.1/32
    172.31.60.0/24
  Routing Information Sources:
    Gateway Distance Last Update
  Distance: internal 90 external 170
```

Обратите внимание, что идентификатор маршрутизатора – 1.1.1.1. Вам нужно понять, что это IP-адрес интерфейса Loopback0, который вы настроили пару минут назад. Также взгляните на две подсети в разделе Routing for Networks. Их имена должны совпадать с теми, что вы указали с помощью команды network. Может показаться странным, что вывод не содержит информацию о том, какие интерфейсы EIGRP включены. Чтобы получить эту информацию, вам понадобится другая команда.

Практикум

На Коммутаторе 1 проверьте, на каких интерфейсах включен протокол EIGRP:

```
show ip eigrp interfaces
```

Затем:

```
show ip interface brief | i up
```

Вы должны увидеть следующее:

```
Switch1#show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(7)
        Xmit Queue PeerQ Mean Pacing Time Multicast Pending
Interface  Peers  Un/Reliable  Un/Reliable  SRTT  Un/Reliable  Flow Timer  Routes
Vl600      0      0/0          0/0          0     0/0          0           0
Vl999      0      0/0          0/0          0     0/0          0           0

Switch1#show ip int br | i up
Vlan1                  192.168.1.101   YES NVRAM  up          up
Vlan600                172.31.60.254   YES NVRAM  up          up
Vlan999                10.0.12.1       YES NVRAM  up          up
FastEthernet0/12       unassigned      YES unset  up          up
Loopback1              1.1.1.1         YES NVRAM  up          up
```

Надеюсь, все, что вы узнали в предыдущих главах, начинает проясняться. Протокол EIGRP включен на SVI-интерфейсах виртуальных сетей `600` и `999`, которые имеют IP-адреса в подсетях `172.31.60.0/24` и `10.0.12.0/30` соответственно. Это не так очевидно из выводов команд. Информация разрознена, и вам нужно собрать ее, поэтому я выполнил обе команды последовательно. Теперь, когда вы настроили протокол EIGRP на Коммутаторе 1, пришло время настроить его соседа, Маршрутизатор 1.

Практикум

На Маршрутизаторе 1 выполните следующие команды:

```
router eigrp 7
network 10.0.12.2 0.0.0.0
network 10.0.21.2 0.0.0.0
```

Вы должны тотчас же увидеть следующее сообщение на консоли:

```
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 7: Neighbor 10.0.12.1 (FastEthernet0/0.999) is up: new adjacency
```

Кстати, `DUAL` – это аббревиатура от словосочетания *Diffusing Update Algorithm* – алгоритма вычисления метрики, который протокол EIGRP использует для расчета наилучшего пути к пункту назначения. Вам не нужно понимать, как работает этот алгоритм, но знание о DUAL и его использовании в протоколе EIGRP может быть полезно, как вы увидите через секунду.

Когда два маршрутизатора с протоколом EIGRP начинают взаимодействие друг с другом, они образуют *смежность* и начинают обмениваться маршрутами. На этом этапе Маршрутизатор 1 должен иметь маршрут к подсети 172.31.60.0/24.

Практикум

Выполните команду `show ip route eigrp`, чтобы увидеть маршруты, полученные по протоколу EIGRP.

Вы должны увидеть только один маршрут по протоколу EIGRP:

```
Router1#show ip route eigrp
     172.31.0.0/24 is subnetted, 1 subnets
D          172.31.60.0 [90/28416] via 10.0.12.1, 01:52:49, FastEthernet0/0.999
```

Обратите внимание на новый маршрут, обозначенный буквой `D` (т. е. DUAL). Он относится к сети `172.31.60.0/24`. Следующий переход – `10.0.12.1`. Это транзитный IP-адрес интерфейса Коммутатора 1, подключенного к Маршрутизатору 1. Если вы видите этот маршрут, вы готовы завершить настройку на Коммутаторе 2.

Практикум

Выполните следующие команды на Коммутаторе 2:

```
router eigrp 7
network 10.0.21.1 0.0.0.0
network 172.31.70.0 0.0.0.255
```

На этот раз проверьте конфигурацию, выполнив команду `show ip eigrp neighbors`.

Вы должны увидеть смежность с транзитным IP-адресом Маршрутизатора 1 – `10.0.21.2`:

```
Switch2#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(7)
H   Address       Interface        Hold Uptime   SRTT   RTO   Q   Seq
                                   (sec)         (ms)         Cnt Num

0   10.0.21.2     Fa0/12           15 00:01:22   1      100   0   9
```

Теперь пришло время для настоящего теста. Если все работает правильно, вы должны увидеть два маршрута по протоколу EIGRP для сетей `172.31.60.0/24` и `10.0.12.0/30`.

Практикум

Выполните команду `show ip route eigrp` на Коммутаторе 2.

Вывод на Коммутаторе 2 будет немного более подробным, чем на Маршрутизаторе 1:

```
Switch2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override
```

```
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
D        10.0.12.0/30 [90/30720] via 10.0.21.2, 00:03:37, FastEthernet0/12
      172.31.0.0/16 is variably subnetted, 3 subnets, 2 masks
D        172.31.60.0/24 [90/30976] via 10.0.21.2, 00:03:37, FastEthernet0/12
```

Основываясь на этом выводе, Коммутатор 2 и устройства в подсети Executives (172.31.70.0/24) должны иметь доступ к устройствам в подсети HR (172.31.60.0/24). В предыдущих главах вы проверили возможность подключения, пропинговав с одного компьютера другой. Но теперь я покажу вам более удобный способ.

Практикум

На Коммутаторе 2 выполните следующую команду:

```
ping 172.31.60.254 source 172.31.70.254
```

Если все правильно, вы должны получить успешный пинг:

```
Switch2#ping 172.31.60.254 source 172.31.70.254
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.31.60.254, timeout is 2 seconds:
Packet sent with a source address of 172.31.70.254
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/5/9 ms
```

Очевидно, что команда выполняет пинг с адреса 172.31.70.254 Коммутатора 2 и отправляет его на адрес 172.31.60.254. Это эффективный способ обеспечения полной IP-связи между двумя подсетями.

16.2.1. Выбор наилучшего маршрута

Одним из важных преимуществ использования протоколов динамической маршрутизации является то, что они могут автоматически настраивать передачу трафика по лучшему пути, если доступно несколько. Для практики подключите Ethernet-кабель непосредственно к Коммутаторам 1 и 2 и назначьте новые IP-адреса каждому интерфейсу.

Практикум

Подключите Ethernet-кабель напрямую между интерфейсами FastEthernet0/24 на Коммутаторах 1 и 2.

Переведите интерфейсы в режим маршрутизации и присвойте им IP-адреса, как показано ниже.

На Коммутаторе 1:

```
interface fa0/24
No switchport
Ip address 10.0.99.1 255.255.255.252
No shutdown
```

На Коммутаторе 2:

```
Interface fa0/24
No switchport
Ip address 10.0.99.2 255.255.255.252
No shutdown
```

Имейте в виду, что вы еще не включили протокол EIGRP на этих новых интерфейсах, поскольку в отношении протокола EIGRP существует только один путь между Коммутаторами 1 и 2. Чтобы убедиться в этом, выполните команду `traceroute`.

Практикум

На Коммутаторе 2 выполните следующую команду:

```
traceroute 172.31.60.254
```

В выводе должно быть указание на то, что трафик идет через Маршрутизатор 1 вместо перехода непосредственно на Коммутатор 1:

```
Switch2#traceroute 172.31.60.254
Type escape sequence to abort.
Tracing the route to 172.31.60.254
VRF info: (vrf in name/id, vrf out name/id)
 1 10.0.21.2 9 msec 0 msec 0 msec
 2 10.0.12.1 0 msec *  0 msec
```

Данные поступают на адрес `10.0.21.2` (Маршрутизатор 1), а затем на адрес `10.0.12.1` (Коммутатор 2) – это два перехода. По умолчанию протокол EIGRP выбирает наилучший путь, основанный частично на количестве переходов. Чем меньше переходов, чтобы добраться до пункта назначения, тем короче путь. Очевидно, что прямое соединение между Коммутаторами 1 и 2 короче, чем переход от Коммутатора 2 к Маршрутизатору 1, а затем к Коммутатору 1. Но чтобы протокол EIGRP рассмотрел новый путь, вам необходимо явно перенастроить протокол EIGRP на Коммутаторах 1 и 2.

Практикум

На Коммутаторе 1 выполните следующие команды, чтобы включить протокол EIGRP для новой транзитной сети:

```
Switch1(config-if)#router eigrp 7
Switch1(config-router)#network 10.0.99.1 0.0.0.0
```

На Коммутаторе 2 выполните следующие команды:

```
Switch2(config-if)#router eigrp 7
Switch2(config-router)#network 10.0.99.2 0.0.0.0
```

Напомню, что число 7 указывает на номер автономной системы, который должен быть одинаковым на всех маршрутизаторах с протоколом EIGRP в топологии. На Коммутаторе 2 выполните команду `show ip eigrp neighbors`.

Вы должны увидеть нечто подобное:

```
Switch2#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(7)
H   Address       Interface        Hold Uptime   SRTT   RTO   Q   Seq
                                   (sec)         (ms)        Cnt Num

1   10.0.99.1     Fa0/24           11 00:00:33   1598   5000   0   9
0   10.0.21.2     Fa0/12           13 00:37:55      1    100   0   13
```

Теперь Коммутатор 2 имеет две смежности: одну с Коммутатором 1 и другую с Маршрутизатором 1. Поскольку лучший путь к сети `172.31.60.0/24` лежит напрямую через Коммутатор 1, а не Маршрутизатор 1, протокол EIGRP должен был изменить маршрут, чтобы отразить новое соединение.

Практикум

На Коммутаторе 2 снова выполните команду `show ip route eigrp`.

Вы должны увидеть, что протокол EIGRP обновил маршруты следующим образом:

```
Switch2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D        10.0.12.0/30 [90/28416] via 10.0.99.1, 00:02:11, FastEthernet0/24
      172.31.0.0/16 is variably subnetted, 3 subnets, 2 masks
D        172.31.60.0/24 [90/28416] via 10.0.99.1, 00:02:11, FastEthernet0/24
```

Наилучший маршрут к адресу `172.31.60.0/24` теперь лежит через адрес `10.0.99.1` (Коммутатор 1). Чтобы по-настоящему оценить, что произошло, выполните еще раз команду `traceroute`.

Практикум

На Коммутаторе 2 снова выполните команду `traceroute`:

```
traceroute 172.31.60.254
```

В данном случае вы должны увидеть только один переход:

```
Switch2#traceroute 172.31.60.254
Type escape sequence to abort.
Tracing the route to 172.31.60.254
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.99.1 0 msec *  0 msec
```

Бинго! Путь пролегает прямо с Коммутатора 2 на Коммутатор 1.

16.2.2. Маршрутизация при сбоях

Одним из основных преимуществ протоколов динамической маршрутизации является их способность автоматически выбирать новые маршруты при сбоях. Протокол EIGRP может быстро определять и реагировать на отказ, иногда даже за доли секунды. Чтобы продемонстрировать это, вы сымитируете отказ прямого соединения между Коммутаторами 1 и 2.

Практикум

Физически отсоедините один из штекеров кабеля между портами FastEthernet0/24 Коммутаторов 1 и 2. Либо же выключите интерфейс одного из портов. Выполните команду `show ip route eigrp`.

Волшебным образом протокол EIGRP изменит маршрут так, чтобы он пролегал через Маршрутизатор 1:

```
Switch2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
D        10.0.12.0/30 [90/30720] via 10.0.21.2, 00:00:05, FastEthernet0/12
      172.31.0.0/16 is variably subnetted, 3 subnets, 2 masks
D        172.31.60.0/24 [90/30976] via 10.0.21.2, 00:00:05, FastEthernet0/12
```

16.2.3. Выводы по протоколу EIGRP

Можно долго рассказывать о протоколе EIGRP, достаточно долго, чтобы написать сотни страниц и занять часы вашего времени, так как рассмотренное мной – всего лишь основы. Чтобы реализовать базовую, минимальную конфигурацию протокола EIGRP, вам необходимо сделать следующее:

1) определить, какие устройства будут участвовать в автономной системе протокола EIGRP;

2) определить номер автономной системы, который будут использовать все устройства;
3) определить, о каких подсетях с протоколом EIGRP будут извещаться устройства;
4) выполнить соответствующие команды протокола EIGRP, описанные в этом разделе.

Прежде чем перейти к протоколу OSPF, я хочу дать вам несколько указаний, когда следует сделать выбор в пользу протокола EIGRP. Как я упоминал в начале главы, EIGRP разрабатывался как проприетарный протокол компании Cisco, что означает, что он изначально не поддерживался устройствами, разработанными не в компании Cisco. В последние годы компания Cisco открыла исходные коды протокола EIGRP, чтобы другие производители могли разрабатывать собственные реализации протокола. Тем не менее, если ваша сеть содержит маршрутизаторы или коммутаторы сторонних производителей, мой совет – избегать протокола EIGRP.

Другое дело – это размер вашей сети. Компания Cisco разработала протокол EIGRP для использования в сетях масштабом в пределах 500 маршрутизаторов. Если ваша организация имеет или в конечном итоге будет иметь большее количество устройств, протокол EIGRP не станет лучшим выбором. Вместо этого вам нужен другой протокол динамической маршрутизации, называемый Open Shortest Path First (OSPF). Как он настраивается, вы узнаете в следующем разделе. Но сначала вам нужно удалить конфигурацию протокола EIGRP.

Практикум

Вновь подключите Коммутаторы 1 и 2 напрямую (либо откройте порт, который вы ранее закрыли).
Выполните следующую команду конфигурации, чтобы отменить использование протокола EIGRP, на Коммутаторах 1 и 2 и Маршрутизаторе 1:

```
no router eigrp 7
```

16.3. Протокол OSPF

Протокол OSPF предназначен для тех же целей, что и EIGRP, но разработан как более масштабируемый для очень крупных сетей. Настройка протокола OSPF во многом аналогична конфигурации EIGRP, с небольшой лишь разницей.

Практикум

На Коммутаторе 2 выполните следующие команды, чтобы включить протокол OSPF:

```
router ospf 1
network 10.0.21.1 0.0.0.0 area 0
network 172.31.70.0 0.0.0.255 area 0
network 10.0.99.2 0.0.0.0 area 1
```

Команда network имеет дополнительный параметр – area, за которым следует номер. Я вскоре объясню значение этого параметра протокола OSPF. Но вначале проверьте вашу конфигурацию, выполнив команду show ip protocols.

Вы должны увидеть следующее:

```
Switch2#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 2.2.2.2
  It is an area border router
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.0.21.1 0.0.0.0 area 0
    10.0.99.2 0.0.0.0 area 1
    172.31.70.0 0.0.0.255 area 0
  Routing Information Sources:
  Gateway           Distance Last Update
    1.1.1.1 110 00:00:08
    12.12.12.12 110 00:00:08
  Distance: (default is 110)
```

Один из способов достижения масштабируемости протокола OSPF заключается в использовании *зон* (area). Я не буду вдаваться в технические подробности о том, что такое зоны и как они работают, но чтобы запустить протокол OSPF, вам нужно настроить, по крайней мере, одну сеть в *зоне 0* (area 0), также известную как *магистральная зона*.

Вот что вам нужно помнить о зонах: протокол OSPF предназначен для поддержания трафика в пределах одной зоны, даже если есть лучший путь через другую зону. В предыдущей конфигурации вы установили прямое соединение между Коммутаторами 1 и 2 в зоне 1, оставив другие подсети в зоне 0. Смотрите рис. 16.3, чтобы получить полное представление о том, как будет выглядеть ваша конфигурация сетевого окружения.

Поскольку путь через Маршрутизатор 1 будет полностью лежать в зоне 0, протокол OSPF предпочтет этот путь, даже если доступен лучший путь. Давайте продолжим настройку.

Практикум

Настройте протокол OSPF на Маршрутизаторе 1 путем выполнения следующих команд:

```
router ospf 1
network 10.0.12.2 0.0.0.0 area 0
network 10.0.21.2 0.0.0.0 area 0
```

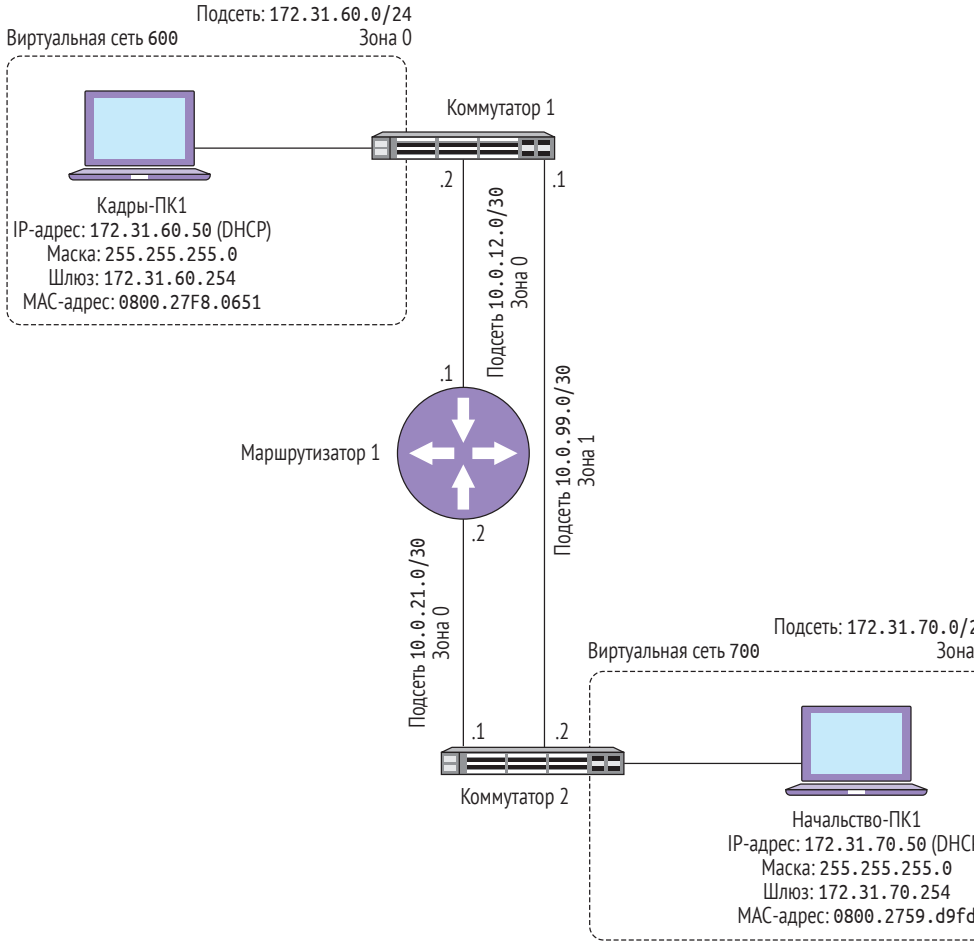

Рис. 16.3 ❖ Все подсети находятся в зоне 0 протокола OSPF, за исключением подсети `10.0.99.0/30` между Коммутаторами 1 и 2

Кстати, число `1` в строке `router ospf 1` указывает номер процесса протокола OSPF, который представляет собой способ однозначной идентификации экземпляра протокола OSPF на маршрутизаторе. Вам не нужно задумываться об этом. Вы просто должны помнить, что нужно его задать при настройке протокола OSPF. В отличие от номера автономной системы EIGRP, номер процесса протокола OSPF не должен совпадать на всех ваших маршрутизаторах. Но рекомендуется придерживаться того же номера для удобства.

Вы должны увидеть следующий вывод:

```
%OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/1 from LOADING to FULL, Loading Done
```

Сообщение в оболочке командной строки указывает, что Маршрутизатор 1 сформировал смежность с адресом 2.2.2.2, что означает, что они обменялись маршрутами и полностью синхронизированы. Запомните, что 2.2.2.2 – это IP-адрес loopback-интерфейса Коммутатора 2. Как и EIGRP, протокол OSPF берет наивысший IP-адрес loopback-интерфейса и использует его в качестве идентификатора маршрутизатора. В отличие от EIGRP, протокол OSPF активно использует идентификатор маршрутизатора в своих выводах.

Практикум

На Маршрутизаторе 1 посмотрите маршруты по протоколу OSPF в таблице маршрутизации:

```
Show ip route ospf
```

Затем посмотрите подробно детали маршрута к сети 172.31.70.0:

```
Show ip route 172.31.70.0
```

Вот что вы должны увидеть:

```
Router1#show ip route ospf
     172.31.0.0/24 is subnetted, 1 subnets
O       172.31.70.0 [110/2] via 10.0.21.1, 00:18:20, FastEthernet0/1
```

Обратите внимание, что система IOS отмечает маршрут буквой O, обозначая ею протокол OSPF. Помимо этого, маршрут выглядит почти идентично тому, который вы видели при настройке протокола EIGRP. Но когда вы просматриваете детали маршрута, обнаруживаются некоторые другие различия:

```
Router1#show ip route 172.31.70.0
Routing entry for 172.31.70.0/24
  Known via "ospf 1", distance 110, metric 2, type intra area
  Last update from 10.0.21.1 on FastEthernet0/1, 00:18:37 ago
  Routing Descriptor Blocks:
  * 10.0.21.1, from 2.2.2.2, 00:18:37 ago, via FastEthernet0/1
      Route metric is 2, traffic share count is 1
```

В разделе Routing Descriptor Blocks вы можете увидеть другую ссылку на адрес 2.2.2.2. Это позволяет легко определить, какое устройство извещает об этом маршруте. В данном случае это Коммутатор 2.

Последний шаг – настройка Коммутатора 1.

Практикум

Настройте протокол OSPF на Маршрутизаторе 1, выполнив следующие команды:

```
router ospf 1
Network 10.0.12.1 0.0.0.0 area 0
Network 172.31.60.0 0.0.0.255 area 0
Network 10.0.99.1 0.0.0.0 area 1
```

Проверьте результат, выполнив команду show ip ospf neighbor.

Вы должны увидеть две смежности: одну – с Маршрутизатором 1, а другую – с Коммутатором 2:

```
Switch1#show ip ospf neighbor

Neighbor ID     Pri   State     Dead Time   Address     Interface
12.12.12.12       1   FULL/DR   00:00:30    10.0.12.2   Vlan999
2.2.2.2           1   FULL/DR   00:00:30    10.0.99.2   FastEthernet0/24
```

Посмотрите, как легко определить, какое это устройство! Это одна из причин, по которой я предпочитаю протокол OSPF вместо EIGRP, особенно в крупных сетях. Если вы видите обе смежности, ваша конфигурация протокола OSPF завершена. Теперь пришло время проверить его работу!

Практикум

На Коммутаторе 2 выполните команду `show ip route ospf`, а затем команду `traceroute 172.31.60.254`.

Вы должны увидеть следующее:

```
Switch2#show ip route ospf
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O        10.0.12.0/30 [110/2] via 10.0.21.2, 00:04:33, FastEthernet0/12
      172.31.0.0/16 is variably subnetted, 3 subnets, 2 masks
O        172.31.60.0/24 [110/3] via 10.0.21.2, 00:03:37, FastEthernet0/12

Switch2#traceroute 172.31.60.254
Type escape sequence to abort.
Tracing the route to 172.31.60.254
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.21.2 8 msec 0 msec 0 msec
  2 10.0.12.1 0 msec *   0 msec
```

Чтобы добраться до сети `172.31.60.0/24`, пакет переходит в пункт `10.0.21.2` (Маршрутизатор 1), а затем в пункт `10.0.12.1` (Коммутатор 1). Общее во всех этих трех подсетях – все они находятся в зоне 0. Запомните, что протокол OSPF предпочитает сохранять трафик в пределах одной зоны, когда это возможно.

Но предположим, что путь через Маршрутизатор 1 недоступен. Протокол OSPF должен соответствующим образом настроить и перенаправить трафик по прямой линии связи между Коммутаторами 1 и 2. Давай попробуем!

Практикум

На Коммутаторе 2 отключите интерфейс FastEthernet0/12, который связан с Маршрутизатором 1:

```
Interface fa0/12
Shutdown
```

Вы должны увидеть следующее сообщение протокола OSPF:

```
%OSPF-5-ADJCHG: Process 1, Nbr 12.12.12.12 on FastEthernet0/12 from FULL to DOWN, Neighbor
Down: Interface down or detached
```

Выполните еще раз команду `show ip route ospf` и затем команду `traceroute 172.31.60.254`.

Вы должны увидеть другой маршрут с Коммутатором 1 в качестве следующего перехода:

```
Switch2#show ip route ospf
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O IA     10.0.12.0/30 [110/2] via 10.0.99.1, 00:00:28, FastEthernet0/24
      172.31.0.0/16 is variably subnetted, 3 subnets, 2 masks
O IA     172.31.60.0/24 [110/2] via 10.0.99.1, 00:00:28, FastEthernet0/24

Switch2#traceroute 172.31.60.254
Type escape sequence to abort.
Tracing the route to 172.31.60.254
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.99.1 9 msec *  0 msec
```

Таблица IP-маршрутизации определяет маршрут как `O IA`, что означает OSPF *`inter-area`* – межзонная маршрутизация. Это указывает на то, что для перехода в сеть `172.31.60.0/24` трафик должен перепрыгивать между зонами 0 и 1 OSPF, отсюда и термин «межзонный».

16.4. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`Interface loopback0`	Глобальный	Создает loopback-интерфейс с именем `loopback0`
`Router eigrp 7`	Глобальный	Открывает режим конфигурирования для автономной системы 7 протокола EIGRP
`Network 10.1.1.1 0.0.0.0`	Маршрутизатор EIGRP	Включает протокол EIGRP на любом интерфейсе с IP-адресом `10.1.1.1`. Также извещает о подсети интерфейса
`Show ip eigrp neighbor`	–	Показывает смежных соседей EIGRP
`Router ospf 1`	Глобальный	Открывает режим конфигурирования маршрутизатора для процесса протокола OSPF 1
`Network 10.1.1.1 0.0.0.0 area 0`	Маршрутизатор OSPF	Включает протокол OSPF на любом интерфейсе с IP-адресом `10.1.1.1`. Также извещает о подсети интерфейса в зоне 0
`Show ip ospf neighbor`	–	Отображает смежных соседей с протоколом OSPF
`Show ip protocols`	–	Отображает информацию об активных протоколах маршрутизации на маршрутизаторе

16.5. Практическое задание

Поместите подсеть `10.0.99.0/30` в зону 0 протокола OSPF. У вас есть несколько способов сделать это, поэтому используйте метод, который кажется вам наилучшим. Если вы добьетесь успеха, протокол OSPF предпочтет прямой путь между Коммутаторами 1 и 2. Как только вы это сделаете, добавьте loopback-интерфейсы на Коммутаторах 1 (`1.1.1.1`) и 2 (`2.2.2.2`) в зону 0.

Глава 17

Обнаружение устройств

В этой главе вы узнаете, как использовать оболочку командной строки для обнаружения устройств в вашей сети. Чтобы было понятно, я говорю не просто о том, чтобы выяснить общее положение устройства в пространстве. Я говорю об обнаружении устройства на конкретном коммутаторе или маршрутизаторе и порту, к которому он подключен. Хотя это не придется делать очень часто, но когда возникнет такая необходимость, зная, как обнаружить этот неуловимый принтер или зараженный вирусом компьютер, вы можете стать героем.

17.1. Сценарии обнаружения устройств

Я не буду обсуждать все возможные сценарии, которые могут использоваться для обнаружения определенного устройства. Но, опираясь на свой собственный опыт, на ум приходят два основных сценария.

Многие организации используют программы мониторинга сети, такие как WhatsUp Gold, для определения использования пропускной способности сети и времени безотказной работы. Когда в сети снижается производительность, ваш менеджер может обратиться к сетевому администратору (вам), чтобы узнать, какие из них – *основные* устройства, которые изначально используют максимальную пропускную способность. В целом, зная IP-адреса этих устройств, вы можете выяснить, где они находятся, особенно если это серверы в центре обработки данных, где устройства не особо часто перемещаются. Но иногда вы можете столкнуться с таинственным устройством – никто не знает, где оно находится и какое оно!

Другая ситуация связана с обнаружением принтера в удаленном офисе без ИТ-персонала. История обычно такая: кому-то нужно переместить принтер в другое место в офисе, что влечет за собой переподключение его на другой порт коммутатора. Все, что вам нужно сделать, – это перенастроить новый порт коммутатора так же, как и исходный. Проблема в том, что вы не знаете, к какому порту подключен принтер, поэтому вы не можете получить информацию о виртуальной сети, скорости и дуплексе.

Возможно, вы закатываете глаза и думаете, что настоящая проблема с этими кейсами – это неспособность правильно документировать сеть, и это правда.

К сожалению, ситуации с недостаточной или вовсе отсутствующей сетевой документацией более распространены, чем вы думаете. Я бы сказал, что в большинстве организаций без штатного сетевого администратора отсутствие сетевой документации – это правило, а не исключение.

Из-за этого обучение методам обнаружения устройств – не только косвенный, полезный навык. Это важно, чтобы иметь возможность документировать вашу сеть. К счастью, вам не нужно прыгать в самолет и посещать удаленные офисы и центры обработки данных – по крайней мере, можно не делать этого. Вместо этого вы можете легко отследить устройства из оболочки командной строки IOS, как правило, всего за пару минут или еще того меньше.

17.2. Этапы обнаружения устройства

Прежде чем начать, примите во внимание высокоуровневый процесс, который вы будете использовать каждый раз, ища устройство, которое я назову *целевым*.

17.2.1. Получение IP-адреса

Почти все конечные устройства в вашей сети – будь то принтер, сервер, ПК, IP-телефон или что-то еще, – имеют IP-адрес. Возможно, вам нужно будет определить устройство в удаленном офисе, не зная его IP-адрес. Способ, как вы получите IP-адрес для такого устройства, может быть различным в зависимости от того, что это за устройство. Если это компьютер или сервер, на котором вы можете выполнить команду `ifconfig` или `ipconfig`, определить IP-адрес вы можете самостоятельно без моей помощи.

17.2.2. Обнаружение устройства до последнего перехода

Последний переход – это последнее устройство, через которое проходит IP-пакет, когда достигает своего адресата. Как вы помните из предыдущей главы, это может быть маршрутизатор или коммутатор уровня 3.

17.2.3. Получение MAC-адреса

Как только вы доберетесь до устройства последнего перехода, будь то коммутатор или маршрутизатор, вы можете запросить таблицу ARP, чтобы определить MAC-адрес устройства. Напомню из главы 2, что ARP – это функция, которая сопоставляет IP-адрес MAC-адресу. Как только у вас есть MAC-адрес, вы можете отследить его вплоть до порта подключения.

В этой главе я покажу вам, как обнаружить два устройства – сервер и принтер. Если вы хотите последовать за мной, что я настоятельно рекомендую, следуйте инструкциям в «практикумах».

Практикум

Выберите два устройства в своей сети и запишите их IP-адреса.

17.3. Пример 1 – обнаружение сетевого принтера

Первое устройство, которое я обнаружу, – сетевой принтер с IP-адресом `172.31.60.50`. Хотя я уже знаю, где физически находится этот принтер (точно позади меня) и даже в какой виртуальной сети он находится, я буду притворяться, что не знаю. Я знаю только об устройствах, перечисленных в табл. 17.1. Давайте начнем!

Таблица 17.1. Интерфейсы и IP-адреса Коммутаторов 1 и 2 и Маршрутизатора 1

Устройство	Интерфейс	IP-адрес
Коммутатор 1	FastEthernet0/12	`10.0.12.1/30`
Коммутатор 1	FastEthernet0/24	`10.0.99.1/30`
Коммутатор 2	FastEthernet0/12	`10.0.21.1/30`
Коммутатор 2	FastEthernet0/24	`10.0.99.2/30`
Маршрутизатор 1	FastEthernet0/0.999	`10.0.12.2/30`
Маршрутизатор 1	FastEthernet0/1	`10.0.21.2/30`

17.3.1. Обнаружение последнего перехода с помощью команды `traceroute`

Обычно не имеет значения, где вы выполняете команду `traceroute`. Но, как правило, вам лучше начать с маршрутизатора, который находится где-то рядом со связующим звеном сети. В моей тестовой сети это будет Маршрутизатор 1. Для этого есть несколько причин. Во-первых, как сетевой администратор вы, скорее всего, будете знакомы с устройствами, которые составляют структуру вашей сети. Во-вторых, эти центральные маршрутизаторы часто видны большинству, если не для всей сети. В частности, они используют протоколы динамической маршрутизации и знают все маршруты к различным подсетям в вашей сети. Я начну с Маршрутизатора 1, чтобы найти принтер с IP-адресом `172.31.60.50`.

Практикум

Выполните команду `traceroute`, указав IP-адрес вашего первого целевого устройства:

```
traceroute 172.31.60.50
```

Вот какой вывод увидел я:

```
Router1#traceroute 172.31.60.50

Type escape sequence to abort.
Tracing the route to 172.31.60.50

  1 10.0.12.1 4 msec 0 msec 4 msec
  2 172.31.60.50 4 msec 4 msec 0 msec
```

Последний IP-адрес (172.31.60.50) – это целевое устройство, которое, как я знаю, не является маршрутизатором или коммутатором, поэтому я проигнорирую его. Вместо этого меня интересует последний переход, `10.0.12.1`.

В идеальном мире у меня была бы схема сети, которая подскажет мне, что это за устройство. В идеале мне нужно суметь авторизоваться на этом устройстве последнего перехода и продолжить поиск. К сожалению, этот мир далек от совершенства, поэтому я должен обдумать, что это за устройство. В этом поможет система IOS!

Практикум

На Маршрутизаторе 1 выполните команду `show ip route`, указав IP-адрес устройства последнего перехода:

```
show ip route 10.0.12.1
```

Вы должны увидеть несколько строк вывода:

```
Router1#show ip route 10.0.12.1
Routing entry for 10.0.12.0/30
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via FastEthernet0/0.999
      Route metric is 0, traffic share count is 1
```

Обратите внимание, что следующий переход напрямую связан с портом FastEthernet0/0.999. Это говорит мне, что устройство, к которому я подключен (Маршрутизатор 1), имеет интерфейс с IP-адресом в той же подсети, что и устройство последнего перехода. В моем примере это субинтерфейс FastEthernet0/0.999. Опять же, симулируя полное незнание сети, мне нужно выяснить, с чем связан этот интерфейс.

17.3.2. Протокол CDP

Один из ваших лучших приятелей при обнаружении устройств – это протокол *Cisco Discovery Protocol (CDP)*. С некоторыми заметными исключениями все маршрутизаторы и коммутаторы Cisco по умолчанию поддерживают протокол CDP. Как следует из названия, любое устройство, работающее с протоколом CDP, оповещает о себе и своих возможностях своих непосредственных соседей. Оповещения CDP содержат имя устройства, платформу и IP-адреса – более чем достаточно информации, чтобы определить, какие устройства находятся рядом с тем, с которым вы работаете.

Практикум

На Маршрутизаторе 1 выполните команду `show cdp neighbors`.

Вы должны увидеть список устройств, предполагая, что это устройства Cisco и на них запущен протокол CDP:

```
Router1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID    Local Intrfce   Holdtme    Capability   Platform   Port ID
Switch2      Fas 0/1         139        R S I        WS-C3560-  Fas 0/12
Switch1      Fas 0/0         145        R S I        WS-C3560-  Fas 0/12
```

Кстати, CDP – это проприетарный протокол компании Cisco, поэтому не рассчитывайте, что он покажет вам информацию об устройствах, разработанных другими компаниями. Обратите внимание, что Коммутатор 1 подключен к интерфейсу FastEthernet0/0, который является родительским для субинтерфейса FastEthernet0/0.999. Затем следующей остановкой является Коммутатор 1.

Дополнительно

По умолчанию оповещения CDP проходят через виртуальную сеть 1, и вы не можете отключить виртуальную сеть 1 на транковом интерфейсе. Вот почему на выходе отображается физический интерфейс, а не субинтерфейс.

17.3.3. Получение MAC-адреса устройства

Следующим шагом будет переход на устройство последнего перехода, которым в моем случае является Коммутатор 1, и запрос таблицы ARP, чтобы получить MAC-адрес принтера.

Практикум

Выполните команду `show arp`, дополненную IP-адресом искомого устройства:

```
show arp 172.31.60.50
```

Вы должны получить вывод, содержащий MAC-адрес:

```
Switch1#show arp 172.31.60.50
Protocol   Address      Age (min)   Hardware Addr   Type   Interface
Internet   172.31.60.50      2     0030.c1c3.80d0   ARPA   Vlan600
```

Как вы, наверное, догадались, `Hardware Addr` – это и есть MAC-адрес принтера. Но значение `Vlan600` в колонке `Interface` может показаться странным. `Vlan600` – это не физический интерфейс. Это SVI-интерфейс Коммутатора 1. Чтобы узнать физический интерфейс, в котором подключен принтер, нужно копнуть немного глубже.

17.3.4. Просмотр таблицы MAC-адресов

Узнав MAC-адрес целевого устройства, можно сузить поиск его местоположения до определенного порта. Для этого нужно запросить таблицу MAC-адресов.

Практикум

Выполните команду `show mac address-table address`, дополненную MAC-адресом:

```
show mac address-table address 0030.c1c3.80d0
```

Как вариант вы можете просмотреть все записи в таблице, введя команду `show mac address-table`.

Если вы просматриваете запись только для одного MAC-адреса, то должны увидеть что-то вроде этого:

```
Switch1#show mac address-table address 0030.c1c3.80d0
          Mac Address Table

-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 600    0030.c1c3.80d0    STATIC      Fa0/5
Total Mac Addresses for this criterion: 1
```

На выходе вы узнаете номер виртуальной сети, MAC-адрес и физический порт, к которому подключено устройство с данным MAC-адресом. Похоже, что принтер подключен напрямую к порту FastEthernet0/5, но, разумеется, рекомендуется проверить конфигурацию порта:

```
Switch1#sh run int fa0/5
Building configuration...

Current configuration : 256 bytes
!
interface FastEthernet0/5
 switchport access vlan 600
 switchport mode access
 switchport port-security maximum 3
 switchport port-security
 switchport port-security aging time 10
 switchport port-security violation restrict
 spanning-tree portfast
End
```

Это явно порт доступа в виртуальную сеть `600`, при этом функция Port Security включена. Хотя нет ничего, что прямо говорит «Это принтер!», по крайней мере вывод выглядит так, как будто это какое-то оконечное устройство. Давайте углубимся, взглянув на функцию Port Security:

```
Switch1#sh port-security interface fa0/5
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Restrict
Aging Time                 : 10 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 3
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0030.c1c3.80d0:600
Security Violation Count   : 0
```

В течение последних 10 минут функция Port Security обнаружила только один MAC-адрес, и это принтер. Исходя из этого, вы можете быть достаточно уверенным, что принтер подключен к порту FastEthernet0/5 на Коммутаторе 1!

Я использую слова «достаточно уверенным», потому что между Коммутатором 1 и принтером может быть другой коммутатор. Маленькие коммутаторы стороннего разработчика, а не Cisco, не используют протокол CDP или не образуют транки виртуальной сети, поэтому они могут быть практически невидимыми. Но если вы сузили свой поиск до одного порта, на котором висит только один MAC-адрес, то вы, вероятно, получили желаемый результат.

17.4. Обнаружение сервера

Теперь я найду Linux-сервер с IP-адресом `172.31.70.51`. Опять же, я сделаю вид, что не знаю, где находится этот сервер, чтобы проиллюстрировать все обычные шаги, которые необходимо выполнить. Я собираюсь повторить те же основные шаги, что и раньше, но на этот раз я немного сокращу свои объяснения. Кроме того, в этом разделе нет практикумов, потому что вы выполните все в практическом задании.

17.4.1. Обнаружение последнего перехода с помощью команды `traceroute`

Я снова авторизовался на Маршрутизаторе 1 и выполнил еще раз команду `traceroute`, указав на этот раз на IP-адрес сервера. Я получаю знакомый вывод:

```
Router1#traceroute 172.31.70.51

Type escape sequence to abort.
Tracing the route to 172.31.70.51

  1 10.0.21.1 4 msec 0 msec 0 msec
  2 172.31.70.51 4 msec 0 msec 0 msec
```

На этот раз последний переход - `10.0.21.1`. Посмотрим, где находится это устройство:

```
Router1#show ip route 10.0.21.1
Routing entry for 10.0.21.0/30
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via FastEthernet0/1
      Route metric is 0, traffic share count is 1
```

Это устройство, независимо от типа, напрямую связано с интерфейсом FastEthernet0/1 Маршрутизатора 1. Предполагая, что это устройство компании Cisco и включен протокол CDP, я могу выполнить команду `show cdp neighbors` и выяснить, что это за устройство:

```
Router1#sh cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID    Local Intrfce   Holdtme   Capability   Platform   Port ID
Switch2      Fas 0/1         164       R S I        WS-C3560-  Fas 0/12
Switch1      Fas 0/0         171       R S I        WS-C3560-  Fas 0/12
```

Кстати, обратите внимание, что для Коммутаторов 1 и 2 в колонке `Capability` указано значение `R`, т. е. оба устройства работают как маршрутизаторы. В этом случае, похоже, Коммутатор 2 является последним переходом. Но прежде чем перейти туда, я хочу обратиться к вопросу, который может вас тревожить.

Возможно, вам будет интересно, как бы я действовал, если бы Коммутатор 2 не был подключен напрямую, а отделен несколькими переходами от Маршрутизатора 1. В этом случае я мог бы использовать протокол Telnet или SSH и IP-адрес маршрутизатора, `10.0.21.1`. Чтобы проиллюстрировать это, я сделаю следующее:

```
Router1#telnet 10.0.21.1
Trying 10.0.21.1 ... Open

User Access Verification

Username: admin
Password:
Switch2#
```

Как только я авторизовался на Коммутаторе 2, я получил привилегированное приглашение, указывающее, что мой вход на Коммутатор 2 был успешным. Отсюда я могу продолжить поиск сервера.

17.4.2. Получение MAC-адреса устройства

Поскольку Коммутатор 2 является последним переходом, он должен иметь запись ARP для моего таинственного сервера:

```
Switch2#sh arp 172.31.70.51
Protocol    Address        Age (min)   Hardware Addr    Type   Interface
Internet    172.31.70.51          0    000c.295c.0254   ARPA   Vlan700
```

Разумеется, MAC-адрес сервера – `000c.295c.0254`. И, как и раньше, интерфейс является не физическим, а SVI-интерфейсом.

17.4.3. Просмотр таблицы MAC-адресов

Последний шаг – выяснить, к какому порту подключено устройство с этим MAC-адресом:

```
Switch2#sh mac address-table address 000c.295c.0254
          Mac Address Table

-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 700    000c.295c.0254    DYNAMIC     Fa0/22
Total Mac Addresses for this criterion: 1
```

В соответствии с выводом этот MAC-адрес доступен через интерфейс FastEthernet0/22. Но это не обязательно означает, что сервер напрямую подключен к этому порту. Между Коммутатором 2 и сервером может быть другой коммутатор или даже несколько. Давайте проверим конфигурацию порта:

```
Switch2#show run interface fa0/22
Building configuration...

Current configuration : 34 bytes
!
interface FastEthernet0/22
End
```

Раздел конфигурации интерфейса пуст, что означает, что порт использует конфигурацию по умолчанию. Чтобы узнать, работает ли он в качестве транка виртуальной сети или порта доступа, нам нужно копать немного глубже:

```
Switch2#show int fa0/22 switchport | i Mode|Trunk
Administrative Mode: dynamic auto
Operational Mode: trunk
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Capture Mode Disabled
```

FastEthernet0/22 – это транк виртуальной сети 802.1Q, но это необязательно означает, что устройство на другом конце является коммутатором. Напомню из главы 10, что гипервизоры, такие как VMware ESXi и Microsoft Hyper-V, также используют транки 802.1Q. Иногда вы даже можете обнаружить устройство конечного пользователя, такое как компьютер или IP-телефон, подключенный к транковому порту. Суть в том, что вы не должны быть уверены, что транко-

вый порт предоставляет окончательные сведения о том, какое устройство подключено на другом конце.

На данный момент я не уверен, был ли обнаружен порт, к которому подключен сервер. Давайте исследуем еще немного, изучив все MAC-адреса на этом порту:

```
Switch2#show mac address-table interface fa0/22
          Mac Address Table

-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 700    000c.295c.0254    DYNAMIC     Fa0/22
 700    000c.29d9.6546    DYNAMIC     Fa0/22
 700    001d.45cf.e817    DYNAMIC     Fa0/22
 700    00d0.b80d.6782    DYNAMIC     Fa0/22
 700    78e3.b50d.56e8    DYNAMIC     Fa0/22
 700    78e3.b510.acd2    DYNAMIC     Fa0/22
   1    001d.45cf.e817    DYNAMIC     Fa0/22
 600    001d.45cf.e817    DYNAMIC     Fa0/22
Total Mac Addresses for this criterion: 8
```

Неудивительно, что Коммутатор 2 знает о нескольких MAC-адресах в виртуальной сети `700`, которые доступны с этого порта. Также имеется один MAC-адрес, который появляется в виртуальных сетях `1`, `600` и `700` одновременно. Это верный признак, что устройство на другом конце является коммутатором. Если это коммутатор Cisco, протокол CDP должен сообщить нам об этом:

```
Switch2#show cdp neighbors fa0/22
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID    Local Intrfce   Holdtme   Capability   Platform   Port ID
Switch3      Fas 0/22        154         S I        WS-C3560-  Fas 0/21
```

Это похоже на другой коммутатор! Кстати, обратите внимание, что значение в колонке `Capability` не указывает, что Коммутатор 3 является маршрутизатором. Вот почему его IP-адрес не появился при выполнении команды `traceroute`! Кстати, как решить эту проблему, не зная IP-адрес устройства для управления? Протокол CDP также может помочь вам в этом:

```
Switch2#show cdp neighbors fa0/22 detail
-------------------------
Device ID: Switch3
Entry address(es):
  IP address: 192.168.1.103
Platform: cisco WS-C3560-24TS,  Capabilities: Switch IGMP
Interface: FastEthernet0/22,  Port ID (outgoing port): FastEthernet0/21
Holdtime : 150 sec
[output truncated]
```

Протокол CDP предоставляет IP-адрес управления коммутатора, 192.168.1.103. Давайте попробуем получить доступ к нему, организовав Telnet-сеанс:

```
Switch2#telnet 192.168.1.103
Trying 192.168.1.103 ... Open

User Access Verification

Username: admin
Password:
Switch3#
```

Бинго! Мы на один шаг ближе к поиску неуловимого сервера! Давайте проверим таблицу MAC-адресов Коммутатора 3:

```
Switch3#show mac address-table address 000c.295c.0254
          Mac Address Table

-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
 700    000c.295c.0254    DYNAMIC     Fa0/10
Total Mac Addresses for this criterion: 1
```

Похоже, сервер подключен к порту FastEthernet0/10 Коммутатора 3. Давайте проверим конфигурацию интерфейса:

```
Switch3#sh run int fa0/10
Building configuration...

Current configuration : 112 bytes
!
interface FastEthernet0/10
 description Linux server
 switchport access vlan 700
 switchport mode access
end
```

Ура! Сервер Linux подключен к порту FastEthernet0/10.

17.5. Команды, использованные в этой главе

Теперь, когда вы хорошо понимаете, как работает IP-маршрутизация, каковы принципы связи между IP-адресами и MAC-адресами, процесс определения местоположения устройства должен стать интуитивным. Вы всегда начинаете с IP-адреса, обнаруживаете его до последнего перехода, а затем определяете MAC-адрес. После того как вы узнали MAC-адрес, вы можете легко определить виртуальную сеть, в которой находится устройство, и к какому коммутатору оно подключено. А затем определить расположение конкретного порта уже несложно.

Выполняя практическое задание, обратитесь к командам в табл. 17.2.

Таблица 17.2. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`traceroute x.x.x.x`	–	Отображает переходы, через которые проходит IP-пакет, чтобы добраться до пункта назначения
`show cdp neighbors`	–	Отображает имя и информацию об интерфейсе непосредственно подключенного устройства Cisco
`show arp x.x.x.x`	–	Запрашивает таблицу ARP и отображает MAC-адрес указанного IP-адреса
`show mac address-table`	–	Отображает таблицу MAC-адресов

17.6. Практическое задание

Изучив материал этой главы, попробуйте найти второе целевое устройство, выбранное в начале главы. Используя команды из табл. 17.2, попытайтесь выяснить, к какому коммутатору и порту оно подключено.

Глава 18

Защита устройств Cisco

Если вы работаете в среде Windows Active Directory, то используете одну *учетную запись* – имя пользователя (логин) и пароль, чтобы авторизоваться почти везде. К сожалению, многие сети Cisco не приняли этот подход «один аккаунт для управления всеми». Вместо этого довольно часто для каждого устройства требуется собственный пароль администратора для авторизации в системе и внесения изменений. В более безопасных средах устройствам может потребоваться уникальное имя пользователя и пароль.

Один из недостатков такого подхода – в том, что когда вы хотите предоставить кому-то доступ к нескольким устройствам, вам необходимо вручную настроить учетные записи для каждого из них. Например, компания, в которой я когда-то работал, нанимала подрядчика для настройки IP-телефонии Cisco. Ему нужно было авторизоваться на нескольких наших маршрутизаторах, которые были разбросаны по всей стране, поэтому я создал индивидуальные привилегированные учетные записи только на тех маршрутизаторах, к которым ему был нужен доступ. Cisco называет каждую из этих учетных записей *локальной учетной записью пользователя*.

Хотя локальные учетные записи для каждого устройства – не идеальное решение, оно используется во многих компаниях. Как сетевому администратору оборудования Cisco вам нужно знать не только то, как создавать локальные учетные записи, но, что более важно, как блокировать доступ к устройствам Cisco для уменьшения ущерба, если такая привилегированная учетная запись попадает в чужие руки.

Один из наиболее эффективных способов сделать это – ограничить доступ через интерфейсы Telnet и Secure Shell (SSH) к определенным IP-адресам или подсетям. Например, если ваша организация поддерживает выделенную подсеть для ИТ-отдела, вы можете заблокировать доступ таким образом, чтобы лишь устройства из этой подсети могли использовать протоколы Telnet или SSH для ваших устройств Cisco.

В этой главе вы создадите привилегированную учетную запись пользователя на Коммутаторе 1. Затем вы настроите SSH-доступ и потребуете допустимое имя пользователя и пароль для управления коммутатором. В качестве бонуса вы также отключите доступ по протоколу Telnet, который по своей сути не-

безопасен, поскольку он не обеспечивает шифрование или аутентификацию. Наконец, вы заблокируете доступ к управлению для ряда подсетей. Давайте приступим!

18.1. Создание привилегированной учетной записи пользователя

Привилегированная учетная запись пользователя не требует, чтобы пользователь явно вводил пароль, дабы перейти в режим привилегированного управления. Как только привилегированный пользователь авторизуется в системе, IOS автоматически переводит его в привилегированный режим. Скоро вы создадите новую учетную запись на Коммутаторе 1, используя имя по вашему выбору. Вы сделаете эту учетную запись привилегированной, назначив уровень привилегий 15, максимально возможный. Хотя есть более низкие уровни, вы, вероятно, никогда не встретите их в работе. Я никогда не настраивал уровень привилегий, отличный от 15, при создании локальной учетной записи для доступа администратора. *Каждому уровню привилегий назначается определенный набор команд, которые допустимо использовать на этом уровне.*

Практикум

На Коммутаторе 1 выполните команду глобальной конфигурации, заменив значения *ben* и *cisco* на имя и пароль пользователя по вашему выбору:

```
username ben privilege 15 secret cisco
```

Дополнительно

Вместо ключевого слова `secret` вы можете использовать слово `password`. Разница в том, что триггеры безопасности IOS шифруют пароль и сохраняют зашифрованную версию в конфигурации. Таким образом, невозможно просмотреть пароль пользователя, просто анализируя текущую конфигурацию. Если используется ключевое слово `password`, то пароль хранится в конфигурации в незашифрованном виде. Это небезопасно. Чтобы скрыть пароль в текущей конфигурации, всегда используйте ключевое слово `secret`.

18.1.1. Проверка учетной записи

В рамках начальной настройки тестовой сети к Коммутаторам 1 и 2 и Маршрутизатору 1 есть доступ по протоколу Telnet. Помимо наличия сервера Telnet, система IOS имеет собственный встроенный клиент Telnet. Это может пригодиться, когда вы находитесь на одном устройстве и хотите протестировать подключение по протоколу Telnet к другому, не загружая программу PuTTY. Вы даже можете использовать встроенный клиент Telnet для подключения к устройству, на котором вы находитесь! Хотя обычно интерфейс Telnet для

доступа с Коммутатора 1 на Коммутатор 2 не используется, это простой способ проверить вашу конфигурацию.

Практикум

С Коммутатора 1 подключитесь по протоколу Telnet к интерфейсу loopback0 Коммутатора 2, который вы настроили ранее в главе 16 (адрес 1.1.1.1):

```
telnet 1.1.1.1
```

В приглашении введите имя и пароль, которые вы только что сохранили.

Когда вы выполните подключение по протоколу Telnet на Коммутатор 1, вы увидите запрос имени пользователя и пароля, примерно так:

```
Switch1#telnet 1.1.1.1
Trying 1.1.1.1 ... Open

User Access Verification

Username: ben
Password:
Switch1#
```

Вы авторизованы! Но не ясно, что фактически вы подключились по протоколу Telnet с Коммутатора 1 к Коммутатору 1, потому что приглашение выглядит точно так же, что и раньше. К счастью, есть способ выяснить, каким образом вы подключены.

Практикум

Выполните команду `show users`.

Вы должны увидеть следующее:

```
Switch1#show users
    Line       User       Host(s)       Idle       Location
   0 con 0                 1.1.1.1       00:00:00
*  1 vty 0     ben        idle          00:00:00   1.1.1.1

  Interface    User       Mode          Idle       Peer Address
```

В колонке `Line` указано значение `vty 0`. VTY означает виртуальный интерфейс *Virtual TeletYpe*, а линия VTY – это то, как IOS определяет отдельные сеансы Telnet и SSH на устройстве. Звездочка в начале строки указывает на виртуальный интерфейс VTY, к которому вы подключены. Если вы снова подключились по протоколу Telnet из текущего сеанса, то увидите второе соединение.

Практикум

С Коммутатора 1 выполните подключение по протоколу Telnet к Коммутатору 1 еще раз:

```
telnet 1.1.1.1
```

Авторизуйтесь, используя те же данные пользователя, что и ранее. После успешной авторизации снова выполните команду `show users`.

Вы должны увидеть новую запись:

```
Switch1#show users
    Line       User       Host(s)       Idle       Location
   0 con 0                 1.1.1.1       00:00:00
   1 vty 0     ben         1.1.1.1       00:00:00   1.1.1.1
*  2 vty 1     ben         idle          00:00:00   1.1.1.1

  Interface    User       Mode          Idle       Peer Address
```

Обратите внимание, что номер виртуального интерфейса VTY увеличился на 1. Следует отметить, что IOS позволяет одновременно выполнять несколько сеансов Telnet или SSH, и вы можете по отдельности идентифицировать каждый сеанс. Это не особенно интересно, но эта информация вскоре поможет вам при переконфигурировании Telnet и SSH. Кстати, максимальное количество одновременных сеансов ограничено количеством доступных линий VTY, и это число зависит от устройства. Я покажу вам, как получить эту информацию, далее.

18.2. Реконфигурация линий VTY

Одна из целей этой главы – включить протокол SSH, чтобы в конечном итоге вы могли отключить небезопасный доступ по протоколу Telnet. Однако прежде чем вы сможете сделать это, нужно понять, как в настоящее время настроены линии VTY.

Практикум

На Коммутаторе 1 выполните команду `show run | s line vty 0`.

Вы должны увидеть следующее:

```
Switch1#show run | s line vty 0
line vty 0 4
 login local
 transport input telnet
```

Давайте разберем эти строки конфигурации:

- `line vty 0 4` – представляет собой диапазон номеров линий VTY, от 0 до 4 включительно. Вы можете использовать встроенную справочную систему для уточнения максимально допустимого количества линий VTY. Введите `?` вместо `4`, и система IOS покажет вам самый большой допустимый номер линии VTY.

- `login local` – сообщает IOS, что нужно требовать локально настроенное имя пользователя и пароль у всех, кто подключается к любой из линий VTY.
- `transport input telnet` – разрешает доступ к коммутатору через интерфейс Telnet.

18.2.1. Включение доступа по SSH и запрет доступа по Telnet

Теперь вы включите SSH и отключите Telnet с помощью одной команды. Даже несмотря на то что вы подключены к Коммутатору 1, отключение Telnet не повлияет на текущие сеансы.

Практикум

Введите следующие команды, чтобы включить SSH и запретить Telnet:

```
line vty 0 4
transport input ssh
```

Теперь, когда вы отключили Telnet, вам понадобится способ для соединения через протокол SSH с Коммутатором 1 с Коммутатора 1. К счастью, сотрудники компании Cisco тоже подумали об этом и включили SSH-клиент в IOS.

Практикум

С Коммутатора 1 авторизуйтесь через интерфейс SSH на Коммутаторе 1, используя указанную ниже команду. Замените слово *ben* именем пользователя, которое вы задали ранее:

```
ssh -l ben 1.1.1.1
```

Как только авторизуетесь, выполните команду `show users`.

Система запросит пароль, как и при подключении по протоколу Telnet:

```
Switch1#ssh -l ben 1.1.1.1
Password:
Switch1#show users
    Line       User       Host(s)       Idle       Location
   0 con 0                 1.1.1.1       00:00:00
   1 vty 0     ben         1.1.1.1       00:00:00   1.1.1.1
   2 vty 1     ben         1.1.1.1       00:00:00   1.1.1.1
*  3 vty 2     ben         idle          00:00:00   1.1.1.1

  Interface    User       Mode          Idle       Peer Address
```

Обратите внимание, что существующие сеансы Telnet не затрагиваются. Отключение Telnet в текущей конфигурации не отключает подключенных пользователей. Эта защитная особенность может помочь вам избежать случайной блокировки!

Также обратите внимание, что нет никакой заметной разницы между сеансом SSH, подключенным к линии VTY 2, и двумя сеансами Telnet в строках 0 и 1. Если вы хотите увидеть только сеансы SSH, вам нужно выполнить другую команду.

Практикум

Выполните команду `show ssh`.

Вы должны увидеть следующее:

```
Switch1#show ssh
Connection  Version  Mode  Encryption  Hmac       State            Username
2           1.99      IN   aes128-cbc  hmac-sha1  Session started       ben
2           1.99      OUT  aes128-cbc  hmac-sha1  Session started       ben
%No SSHv1 server connections running.
```

В первой колонке указано значение `2`, обозначающее номер линии VTY. В последней колонке отображается имя пользователя.

Мы открыли два сеанса Telnet, чтобы убедиться, что вы случайно не заблокировали себя. Теперь, когда вы отключили доступ по протоколу Telnet и проверили SSH, можно безопасно закрыть все свои сеансы Telnet.

Практикум

Закройте сеанс SSH, выполнив команду `exit`.

Затем выполните команду `exit` дважды, чтобы закрыть два сеанса Telnet.

Используя встроенный клиент SSH, авторизуйтесь снова на Коммутаторе 1:

```
ssh -l ben 1.1.1.1
```

(По поводу имени *ben* вы помните.)

Наконец, проверьте, что все сеансы Telnet закрылись:

```
show users
```

Вы должны успешно подключиться и увидеть, что сеансов Telnet нет.

```
Switch1#show users
    Line       User       Host(s)       Idle       Location
   0 con 0                 1.1.1.1       00:00:00
*  1 vty 0     ben        idle          00:00:00   1.1.1.1

  Interface    User       Mode          Idle       Peer Address
```

18.2.2. Ограничение доступа по протоколу SSH с использованием списков доступа

Если привилегированное имя пользователя и пароль станут известны всем, повышается риск, что какой-нибудь злоумышленник внутри или за пределами

организации использует его для входа в ваши устройства Cisco и блокировки работы сети. Чтобы реализовать дополнительный уровень безопасности, вам следует рассмотреть возможность ограничения доступа к управлению этими IP-адресами, которые могут использоваться вами и другими авторизованными администраторами.

В следующей сети нужно ограничить доступ через интерфейс SSH к следующим подсетям и IP-адресам:

- подсеть VLAN1 – 192.168.1.0/24;
- Loopback-интерфейс Коммутатора 1 – 1.1.1.1;
- Loopback-интерфейс Коммутатора 2 – 2.2.2.2.

Из главы 9 вы узнали, как использовать списки доступа для блокировки трафика для отдельных интерфейсов. Хотя вы можете использовать этот метод для управления удаленным доступом к вашим устройствам Cisco, это большая работа, потому что вам придется применять список доступа к каждому интерфейсу, куда кто-либо может подключиться. Это также опасно, особенно в реальной сети, потому что одна неправильно настроенная запись ACL может заблокировать всю сеть.

Вместо того чтобы применять списки ACL к совокупности интерфейсов, вы можете применить их непосредственно к линиям VTY. Мало того, что это проще, но это также безопасно делать в реальной сети. Если вы допустили ошибку, самое худшее, что произойдет, – это то, что вы заблокируете последующие сеансы SSH. Но ваши существующие сеансы не будут затронуты.

Практикум

Создайте новый расширенный список доступа IP-адресов с именем Management:

```
ip access-list extended Management
permit ip 192.168.1.0 0.0.0.255 any
permit ip host 1.1.1.1 any
permit ip host 2.2.2.2 any
```

Примените список доступа к линиям VTY:

```
line vty 0 4
access-class Management in
```

Финальная часть каждой записи ACL – any – допускает сеансы SSH с указанными адресами источника на любой IP-адрес интерфейса на Коммутаторе 1. Это хорошо, поскольку, пока вы знаете один из активных IP-адресов устройства Cisco, вы можете использовать интерфейс SSH прямо на нем.

Практикум

Попробуйте подключиться через SSH с Коммутатора 2 к адресу loopback-интерфейса Коммутатора 1:

```
ssh -l ben 1.1.1.1
```

Даже если вы все настроили правильно, вы не сможете установить соединение:

```
Switch2#ssh -l ben 1.1.1.1
% Connection refused by remote host
```

Запомните, что разрешены только исходные адреса: `1.1.1.1` и `2.2.2.2`. Коммутатор 2 имеет несколько IP-адресов и, по-видимому, не использует IP-адрес своего интерфейса loopback1 – `2.2.2.2`. Чтобы исправить это, вам нужно явно указать IP-адрес этого интерфейса.

Практикум

На Коммутаторе 2 выполните следующую команду в режиме глобальной конфигурации:

```
ip ssh source-interface loopback 1
```

Попробуйте еще раз соединиться с Коммутатором 1 по протоколу SSH:

```
ssh -l ben 1.1.1.1
```

Теперь соединение должно быть установлено:

```
Switch2#ssh -l ben 1.1.1.1
Password:

Switch1#
```

18.3. Защищаем консольный порт

Вашей последней задачей будет защита на Коммутаторе 1 последовательного консольного порта, чтобы выводилось требование имени пользователя и пароля. По сравнению с защитой линий VTY, эта конфигурация практически примитивна.

Практикум

На Коммутаторе 1 выполните следующие команды в режиме глобальной конфигурации:

```
line con 0
login local
```

В основном я молчал о том, как вы подключались к своим устройствам. Если вы создали тестовую конфигурацию, то использовали консольный порт для входа в систему и установки исходных конфигураций. Если вы используете протокол Telnet или SSH для подключения к вашим устройствам, возможно, вы не сможете проверить консольный порт. Если у вас есть консольный порт, подключенный к Коммутатору 1, продолжайте выполнять практикум. В противном случае займетесь этим во время практической работы.

Практикум

Подключитесь к консольному порту Коммутатора 1, как об этом было рассказано при настройке тестовой сети, и войдите в систему, используя учетные данные, настроенные ранее в этой главе.
Затем выполните команду `show users`.

Вы должны увидеть следующее:

```
Switch1 con0 is now available

Press RETURN to get started.

User Access Verification

Username: ben
Password:
Switch1#show users
    Line       User       Host(s)       Idle       Location
   0 con 0     ben        idle          00:00:00

  Interface    User       Mode          Idle       Peer Address
```

Вместо линии VTY вы видите значение `con 0`, указывающее, что вы подключены к консольному порту.

18.4. Команды, использованные в этой главе

Обратитесь к табл. 18.1 для выполнения практического задания.

Таблица 18.1. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`username ben privilege 15 secret cisco`	Глобальный	Создает привилегированного пользователя с именем `ben` и зашифрованным паролем `cisco`
`telnet 1.1.1.1`	–	Сеанс Telnet с устройством по адресу `1.1.1.1` с помощью встроенного в систему IOS Telnet-клиента
`show users`	–	Показывает пользователей, подключенных по линиям VTY и консольному порту
`line vty 0 4`	Глобальный	Вход в режим конфигурирования линий VTY с 0 по 4 включительно
`transport input ssh`	Линия VTY	Включает протокол SSH и отключает Telnet
`access-class Management in`	Линия VTY	Применяет список доступа с именем `Management` к выбранным линиям VTY
`ssh -l ben 1.1.1.1`	–	Устанавливает SSH-соединение с устройством по адресу `1.1.1.1` с помощью SSH-клиента системы IOS
`show ssh`	–	Показывает пользователей, подсоединенных через SSH к линии VTY
`ip ssh source-interface loopback 1`	Глобальный	Задает исходный адрес, используемый в IOS SSH-клиентом

Окончание табл. 18.1

Команда	Режим конфигурирования	Описание
`line con 0`	Глобальный	Вводит в режим конфигурирования консоли
`login local`	Консольная линия	Требует локальное имя пользователя и пароль для входа через консольный порт

18.5. Практическое задание

Сейчас вы защитите Коммутатор 2 и Маршрутизатор 1. Выполните следующие шаги на обоих устройствах:

1) задайте имя пользователя и пароль;
2) включите протокол SSH и выключите доступ по протоколу Telnet. Убедитесь, что можете авторизоваться через интерфейс SSH;
3) обезопасьте консольный вход и убедитесь, что вы можете авторизоваться;
4) сохраните конфигурацию на Коммутаторах 1 и 2 и Маршрутизаторе 1.

Глава 19

Содействие устранению неполадок с помощью журналирования и отладки

Когда вы настраиваете сеть Cisco с нуля, то можете добиться успеха и бесперебойной работы при условии, что все настроили правильно. Но есть вероятность, что в повседневной работе вы не можете построить сеть с нуля. Вы наследуете существующую сеть, и у нее, вероятно, будет несколько неправильных настроек. В худшем случае сеть может быть даже неработоспособна. Но даже если ваша сеть не порадовала вас неправильной конфигурацией, маршрутизаторы и коммутаторы в конечном итоге выходят из строя. Порты перестают работать, система IOS сбоит, а источники питания отказывают.

Суть в том, что, в конце концов, вы столкнетесь с различными проблемами, которые связаны с сетью. Ниже представлены некоторые симптомы:

- компьютер не может получить IP-адрес, назначенный DHCP-сервером;
- пользователь не может получить доступ к сетевым ресурсам;
- на коммутаторе отсутствует виртуальная сеть;
- IP-маршрут отсутствует у группы маршрутизаторов.

Некоторые симптомы, очевидно, будут связаны с проблемой конфигурации маршрутизатора или коммутатора, тогда как другие могут быть двусмысленными. Прежде чем приступать к устранению неполадок в сети, вам нужно определить, находится ли источник проблемы где-то в вашей сети Cisco или в другом месте. Для этого вы будете использовать два инструмента: команды отладки и буфер регистрации событий (журналирования).

Команды отладки – это команды IOS, которые генерируют подробные сообщения о том, чем занимается конкретная технология в данный момент. Система IOS может выводить эти сообщения в оболочке командной строки или записывать их в буфер регистрации событий (журналирования). *Буфер регистрации событий* – это место в оперативной памяти, которое содержит выходные данные различных событий, таких как сбой интерфейса, новая смежность OSPF и, конечно же, отладочные сообщения.

В этой главе будет показано, как использовать инструменты регистрации событий и отладки для сбора подробной информации о некоторых технологиях, которые вы настраивали на протяжении всей книги. Идея состоит в том, чтобы помочь вам собрать достаточно информации, чтобы либо самостоятельно начать поиск неисправностей, либо заручиться помощью коллеги или вендора.

Отмечу, устранение неполадок по каждой из технологий, которые вы узнали, выходит за рамки этой книги. Я не буду описывать всех возможных способов отладки и интерпретировать все сообщения по отладке. Также не будет много «Практикумов». Если вы хотите углубленно изучить материалы по устранению неполадок, просмотрите ссылки на соответствующие учебные ресурсы в последней главе книги.

Прежде чем вы начнете, прочитайте предостережение: *не включайте режим отладки в реальной сети без получения соответствующего разрешения*. Включение режима отладки создает дополнительную нагрузку на центральный процессор, и хотя большинство задач отладки безопасно, некоторые могут перегружать маршрутизатор или коммутатор и приводить к прекращению передачи данных. Я предупрежу вас о более опасных командах отладки, но не выполняйте ни одну из команд из этой главы в реальной сети, пока она работает. Давайте начнем!

19.1. Настройка журналирования

По умолчанию система IOS хранит все отладочные сообщения в буфере регистрации событий, который по умолчанию ограничен размером в 4096 байт. Как только журнал заполняется, IOS начинает перезаписывать самые старые события. Я рекомендую проверить, правильно ли настроен буфер регистрации событий.

Практикум

Выберите маршрутизатор или коммутатор и выполните на нем следующую команду:

```
show logging
```

Вы должны увидеть нечто подобное:

```
Syslog logging: enabled (0 messages dropped, 0 messages rate-limited, 0 flushes, 0 overruns, xml
disabled, filtering disabled)

No Active Message Discriminator.


No Inactive Message Discriminator.

    Console logging: level debugging, 58 messages logged, xml disabled, filtering disabled

    Monitor logging: level debugging, 0 messages logged, xml disabled, filtering disabled
```

```
    Buffer logging: level debugging, 8 messages logged, xml disabled, filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    File logging: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 62 message lines logged
        Logging Source-Interface: VRF Name:

Log Buffer (4096 bytes):
```

Обратите внимание, что запись в консоль и буфер регистрации событий (`Console logging` и `Buffer logging`) осуществляется в режиме отладки (`debugging`). Это означает, что после включения отладки система IOS будет выводить отладочные сообщения в оболочку командной строки и запишет их в буфер регистрации событий. Также обратите внимание, что буфер регистрации событий ограничен размером в 4096 байт. Это соответствует примерно 50 строкам текста, что в большинстве случаев вполне достаточно. Но, на всякий случай, давайте увеличим его размером до 8192 байт.

Практикум

Выполните следующую команду в режиме глобальной конфигурации, чтобы система IOS передавала результаты отладки в буфер регистрации событий:

```
logging buffered debugging
```

Увеличьте размер буфера до 8192 байт:

```
logging buffered 8192
```

При желании вы можете очистить журнал, чтобы его было легче читать. Выполните следующую команду в режиме администратора:

```
clear logging
```

Кстати, как только буфер регистрации событий заполнится, система IOS начнет стирать самые старые события. Она не перестанет регистрировать новые события, не волнуйтесь. Как только вы все это сделаете, можно начать отладку!

19.2. Инструменты отладки

В большинстве случаев вы можете использовать встроенную функцию поддержки системы IOS, чтобы узнать, какие задачи отладки вам нужно включить. Но, как вы увидите, некоторые команды отладки неясны, или их поиск затруднен. Лучше всего поискать в Интернете команду, которая вам интересна. На следующих страницах я расскажу вам, как включать и использовать различные инструменты отладки, чтобы вы увидели процесс отладки в действии.

19.2.1. Отладка функции Port Security

В первом примере компьютер Начальство-ПК1 не может получить доступа к каким-либо сетевым ресурсам. После проверки, имеет ли компьютер физическое подключение к сети, вы проверяете, не блокирует ли функция Port Security MAC-адрес компьютера:

```
Switch2#show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
               (Count)       (Count)         (Count)
------------------------------------------------------------------------
     Fa0/21        1             1              172             Restrict
------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 6144
```

Похоже, что функция Port Security что-то блокирует, но не ясно, компьютер Начальство-ПК1 или другое устройство. Чтобы получить дополнительную информацию, вам нужно включить отладку функции Port Security.

Практикум

Включите отладку для Port Security, выполнив следующую команду в привилегированном режиме:

```
debug port-security
```

Пусть поработает с минуту, затем выключите ее:

```
undebug all
```

Теперь просмотрите буфер регистрации событий:

```
show logging
```

Вы увидите следующее:

```
*Mar  2 19:49:26.156: PSECURE: psecure_packet_enqueue: psecure receives
      a packet: addr = 0800.2759.d9fd, swidb = Fa0/21, vlan = 700,
      linktype = NullPak
*Mar  2 19:49:26.156: PSECURE: Read:197, Write:198
*Mar  2 19:49:26.156: PSECURE: swidb = FastEthernet0/21 mac_addr =
      0800.2759.d9fd vlanid = 700
*Mar  2 19:49:26.156: PSECURE: Violation/duplicate detected upon receiving
      0800.2759.d9fd on vlan 700: port_num_addrs 1 port_max_addrs
      1 vlan_addr_ct 1: vlan_addr_max 1 total_addrs 0: max_total_addrs 6144
*Mar  2 19:49:26.156: PSECURE: Security violation, TrapCount:197
```

Несмотря на то что не обошлось без непонятных моментов, в целом этот вывод довольно прост. Строка `Violation/duplicate detected upon receiving 0800.2759.d9fd on vlan 700` сообщает MAC-адрес, который блокирует функция Port Security,

и даже то, в какой виртуальной сети он используется. Вооружившись этой информацией, вы можете определить, принадлежит ли этот MAC-адрес устройству Начальство-ПК1 или другому компьютеру. Затем вы можете настроить защиту порта, чтобы разрешить этот адрес (подробнее см. главу 5).

19.2.2. Отладка DHCP-сервера

В этом примере компьютер Начальство-ПК1 не получает назначенный DHCP-сервером IP-адрес от Коммутатора 2. Чтобы добраться до сути, включите отладку DHCP-сервера. Предупреждение: эта отладка может содержать много информации – в зависимости от того, сколько DHCP-клиентов использует сервер. Не запускайте отладку в реальной сети, если вы не выполняете настоящую диагностику!

Практикум

Обнулите буфер регистрации событий:

```
clear logging
```

Включите отладку событий DHCP-сервера:

```
debug ip dhcp server events
```

Подождав примерно 10 секунд, выключите отладку:

```
undebug all
```

Теперь просмотрите буфер регистрации событий:

```
show logging
```

Ниже представлена выдержка из полного вывода:

```
*Mar  2 20:03:36.744: DHCPD: subnet [172.31.70.1,172.31.70.254]
      in address pool Executives is empty.
*Mar  2 20:03:36.744: DHCPD: Sending notification of ASSIGNMENT FAILURE:
*Mar  2 20:03:36.744: DHCPD: htype 1 chaddr 0800.2759.d9fd
*Mar  2 20:03:36.744: DHCPD: remote id 020a0000ac1f46fe0c000000
*Mar  2 20:03:36.744: DHCPD: interface = Vlan700
*Mar  2 20:03:36.744: DHCPD: class id 4d53465420352e30
*Mar  2 20:03:36.744: DHCPD: out_vlan_id 0
*Mar  2 20:03:36.744: DHCPD: Sending notification of ASSIGNMENT_FAILURE:
*Mar  2 20:03:36.744: DHCPD: due to: POOL EXHAUSTED
*Mar  2 20:03:36.744: DHCPD: htype 1 chaddr 0800.2759.d9fd
*Mar  2 20:03:36.744: DHCPD: remote id 020a0000ac1f46fe0c000000
*Mar  2 20:03:36.744: DHCPD: interface = Vlan700
*Mar  2 20:03:36.744: DHCPD: class id 4d53465420352e30
*Mar  2 20:03:36.744: DHCPD: out_vlan_id 0
```

Иногда приходится анализировать внушительный вывод, чтобы найти причину. В этом случае две строки, `pool Executives is empty` и `POOL EXHAUSTED`, – как раз те,

в которых приведена наиболее ценная информация. Компьютер Начальство-ПК1 не может получить IP-адрес, потому что в пуле DHCP недоступны адреса для назначения!

Стоит отметить, что команды `show` могут предоставить вам бóльшую часть той же информации, что и отладка. Например, команда `show ip dhcp pool` сообщит, что пул DHCP не имеет доступных адресов. Но он не укажет вам конкретный MAC-адрес компьютера, запрашивающий IP-адрес, назначенный DHCP-сервером, и не даст вам отметки времени, когда этот запрос возник. Только отладка может дать вам почти реальное отражение того, что происходит «за кадром» системы IOS.

19.2.3. Отладка протокола VTP

Большинство команд отладки довольно очевидны, но некоторые нужно отыскать. Команды отладки протокола VTP относятся как раз к таким.

В этом примере Коммутатор 1 является VTP-сервером, а Коммутатор 2 – клиентом. Коммутатор 2 не получает достоверную виртуальную сеть от Коммутатора 1. Команда для отладки событий VTP вложена в другой набор команд отладки, которые не столь интуитивно понятны.

Практикум

Введите следующую команду, чтобы включить отладку для событий VTP-сервера:

```
debug sw-vlan vtp events
```

Результат появится не так быстро, как в предыдущих случаях. Возможно, вам придется подождать вплоть до пяти минут, чтобы VTP-сервер отправил оповещения. После того как вы дали ему некоторое время на выполнение данной задачи, отключите отладку и проверьте содержимое журнала:

```
undebug all
show logging
```

Вы должны увидеть следующее:

```
*Mar  1 00:38:07.833: VTP LOG RUNTIME: Summary packet received,
      domain = cisco, rev = 11, followers = 1, length 77, trunk Fa0/24

*Mar 1 00:38:07.833: VTP LOG RUNTIME: Summary packet rev 11 greater than
      domain cisco rev 10

*Mar 1 00:38:07.833: VTP LOG RUNTIME: Domain cisco currently not in
      updating state

*Mar 1 00:38:07.833: VTP LOG RUNTIME: pdu len 77, #tlvs 1

*Mar 1 00:38:07.833: VTP LOG RUNTIME: Subset packet received, domain =
      cisco, rev = 11, seq = 1, length = 348

*Mar 1 00:38:07.833: VTP LOG RUNTIME: MD5 digest failing
```

```
calculated = B0 21 87 A1 4F 00 0C F4 14 7C C5 68 C6 84 A2 60
transmitted = B8 96 82 DE 39 52 A7 F7 6E 9C 60 EF 16 E4 77 23
```

Этот вывод немного сложнее расшифровать. Как правило, каждый раз, когда вы сталкиваетесь с незнакомой отладочной информацией и не уверены, что с ней делать, найдите слова, имеющие негативную окраску. Последний элемент в журнале говорит о сбое дайджеста MD5 (`MD5 digest failing`), означающем, что пароли VTP на стороне клиента и сервера не совпадают. Это не очевидно, если вы незнакомы с внутренней работой протокола VTP. В большинстве случаев вы можете скопировать и вставить название ошибки в поисковую систему и получить быстрый ответ.

19.2.4. Отладка IP-маршрутизации

В последнем примере я расскажу вам, как обнаружить изменения в таблице IP-маршрутизации. Как вы помните из главы 16, протоколы динамической маршрутизации, такие как OSPF и EIGRP, автоматически выбирают наилучший маршрут в сети в таблице IP-маршрутизации. Если соединение глобальной сети между маршрутизаторами ухудшается, это может вызвать проблемы с производительностью сети, особенно если соединение неоднократно прерывается и восстанавливается. Отладка IP-протоколов маршрутизации может точно сказать, когда произошли эти события.

Взгляните на маршруты по протоколу OSPF в таблице IP-маршрутизации на Коммутаторе 1:

```
Switch1#show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/2] via 10.0.99.2, 03:51:43, FastEthernet0/24
      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O        10.0.21.0/30 [110/2] via 10.0.99.2, 03:51:43, FastEthernet0/24
                      [110/2] via 10.0.12.2, 03:51:53, Vlan999
      172.31.0.0/16 is variably subnetted, 3 subnets, 2 masks
O        172.31.70.0/24 [110/2] via 10.0.99.2, 03:51:43, FastEthernet0/24
```

Обратите внимание, что сеть `2.2.2.2` доступна с интерфейса FastEthernet0/24 с адресом `10.0.99.2` (Коммутатор 2) в качестве следующего перехода. Чтобы увидеть, что происходит за кадром, когда это соединение сбоит, вы можете включить отладку для таблицы IP-маршрутизации.

Практикум

На маршрутизаторе или коммутаторе 3-го уровня включите отладку IP-маршрутизации:

```
debug ip routing
```

Посмотрите, что произойдет, если соединение на Коммутаторе 2 – FastEthernet0/24 – нарушится:

```
Switch1#debug ip routing
IP routing debugging is on
Switch1#
00:20:19: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
     changed state to down
00:20:20: %LINK-3-UPDOWN: Interface FastEthernet0/24, changed state to down
*Mar  1 00:20:20.232: is_up: FastEthernet0/24 0 state: 0 sub state: 1 line: 0
00:20:20: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/24 from
     FULL to DOWN, Neighbor Down: Interface down or detached
*Mar  1 00:20:20.232: RT: interface FastEthernet0/24 removed from routing table
*Mar  1 00:20:20.232: RT: del 10.0.99.0 via 0.0.0.0, connected metric [0/0]
*Mar  1 00:20:20.232: RT: delete subnet route to 10.0.99.0/30
*Mar  1 00:20:20.232: RT: del 10.0.99.1 via 0.0.0.0, connected metric [0/0]
*Mar  1 00:20:20.232: RT: delete subnet route to 10.0.99.1/32
*Mar  1 00:20:23.688: RT: updating ospf 2.2.2.2/32 (0x0): via 10.0.12.2 Vl999
*Mar  1 00:20:23.688: RT: closer admin distance for 2.2.2.2, flushing 1 routes
*Mar  1 00:20:23.688: RT: add 2.2.2.2/32 via 10.0.12.2, ospf metric [110/3]
*Mar  1 00:20:23.688: RT: updating ospf 172.31.70.0/24 (0x0): via 10.0.12.2 Vl999
*Mar  1 00:20:23.688: RT: closer admin distance for 172.31.70.0, flushing 1 routes
*Mar  1 00:20:23.688: RT: add 172.31.70.0/24 via 10.0.12.2, ospf metric [110/3]
*Mar  1 00:20:23.688: RT: del 10.0.21.0 via 10.0.99.2, ospf metric [110/2]
```

Коммутатор 1 удаляет подсеть `10.0.99.0/30` из своей таблицы IP-маршрутизации, как показывает запись `delete subnet route to 10.0.99.0/30`. После этого Коммутатор 2 обновляет маршрут для адреса `2.2.2.2/32`, чтобы использовать другой путь через `10.0.12.2`.

Имейте в виду, что вывод отладки IP-маршрутизации, вероятно, не будет иметь большого смысла, если вы не взглянете на сетевую диаграмму. Важно то, что вы знаете, как включить отладку и просмотреть ее вывод, если вам когда-либо это понадобится.

19.3. Уровни важности событий

В начале главы я сказал вам, что по умолчанию система IOS хранит все отладочные сообщения в буфере регистрации событий. Но вы, возможно, заметили, что система IOS также записывает некоторые сообщения в журнал без включенного режима отладки.

Практикум

Выберите коммутатор или маршрутизатор и перезагрузите его. Не сохраняйте текущую конфигурацию!

```
reload
Proceed with reload? [confirm]
00:03:06: %SYS-5-RELOAD: Reload requested by admin on console. Reload
Reason: Reload command.
```

Когда он загрузится, просмотрите буфер регистрации событий:

```
show logging
```

Вы можете увидеть десятки сообщений наподобие следующих:

```
*Mar  1 00:00:33.546: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1,
     changed state to down
*Mar  1 00:00:35.508: %SPANTREE-5-EXTENDED_SYSID: Extended SysId enabled for type vlan
*Mar  1 00:00:37.295: %DC-4-FILE_OPEN_WARNING: Not able to open flash:/
     dc_profile_dir/dc_default_profiles.txt
*Mar  1 00:00:37.295: %DC-6-DEFAULT_INIT_INFO: Default Profiles DB not loaded.
00:00:38: %SYS-6-CLOCKUPDATE: System clock has been updated from 00:00:38 UTC
     Mon Mar 1 1993 to 20:00:38 EST Sun Feb 28 1993, configured from console by console.
00:00:39: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
     changed state to down
00:00:39: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan600, changed state to down
00:00:39: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan999, changed state to down
00:00:39: %SYS-5-CONFIG_I: Configured from memory by console
```

Имейте в виду, что задачи отладки отключены и буфер регистрации событий очищается при перезагрузке, поэтому это не отладочные сообщения. Большинство из этих сообщений – уведомления. Термин *уведомления* относится к одному из восьми *уровней важности события*, или, для краткости, *протоколирования*. Чтобы узнать, что они собой представляют, выполните практикум.

Практикум

Просмотрите встроенную справку, выполнив команду `logging buffered ?` в режиме глобальной конфигурации.

Система IOS отображает соответствующую встроенную справку следующим образом:

```
Switch1(config)#logging buffered ?
  <0-7>              Logging severity level
  <4096-2147483647>  Logging buffer size
  alerts             Immediate action needed           (severity=1)
  critical           Critical conditions               (severity=2)
  debugging          Debugging messages                (severity=7)
  discriminator      Establish MD-Buffer association
```

```
  emergencies        System is unusable               (severity=0)
  errors             Error conditions                 (severity=3)
  filtered           Enable filtered logging
  informational      Informational messages           (severity=6)
  notifications      Normal but significant conditions (severity=5)
  warnings           Warning conditions               (severity=4)
  xml                Enable logging in XML to XML logging buffer
  <cr>
```

Восемь уровней регистрации событий нумеруются в порядке возрастания от самых важных до наименее важных. Например, аварийные ситуации (`emergencies`) имеют уровень важности 0, тогда как уведомления (`notifications`) имеют уровень важности 5. Самый высокий уровень 7 используется для отладки. Важно отметить, что каждый уровень регистрации неявно включает в себя все уровни ниже него. Например, когда вы устанавливаете уровень ведения журнала на уровень 7, отладка (`debugging`) IOS будет записывать сообщения и с более низким уровнем важности.

Практикум

Выберите маршрутизатор или коммутатор и задайте уровень важности событий, равный 4 (`warnings`):

```
logging buffered warnings
```

Сохраните текущую конфигурацию и перезагрузитесь:

```
write memory
reload
```

Наконец, просмотрите буфер регистрации событий:

```
show logging
```

Вы должны увидеть гораздо меньше выходных данных:

```
*Mar  1 00:00:37.337: %DC-4-FILE_OPEN_WARNING: Not able to open flash:/
      dc_profile_dir/dc_default_profiles.txt
00:00:42: %LINK-3-UPDOWN: Interface FastEthernet0/12, changed state to up
```

Обратите внимание, что каждая запись содержит символ `%`, за которым следуют прописные буквы, дефис и номер. Номер указывает на уровень важности. Например, событие `%LINK-3-UPDOWN` имеет уровень важности 3, `errors` (ошибки).

19.4. Настройка syslog-сервера

Буфер регистрации событий – хороший инструмент, но имеет существенный недостаток: он хранится в оперативной памяти, т. е. он ограничен по размеру, и удаляется, когда устройство перезагружается или отключается. Если вам нужно проверить журналы на нескольких устройствах, авторизация на каждом из них и просмотр буфера регистрации событий могут стать утомительным

и трудоемким процессом. Чтобы упростить жизнь, вы можете настроить на устройствах Cisco передачу журналов на syslog-сервер.

Настройка syslog-сервера выходит далеко за рамки этой книги, но большинство платных систем сетевого мониторинга включают в себя службу syslog-сервера. Существуют также бесплатные приложения, например Kiwi. Syslog-сервер подключается к сети и постоянно получает журналы с устройств, настроенных для их отправки. Как и в буфере регистрации событий, вы указываете уровни важности события для управления, какие типы сообщений системы IOS отправляются на syslog-сервер.

Чтобы настроить маршрутизатор или коммутатор для отправки журналов на syslog-сервер, вам понадобится следующее:

- IP-адрес syslog-сервера;
- уровень важности событий.

Практикум

Выберите маршрутизатор или коммутатор и настройте его так, чтобы он пересылал журналы на syslog-сервер с IP-адресом `1.2.3.4`:

```
logging host 1.2.3.4
00:27:11: %SYS-6-LOGGINGHOST_STARTSTOP: Logging to host 1.2.3.4 port 514
started - CLI
```

Установите уровень важности 7, отладка (`debugging`):

```
logging trap debugging
```

Проверьте конфигурацию, выполнив команду `show logging`.

Начиная примерно с 20-й строки вывода, вы должны увидеть следующее:

```
Trap logging: level debugging, 29 message lines logged
        Logging to 1.2.3.4  (udp port 514, audit disabled, link up),
              2 message lines logged,
              0 message lines rate-limited,
              0 message lines dropped-by-MD,
              xml disabled, sequence number disabled
              filtering disabled
        Logging Source-Interface: VRF Name:
```

Включение syslog-сервера не прекращает вывода событий в буфер регистрации. Для удобства рекомендуется указывать один и тот же уровень важности событий как для буфера регистрации, так и для syslog-сервера.

19.5. Команды, использованные в этой главе

По большей части команды ведения журнала и отладки интуитивно понятны. В табл. 19.1 перечислены те, которые вы видели в этой главе. Как обычно, используйте встроенную справочную систему для поиска определенных команд отладки по мере необходимости.

Таблица 19.1. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`show logging`	–	Показывает установки журналирования и содержание буфера регистрации событий
`logging buffered debugging`	Глобальный	Устанавливает уровень важности событий на уровень отладки, т. е. 7
`logging buffered 8192`	Глобальный	Устанавливает размер буфера регистрации событий, равный 8192 байтам
`clear logging`	–	Очищает буфер регистрации событий
`debug port-security`	–	Включает отладку функции Port Security
`undebug all`	–	Отключает отладку всех видов
`debug sw-vlan vtp events`	–	Включает отладку событий VTP
`debug ip routing`	–	Включает отладку для таблицы маршрутизации IP
`reload`	–	Перезагружает устройство
`logging buffered warnings`	Глобальный	Устанавливает уровень важности 4 для буфера регистрации событий
`logging host 1.2.3.4`	Глобальный	Отсылает журналы на syslog-сервер по адресу `1.2.3.4`
`logging trap debugging`	Глобальный	Устанавливает уровень 7 важности событий для syslog-сервер

19.6. Практическое задание

Вы можете выполнить команду `undebug all`, чтобы отключить все отладки. Но по какой-то непостижимой причине компания Cisco позволила одновременно включать все отладки, используя команду `debug all`. В этом задании вы исследуете последствия включения всех типов отладки сразу.

Выполните следующие шаги на своем тестовом маршрутизаторе или коммутаторе. Не выполняйте ни одну из этих команд на реальном оборудовании!

1. Включите вывод отладочных сообщений в оболочку командной строки, выполнив команду `logging console debugging`. Это приведет к тому, что система IOS выведет все сообщения отладки в оболочку командной строки, чтобы вы могли просматривать его в режиме реального времени.
2. Выполните команду `debug all`. Что произошло?
3. Быстро выполните команду `undebug all` и понаблюдайте, сколько времени требуется, чтобы сообщения отладки прекратили выводиться.
4. Предположим, что у вас есть коммутатор, настроенный как DHCP-сервер, но клиенты не получают от него IP-адрес. Какие команды отладки могут помочь вам понять, в чем дело?

Глава 20

Восстановление после сбоя

В своей карьере в качестве сетевого администратора Cisco вы столкнетесь с ситуациями, когда большие части сети или вся сеть необъяснимо перестают работать. В предыдущей главе вы узнали о некоторых методах устранения неполадок, связанных с конкретными технологиями, где наблюдается сбой в работе.

Когда вам необходимо восстановить сеть после сбоя, у вас может не быть такой шикарной возможности, как глубокое исследование произошедшей катастрофы. Вам нужно вернуть сеть в работоспособное состояние как можно быстрее! Цель восстановления после сбоя – не вернуть все в исходное состояние, но заставить все работать с тем, что есть. Перефразируя: вы не собираетесь выиграть спортивные соревнования, а просто пытаетесь вернуться к спортивной форме.

Упражнения в этой главе предназначены для применения только по мере необходимости, когда начальство вашей организации требует, чтобы вы сделали все возможное, дабы восстановить сеть. Это одна из последних глав книги, и упражнения в этой главе должны использоваться в качестве крайней меры.

Вот основные шаги, которые вам следует выполнить для восстановления после сбоя.

1. Сократите проблему до группы IOS-устройств. Маловероятно, что все или даже большинство ваших сетевых устройств будут работать неправильно одновременно. Скорее всего, одно или несколько устройств вызывают проблему.
2. Перезагрузите устройства по очереди. Хотя мы, ИТ-специалисты, часто недовольны советом по устранению неполадок типа «сначала перезагружаемся», иногда это оправдано. В этом случае мы предполагаем, что ваша организация дала указание вам сделать все возможное, чтобы восстановить сетевое подключение.
3. Если проблема не устранена, возможно, вам придется удалить конфигурацию запуска одного или нескольких устройств и перенастроить их с нуля. Это самый болезненный и наименее желательный вариант, но как сетевой администратор Cisco вы должны быть к этому готовы.

Давайте начнем!

20.1. Ограничьте область поиска подмножеством устройств

Каждый раз, когда вы имеете дело с крупномасштабным отключением сети, есть один шаг по устранению неполадок, который вам нужно выполнить независимо от того, сколько людей кричит на вас, что нужно восстановить сетевое подключение. Вам нужно сократить область поиска источника проблемы до нескольких IOS-устройств. Например, если все компьютеры в офисе потеряли сетевое подключение, вам нужно будет проверить каждое устройство, подключенное к каждому коммутатору, а также каждый маршрутизатор в этом офисе. С другой стороны, если отключение сети ограничивается отделом в офисе, вы можете ограничить поиск источника проблемы одним или двумя коммутаторами.

Если отключение сети влияет на несколько офисов одновременно, вероятность того, что проблема связана с определенной группой маршрутизаторов и коммутаторов, маловероятна. Скорее всего, возникла проблема с глобальными соединениями между офисами. В этом случае вы не можете сделать ничего, кроме как связаться с вашим провайдером и попросить его выполнить свои обязанности.

Предостережение: если вы не можете сократить проблему до некоего подмножества устройств, не выполняйте остальные шаги. Самое худшее, что вы можете сделать, – начать хаотичную работу с устройствами. Внедрение радикальных изменений в надежде на то, что сеть заработает, неизбежно приведет к ухудшению проблемы или вызовет новую проблему, которая проявится позже.

20.2. Перезагрузка устройства

Как только вы сократили проблему до некой группы устройств (или, если вам повезло, всего до одного устройства), следующий шаг – перезагрузить его командой `reload`.

Перезагрузка коммутатора или маршрутизатора отличается от перезагрузки компьютера. Это универсальная технология «попробуйте в первую очередь», которая имеет репутацию рабочего устранения неприятностей в большинстве случаев. Некоторым технологическим пуританам не нравится идея перезагрузки в качестве первого шага. Но в организации с деструктивным отключением сети у вас нет времени, чтобы следить за научным процессом устранения неполадок. Наиболее важной является работоспособность сети.

Перезагрузка делает одну очень полезную вещь: она возвращает ваше устройство к конфигурации запуска. Если вы (или кто-то еще) изменили что-либо в текущей конфигурации без сохранения, перезагрузка сбросит изменения и может даже устранить проблему. Перезагрузка также может очищать таблицы ARP и MAC-адресов, сбрасывать записи функции Port Security, восстанавливать протоколы маршрутизации, транковые соединения и агрегирован-

ные каналы, а также сбрасывать множество других параметров – любой из них может помочь решить проблему.

Возвращаясь к моему предыдущему предупреждению об осторожности, важно помнить, что последствия перезагрузки могут повлиять на другие устройства, помимо того, которое вы перезагружаете, по крайней мере временно. Следовательно, возможно, что перезагрузка коммутатора или маршрутизатора приведет к ухудшению работы сети, прежде чем ситуация снова улучшится. Например, если вам нужно перезагрузить маршрутизатор с протоколом OSPF, другие маршрутизаторы с протоколом OSPF, подключенные к нему, на мгновение должны будут перенаправить свой трафик по другому пути. Если окольные пути не будут обнаружены, у этих маршрутизаторов не останется выбора, кроме как отключить передачу данных.

Практикум

Перезагрузите какое-либо устройство или коммутатор в вашей тестовой сети, используя следующую команду:

```
Reload
```

Устройство потребует подтверждения и затем немедленно перезагрузится:

```
Router1#reload
Proceed with reload? [confirm]
```

Вы должны сразу увидеть вывод, указывающий на ожидающую перезагрузку:

```
*Nov 12 03:14:34.142: %SYS-5-RELOAD: Reload requested by admin on console.
     Reload Reason: Reload Command.
```

20.2.1. Перезагрузка по расписанию

В случае проблемы с сетью, решение которой может подождать, когда сотрудники уйдут домой (включая вас), у вас есть возможность запланировать перезагрузку либо в определенное время, либо через определенное количество минут. Это особенно удобно, если ваша организация требует отложить перезагрузку до некоторого запланированного часа, например в ночное время.

Практикум

На любом устройстве в вашей тестовой сети запланируйте перезагрузку через 15 минут, начиная с сего момента:

```
Reload in 15
```

Вновь будет выведено сообщение о подтверждении:

```
Switch2#reload in 15
Reload scheduled for 22:01:04 UTC Fri Nov 11 2016 (in 15 minutes) by admin on console
Proceed with reload? [confirm]
```

```
Switch2#
Nov 11 21:46:06.073: %SYS-5-SCHEDULED_RELOAD: Reload requested for 22:01:04
     UTC Fri Nov 11 2016 at 21:46:04 UTC Fri Nov 11 2016 by admin on console.
```

Вы можете указать время перезагрузки в минутах или часах:минутах. Например, если вы хотите перезагрузить устройство через 1 час и 15 минут, нужно выполнить команду `reload in 1:15`. С другой стороны, используя команду `reload in`, вы можете не блокировать свой доступ при внесении изменений на удаленном устройстве. Например, если вы подключились к маршрутизатору в удаленном офисе, вы можете запланировать перезагрузку через несколько минут, и тут же внести изменения в конфигурацию. Если ваши изменения приведут к потере соединения, вам придется подождать пару минут, пока маршрутизатор перезагружается. Как только он перезагрузится, он отменит внесенные изменения, и вы сможете вновь подключиться к нему и повторить попытку.

Но что делать, если изменения привели к положительному результату? В этом случае вам нужно отменить или прервать перезагрузку.

Практикум

На том же устройстве, где вы записали в расписание перезагрузку, выполните следующую команду, чтобы ее отменить:

```
Cancel reload
```

Система IOS немедленно обнулит отсчет и *не* перезагрузит устройство:

```
Switch2#reload cancel

***
*** --- SHUTDOWN ABORTED ---
***

Nov 11 21:46:38.721: %SYS-5-SCHEDULED_RELOAD_CANCELLED: Scheduled reload
     cancelled at 21:46:38 UTC Fri Nov 11 2016
```

Если вы уточнили, какое устройство вызывает проблему, а перезагрузка не исправляет ситуацию, это может указывать на то, что проблема связана с конфигурацией запуска, которая не может быть исправлена путем перезагрузки.

В этой ситуации рекомендуется сохранить копии рабочих конфигураций. Если одно из устройств под управлением IOS не работает должным образом даже после перезагрузки, вы сможете сравнить конфигурацию запуска на устройстве с вашей копией. Напомню, что когда маршрутизатор или коммутатор загружается, система IOS копирует конфигурацию запуска в текущую конфигурацию, поэтому после перезагрузки обе должны быть идентичны. Если вы обнаружите какие-либо расхождения между текущей конфигурацией и резервной копией, то можете исправить их и, надеюсь, решить вашу проблему. (Добавлю: копирование текущей конфигурации в резервную копию выполняется командой `copy running-config flash:`*`резервный-файл`*, где *`резервный-файл`* – файл с именем на ваш выбор. Рекомендую указывать в имени дату изменения.)

Но если у вас нет резервной копии работающей конфигурации запуска, тогда вам предстоит то, что я называю крайней мерой, – удаление конфигурации запуска.

20.3. Удаление конфигурации запуска

Удаление конфигурации запуска на устройстве под управлением операционной системы IOS означает, что вам нужно перенастроить устройство с нуля. Это не то, что вы делаете волей-неволей. Прежде чем решиться на этот шаг, вы должны убедиться, что потребуется больше времени на изменение неправильной конфигурации, чем на начало с нуля. Другими словами, это не для слабонервных!

Удаление конфигурации запуска осуществляется довольно просто, а процесс одинаков как для маршрутизаторов, так и для коммутаторов. Если вы следовали руководству по настройке тестового окружения и выполнили все упражнения в этой книге, у вас есть довольно хорошее представление о том, как начинать с нуля. Я не буду повторять все шаги здесь. Просто имейте в виду, что удаление конфигурации запуска означает, что почти ничего не будет работать, пока вы не настроите устройство.

Практикум

Выберите устройство, на котором вы хотите удалить конфигурацию запуска. В целях безопасности сохраните текущую конфигурацию в файл. Просмотрите конфигурацию запуска устройства, выполнив команду:

```
show startup
```

Скопируйте весь вывод в текстовый редактор.
Удалите конфигурацию запуска с помощью следующей команды в привилегированном режиме:

```
Delete nvram:startup-config
```

После введения команды IOS попросит вас ввести имя файла и подтвердить действие.
Дважды нажмите клавишу **Enter**.

Вы должны увидеть следующее:

```
Router1#delete nvram:startup-config
Delete filename [startup-config]?
Delete nvram:startup-config? [confirm]
[OK]
```

Энергонезависимая память произвольного доступа (NVRAM) на устройствах под управлением IOS хранит конфигурацию запуска, а также базу данных виртуальных сетей. Предыдущая команда удаляет из памяти NVRAM не все дан-

ные, а только файл конфигурации запуска. Поскольку система IOS использует текущую конфигурацию в настоящее время, удаление конфигурации запуска не влияет на работу, пока вы не перезагрузите устройство.

После перезагрузки у устройства будет отсутствовать IP-адрес управления. Это означает, что единственный доступ к нему – через последовательную консоль, как и при настройке тестовой сети.

Практикум

Если необходимо перейти к настройке тестового окружения, подключите компьютер к последовательной консоли устройства, с которого вы только что удалили конфигурацию запуска.
Перезагрузите устройство, выполнив команду `reload`.

Как только оно загрузится, вы сможете авторизоваться в привилегированном режиме без пароля:

```
Router>enable
Router#
```

Обратите внимание, что предыдущее имя маршрутизатора – `Router1` – отсутствует, и теперь он идентифицирован как просто `Router`. На этом этапе вы можете настроить маршрутизатор с нуля.

20.4. Сброс пароля

Другая интересная проблема возникает чаще, чем вы думаете: невозможность авторизоваться на устройстве, потому что вы не знаете пароль. Это не катастрофа, но может помочь при возникновении фактической катастрофы, если она когда-либо случится. Процесс сброса пароля различен для маршрутизаторов и коммутаторов, но в обоих случаях вам нужно использовать последовательную консоль (как и при настройке тестовой сети) для его сброса.

Лично мне никогда не приходилось сбрасывать пароль. Но я приобрел подержанные устройства с набором паролей, которые мешали мне авторизовываться в системе. Это наиболее вероятный сценарий, с которым вы столкнетесь: вы приобретаете устройство у кого-то и не знаете пароль.

Важно отметить, что сброс пароля в том виде, в котором я собираюсь описать, – это удаление конфигурации запуска. Как правило, вы не должны выполнять эти шаги на устройстве, которое работает корректно.

Процесс сброса пароля несколько отличен для маршрутизаторов и коммутаторов. В обоих случаях вам нужно авторизоваться в режиме *Read-only Memory Monitor (ROMMON)*. Режим ROMMON позволяет обойти обычный процесс загрузки, чтобы устройство не считывало конфигурацию запуска, а вместо этого вводило вас в конфигурацию запуска по умолчанию. Здесь вы можете свободно вносить любые изменения, в том числе создавать имена пользователей и па-

роли. После внесения изменений вы сохраните новую конфигурацию запуска с перезаписью старой и, наконец, перезагрузите устройство.

Разница между сбросом пароля на маршрутизаторе и коммутаторе заключается в том, что вам нужно нажать кнопку выбора режима на передней панели коммутатора, чтобы перейти в режим ROMMON, тогда как на маршрутизаторе вам нужно отправить специальную команду *break*, используя оболочку командной строки. Учитывая, что вам все равно придется подключаться к последовательной консоли коммутатора, нажатие одной кнопки на передней панели коммутатора не должно стать большой проблемой. На самом деле гораздо проще попасть в ROMMON на коммутаторе, чем на маршрутизаторе, поэтому начнем с самой сложной задачи.

20.4.1. Сброс пароля на маршрутизаторе

Первый шаг – перезагрузить маршрутизатор. Этот шаг заключается в отключении питания с помощью кнопки или вытаскивании кабеля питания из розетки и его обратного подключения. Опять же, поскольку вы будете перезагружать и уничтожите конфигурацию запуска, перед запуском вам нужно будет подключиться к маршрутизатору через последовательную консоль.

Практикум

С помощью последовательной консоли, подключенной к маршрутизатору, перезагрузите маршрутизатор. Следите за процессом загрузки.

Вы должны увидеть нормальный вывод загрузки:

```
System Bootstrap, Version 12.3(8r)T8, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2004 by cisco Systems, Inc.
PLD version 0x10
GIO ASIC version 0x127
c1841 processor with 393216 Kbytes of main memory
Main memory is configured to 64 bit mode with parity disabled
```

Вам нужно будет прервать процесс загрузки, отправив сигнал прерывания из вашего эмулятора терминала. Не все эмуляторы терминала могут это сделать, PuTTY из них редкое исключение. См. рис. 20.1, на котором показано, как отправить сигнал прерывания на маршрутизатор.

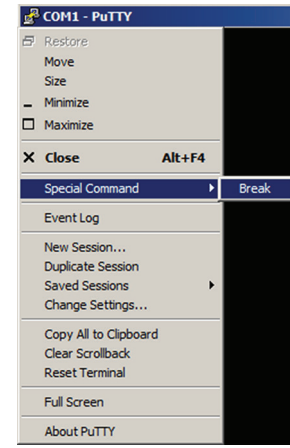

Рис. 20.1 ❖ Отправка сигнала прерывания в программе PuTTY

Практикум

Отправьте сигнал прерывания с вашего эмулятора терминала. В меню PuTTY выберите команду **Special Command ⇨ Break** (Специальные команды ⇨ Break).

Вы должны увидеть что-то, похожее на ошибку и последующее приглашение:

```
*** Address Error (Load/Fetch) Exception ***
Access address = 0x1

PC = 0x1, Cause = 0x10, Status Reg = 0x3041e803
rommon 1 >
```

Если вы видите приглашение `rommon`, то вы успешно прервали процесс загрузки и вошли в режим ROMMON! Следующий шаг – изменить *регистр конфигурации*, чтобы игнорировать конфигурацию запуска. С помощью регистра конфигурации настраивается несколько моментов работы маршрутизатора, включая то, считывает ли он конфигурацию запуска или игнорирует ее.

Практикум

В приглашении `rommon` введите команду `confreg 0x2142` и нажмите клавишу **Enter**. После того как команда будет выполнена, вы увидите сообщение о перезагрузке:

```
rommon 1 > confreg 0x2142
```

Вам нужно перезагрузить маршрутизатор с помощью кнопки или отключить/включить питание, чтобы привести в действие новую конфигурацию. Наберите команду `reset` и нажмите клавишу **Enter**:

```
rommon 2 > reset
```

Маршрутизатор перезагрузится, после чего, если все идет гладко, вы увидите следующее сообщение:

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

На этом этапе вы можете ответить `no` на предложение загрузить диалоговое окно начальной конфигурации, чтобы перейти к обычному приглашению системы IOS, или `yes`, чтобы система IOS провела вас через некоторые базовые настройки конфигурации. Независимо от того, что вы выберете, вы еще не закончили!

Практикум

Войдите в режим глобальной конфигурации и создайте нового пользователя, указав его имя и пароль:

```
Router(config)#username ben privilege 15 secret cisco
```

Измените пароль привилегированного пользователя:

```
Router(config)#enable secret cisco
```

Вы можете внести другие изменения по вашему усмотрению, например задать имя маршрутизатора. Перед тем как закончить, установите значение регистра конфигурации равным `0x2102`:

```
Router(config)#config-register 0x2102
```

Так вы инструктируете ROMMON при загрузке инсталлировать конфигурацию запуска, которую вы сейчас собираетесь сохранить.
В конце сохраните рабочую конфигурацию и перезагрузите маршрутизатор:

```
Write memory
Reload
```

Порядок операций здесь важен. После изменения регистра конфигурации вам нужно сохранить конфигурацию. Если все пойдет хорошо, маршрутизатор загрузит систему IOS с внесенными вами изменениями.

20.4.2. Сброс пароля на коммутаторе

Процесс сброса пароля на коммутаторе почти идентичен такому же процессу для маршрутизатора, за исключением того, *как* вы попадаете в режим ROMMON.

Практикум

Включите питание коммутатора и в процессе загрузки нажмите кнопку выбора режима на передней панели. Коммутатор должен перевести вас в режим ROMMON. Шаги с этого момента такие же, что и для маршрутизатора, поэтому я просто кратко изложу их здесь. В режиме ROMMON измените регистр конфигурации, присвоив ему значение `0x2142`, используя команду `confreg 0x2142`.
Перезагрузите коммутатор.
Когда он загрузится, внесите изменения, присвойте регистру конфигурации обратно значение `0x2102`, используя команду конфигурации `config-register 0x2102` в режиме глобальной конфигурации.
Сохраните текущую конфигурацию, а затем перезагрузите коммутатор, чтобы новая конфигурация заработала!

20.5. Команды, использованные в этой главе

В условиях сетевого бедствия ваше внимание будет сосредоточено на том, чтобы все снова запустилось, поэтому в вашем распоряжении имеется ограниченное количество команд. В табл. 20.1 перечислены эти команды и их описания.

Таблица 20.1. Команды, использованные в этой главе

Функция	Команда	Режим конфигурирования	Описание
Перезагрузка устройства	`Reload in 15`	–	Программирует перезагрузку через 15 минут, начиная с текущего момента
Перезагрузка устройства	`Reload cancel`	–	Отменяет запланированную перезагрузку

Окончание табл. 20.1

Функция	Команда	Режим конфигурирования	Описание
Удаление конфигурации запуска	`Delete nvram:startup-config`	–	Удаляет загрузочную конфигурацию из NVRAM
Сброс пароля	`confreg 0x2142`	ROMMON	Инструктирует ROMMON не использовать конфигурацию запуска
Сброс пароля	`config-register 0x2102`	Глобальный	Инструктирует ROMMON использовать конфигурацию запуска при загрузке

Глава 21

Контрольный список производительности и работоспособности

В этой главе я продемонстрирую вам контрольный список показателей производительности и работоспособности, которые могут пригодиться, когда вы столкнетесь с неясной жалобой на медленно работающую сеть. Даже если жалоба исходит только от одного пользователя, вам не следует сразу бросаться исследовать ее. Этот контрольный список даст вам несколько подсказок относительно того, нужно ли сразу устранять проблему или исследовать ситуацию глубже. Я подчеркиваю, что этот контрольный список не является руководством по устранению неполадок. Он только покажет вам, есть ли проблема – все равно какая – и как ее исправить.

Помимо предоставления контрольного списка, я покажу вам, как проверить каждый элемент. Что-то из этого вы видели раньше, а кое-что будет новым. Имейте в виду, что вам, возможно, придется работать на нескольких устройствах в вашей сети, используя этот контрольный список. Редко, что сетевая проблема будет очевидным образом проявляться на каждом маршрутизаторе и коммутаторе. Кстати, все команды IOS, которые я вам покажу в этой главе, за одним исключением, будут работать как на маршрутизаторах, так и на коммутаторах. Я расскажу про исключение, когда мы доберемся до него.

Итак, контрольный список:

- перегружен ли процессор?
- каково время безотказной работы системы?
- поврежден ли сетевой кабель или разъем?
- отклик на пинг необычно медленный или непоследовательный?
- нестабильны ли маршруты?

21.1. Перегружен ли процессор?

Обычная загрузка центрального процессора (ЦП) зависит от устройства, какую роль оно выполняет в сети и от того, насколько загружена сеть в целом. Команда `show process cpu history` отобразит три графика, показывающих использование ЦП по времени за последние 60 секунд, 60 минут и 72 часа:

```
                          11111
   5555544444444444444455555777774444455555444444444444444555555
100
 90
 80
 70
 60
 50
 40
 30
 20                        *****
 10 *****               **********     *****               ***
   0....5....1....1....2....2....3....3....4....4....5....5....6
            0    5    0    5    0    5    0    5    0    5    0
            CPU% per second (last 60 seconds)

   1111111111111 1111111115111111111111111111 1 1111111111111
   7755553554524745255255273555523555551555155565715555505115 34
100
 90
 80
 70
 60 *
 50 *
 40 *
 30 *
 20 ****** ** *    * ** ** * ****  ***** *** *** *  ***** *  *
 10 ########################################*###*#*############
   0....5....1....1....2....2....3....3....4....4....5....5....6
            0    5    0    5    0    5    0    5    0    5    0
            CPU% per minute (last 60 minutes)
          * = maximum CPU%    # = average CPU%

   564545555354545
   512522004969548
100
 90
 80
 70
 60 ** *      * * *
 50 ** * **** *** *
 40 ***************
 30 ***************
```

```
 20 ***************
 10 ###############
   0....5....1....1....2....2....3....3....4....4....5....5....6....6....7.
   .
            0    5    0    5    0    5    0    5    0    5    0    5    0
                CPU% per hour (last 72 hours)
               * = maximum CPU% # = average CPU%
```

На последних двух графиках символ # указывает на среднее использование ЦП. На Коммутаторе 1 оно составляет 10% или того меньше за последний час. Вторичные колебания использования ЦП являются нормальными, а вы хотите найти высокий средний уровень использования ЦП. Если вы видите устойчивое среднее потребление ЦП выше 80%, это может указывать на проблему.

Обратите внимание, что на 72-часовом графике данные заканчиваются на 15-часовой отметке. Это означает, что коммутатор был отключен примерно 15 часов назад.

21.2. Каково время непрерывной работы системы?

Как правило, маршрутизаторы и коммутаторы всегда должны быть включены. Если что-то перезагружается неожиданно, это почти всегда человеческий фактор или проблема с устройством. Вы можете выяснить, когда устройство перезагрузилось, выполнив команду `show version`:

```
Switch1#show version
Cisco IOS Software, C3560 Software (C3560-IPSERVICESK9-M), Version
     15.0(2)SE5, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Fri 25-Oct-13 13:18 by prod_rel_team

ROM: Bootstrap program is C3560 boot loader
BOOTLDR: C3560 Boot Loader (C3560-HBOOT-M) Version 12.2(44)SE6, RELEASE
     SOFTWARE (fc1)

Switch1 uptime is 14 hours, 43 minutes
System returned to ROM by power-on
System image file is "flash:/c3560-ipservicesk9-mz.150-2.SE5.bin"
```

Коммутатор 1 был перезагружен почти 15 часов назад, что прекрасно согласуется с графиком ЦП. Но строка `System returned to ROM by power-on` вводит в заблуждение. Это не обязательно означает, что кто-то отключил питание на коммутаторе. Система IOS отобразит это сообщение, даже если вы выполните мягкую перезагрузку, запустив команду `reload`.

21.3. Поврежден ли сетевой кабель или разъем?

Если только один пользователь жалуется на сеть, может иметь место проблема физического подключения где-то между коммутатором и пользователем.

Ошибки счетчиков команд `show interfaces counters` – надежный способ определения таких проблем. Единственный недостаток – вы не получите никакой полезной информации о маршрутизаторе:

```
Switch1#show interfaces counters errors

Port      Align-Err   FCS-Err   Xmit-Err   Rcv-Err   UnderSize   OutDiscards
Fa0/1             0         0          0         0           0             0
Fa0/2             0         0          0         0           0             0
Fa0/3             0         0          0         0           0             0
Fa0/4             0         0          0         0           0             0

…

Port      Single-Col  Multi-Col   Late-Col  Excess-Col   Carri-Sen  Runts Giants
Fa0/1             0          0          0           0           0             0
Fa0/2             0          0          0           0           0             0
Fa0/3             0          0          0           0           0             0
Fa0/4             0          0          0           0           0             0
```

Если вы запустите эту команду на одном из ваших коммутаторов, то увидите еще множество строк вывода. Я сократил его. Вам не нужно знать, что означает каждая из этих строк. Вам просто нужно знать, что значения в них в идеале должны быть равны `0`. Если какой-либо элемент не равен нулю, это не обязательно указывает на проблему, но любые ненулевые значения должны оставаться неизменными. Если вы видите, что какое-либо из чисел постоянно увеличивается, налицо проблемы с физическим подключением, такие как поврежденный кабель, неправильное подключение к коммутационной панели или сетевому разъему или даже неправильный тип сетевого кабеля.

21.4. Пинг необычно велик или сбоит?

Долгий пинг, особенно в глобальных линиях связи, может указывать на физическую проблему подключения или сильно загруженное соединение. Время отклика пинга около 100 мс не обязательно указывает на проблему, но требует дальнейшего изучения. Изменяемое время пинга может указывать на то, что устройство, соединение или протокол маршрутизации постоянно то работает, то нет. Следует составить график времени пингов для изучения ситуации.

Мой любимый инструмент для графического отображения времени пинга в реальном времени – Colasoft Ping Tool (colasoft.com). На рис. 21.1 показано, как выглядит серия нормальных, непрерывных пингов.

Визуально график не очень интересен. Он в основном плоский, с некоторыми выбросами здесь и там. Бóльшую часть время отклика пинга составляет 1 миллисекунду (мс), что прекрасно. Имейте в виду, это значение иллюстрирует время пинга в моей тестовой сети, поэтому график выглядит на порядок лучше, чем тот, что вы увидите в реальной сети.

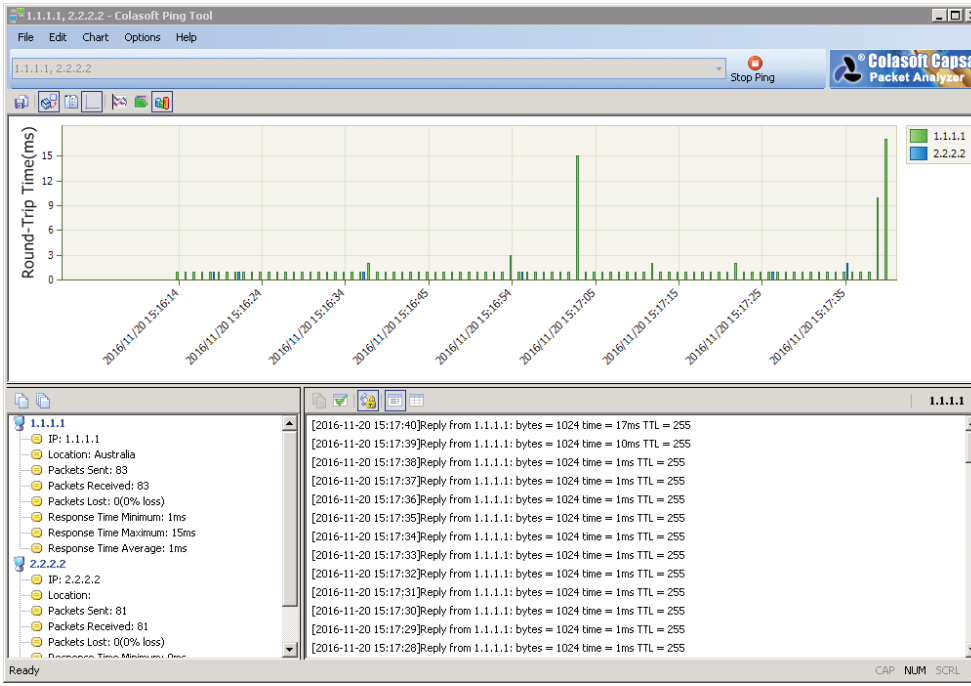

Рис. 21.1 ❖ Программа Colasoft Ping Tool отображает серию пингов

21.5. Нестабильны ли маршруты?

IP-маршрут, пропадающий и появляющийся или нестабильный без объяснений, всегда требует исследования. Самый простой способ проверить эту проблему – выполнить команду `show ip route`:

```
Switch1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback1
      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/2] via 10.0.99.2, 00:10:17, FastEthernet0/24
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
```

```
C        10.0.99.0/30 is directly connected, FastEthernet0/24
L        10.0.99.1/32 is directly connected, FastEthernet0/24
      172.31.0.0/24 is subnetted, 1 subnets
O        172.31.70.0 [110/2] via 10.0.99.2, 00:10:17, FastEthernet0/24
```

Обратите внимание, что два маршрута OSPF появились в таблице маршрутизации всего 10 минут назад. В стабильной сети возраст маршрута будет составлять несколько дней, а не минут. Если у вас есть «неустаревающие» маршруты, вы можете воспользоваться *профилированием таблицы маршрутизации* для отслеживания количества изменений таблицы IP-маршрутизации на устройстве. В режиме глобальной конфигурации за эту функцию отвечает команда `ip route profile`.

Как только вы выполните эту команду, система IOS начнет проверять таблицу маршрутизации каждые пять секунд и записывать количество изменений. А просмотреть количество изменений вы сможете, выполнив команду `show ip route profile`:

```
Switch1#show ip route profile
IP routing table change statistics:
Frequency of changes in a 5 second sampling interval
-------------------------------------------------------------
Change/   Fwd-path  Prefix   Nexthop  Pathcount  Prefix
interval  change    add      change   change     refresh
-------------------------------------------------------------
0         251       251      260      260        260
1         0         0        0        0          0
2         9         9        0        0          0
3         0         0        0        0          0
4         0         0        0        0          0
5         0         0        0        0          0
10        0         0        0        0          0
```

Я должен признать, что мне все еще трудно запомнить, как интерпретировать эту таблицу, поэтому я буду краток. В стабильной сети числа в строке `0` должны увеличиваться каждые пять секунд. `0` указывает на количество изменений в таблице IP-маршрутизации, которые произошли за последние пять секунд. В стабильной сети вы не должны видеть каких-либо изменений. Числа в других строках не должны увеличиваться. Если это не так, маршруты меняются, и вам нужно изучить проблему.

21.6. Команды, использованные в этой главе

Обратитесь к табл. 21.1, чтобы выполнить практическое задание.

Таблица 21.1. Команды, использованные в этой главе

Команда	Режим конфигурирования	Описание
`show processes cpu history`	–	Отображает загрузку ЦПУ в ретроспективе
`show version`	–	Отображает время работы устройства
`show interfaces counters errors`	–	Отображает ошибки на интерфейсах
`show ip route`	–	Отображает таблицу IP-маршрутизации
`ip route profile`	Глобальный	Включает профилирование таблицы IP-маршрутизации
`show ip route profile`	–	Отображает количество изменений в таблице маршрутизации IP

21.7. Практическое задание

Изучая приведенный в этой главе контрольный список, учтите, что конкретные корректные значения могут быть различны в зависимости от сети. В свободное время заполните указанные ниже пункты, чтобы сформировать представление о каждом устройстве и его нормальном поведении в сети. Обязательно сделайте это, когда сеть работает без сбоев и пользователи не жалуются.

Вот некоторые моменты, которые нужно записать:

- средняя загрузка ЦП за последние 24 часа;
- среднее время пинга для сетевых ресурсов (серверов) из разных офисов;
- существуют ли какие-либо порты, на которых всегда имеются ошибки?
- типичный возраст маршрутов;
- существуют ли какие-либо маршрутизаторы или коммутаторы, которые работали без перезагрузки на удивление долгое или короткое время?

Может потребоваться некоторое время, чтобы проанализировать всю эту информацию, поэтому не рассчитывайте сделать это за один присест. Вам, возможно, никогда не понадобится эта информация, но, имея ее в своем распоряжении, в случае сбоя поставить диагноз будет намного легче.

Глава 22

Следующие шаги

Поскольку ваш курс обучения приблизился к концу, у вас должно сформироваться довольно хорошее представление о том, хотите ли вы продолжить развивать навыки работы с сетями Cisco. В этой главе я продемонстрирую вам некоторые учебные ресурсы, чтобы вы могли продолжить обучение.

22.1. Сертификационные ресурсы

Если вы серьезно относитесь к карьере специалиста Cisco, следующим логичным шагом становится получение сертификата Cisco. Cisco предлагает различные сертификации, ориентированные на различные технологии, и наиболее популярной остается тема маршрутизации и коммутации (Routing and Switching). Cisco Certified Entry Networking Technician (CCENT) и Cisco Certified Network Associate (CCNA) – два сертификата начального уровня, но многие менеджеры по персоналу даже не слышали о CCENT. Мой совет: если вы собираетесь пройти сертификацию, начните с CCNA.

Сертификация предоставит вам следующее:

- прочный базис теории сетей;
- практику в расширенной конфигурации и устранении неполадок;
- более высокий оклад.

Сертификация часто бывает решающим фактором между приглашением на интервью и выслушиванием вопросов типа «Знаете ли вы кого-нибудь еще, кто может подойти?». Много лет назад я сидел в офисе с менеджером по техническому персоналу, и ситуация была именно такой. Вряд ли вы найдете много рабочих мест, где вас будут воспринимать всерьез как настоящего администратора сетей Cisco, если у вас нет сертификации.

Сертификация не бесплатна. Стоимость экзамена CCNA составляет около 320 долларов США, и эта сумма не возвращается. Чтобы сдать экзамен, вам понадобится солидная подготовка и много практики. На сайте Pluralsight.com вы найдете курсы для обучения всем уровням сертификации Cisco, начиная от начального и заканчивая экспертным.

22.2. Лаборатория виртуальной интернет-маршрутизации Cisco

В главе 1 я посоветовал не использовать лабораторию виртуальной интернет-маршрутизации Cisco (Virtual Internet Routing Lab – VIRL) для выполнения

практических заданий из этой книги, поскольку она не даст вам такого же опыта, который вы получите при работе с реальным оборудованием. Это может показаться тривиальным, но если вы собираетесь сосредоточиться на карьере администратора сетей Cisco, вам нужно знать, как работать с реальными маршрутизаторами и коммутаторами. Однажды сотрудник рассказал мне историю о необычном опыте работы своего брата. Менеджер, проводящий собеседование, сел перед маршрутизатором Cisco и попросил настроить его. Как ни странно, работа была даже не на замещение вакансии сетевого администратора! Вы можете себе представить, *что* потенциальный работодатель может спросить вас, если вы проходите интервью на позицию сетевого администратора?

Услышав это, вам захочется попрактиковаться с несколькими маршрутизаторами и коммутаторами, чтобы понять, каково это – настраивать и устранять неполадки в крупной сети. Настройка и эксплуатация такой сети с использованием реального оборудования Cisco обойдется довольно дорого. Если вы готовы предпринять решительный шаг и настроить VIRL, ознакомьтесь с курсом Pluralsight компании Brandon Carroll, используя Cisco VIRL для сертификации CCENT и CCNA (www.pluralsight.com/courses/cisco-virl-ccent-ccna-studies).

22.3. Устранение неполадок с позиции конечного пользователя

В этой книге вы узнали, как настраивать маршрутизаторы и коммутаторы Cisco, а также выполнять некоторые функции по устранению неполадок. Но чтобы стать хорошо зарекомендовавшим себя сетевым администратором, вы также захотите узнать, как устранять некоторые неполадки сети с рабочей станции конечного пользователя. Это особенно важно, если у вас нет доступа к устройствам Cisco на работе, а вы хотите развивать навыки сетевого администрирования.

Мой курс Pluralsight, Practical Networking (www.pluralsight.com/courses/practical-networking) даст вам практическое понимание того, как устранять проблемы, связанные с сетью, с позиции рабочей станции конечного пользователя. Если вы объедините эти знания с тем, что узнали в этой книге, то сможете лучше понять, как клиентские компьютеры и серверы взаимодействуют с маршрутизаторами и коммутаторами.

22.4. Никогда не останавливайтесь

Представленные ресурсы должны предоставить достаточно информации для старта, хотите ли вы углубиться в администрирование сетей или просто возиться с маршрутизаторами и переключателями как любитель. На данный момент у вас должен быть достаточный фундамент для выполнения наиболее общих задач администрирования. Как далеко вы зайдете в дальнейшем, зависит только от вас!

Книги издательства «ДМК Пресс» можно заказать
в торгово-издательском холдинге «Планета Альянс» наложенным платежом,
выслав открытку или письмо по почтовому адресу:
115487, г. Москва, 2-й Нагатинский пр-д, д. 6А.
При оформлении заказа следует указать адрес (полностью),
по которому должны быть высланы книги;
фамилию, имя и отчество получателя.
Желательно также указать свой телефон и электронный адрес.
Эти книги вы можете заказать и в интернет-магазине: **www.alians-kniga.ru**.
Оптовые закупки: тел. **(499) 782-38-89**.
Электронный адрес: **books@alians-kniga.ru**.

Бен Пайпер

Администрирование сетей Cisco: освоение за месяц

Главный редактор *Мовчан Д. А.*
dmkpress@gmail.com
Перевод *Райтман М. А.*
Корректор *Синяева Г. И.*
Верстка *Чаннова А. А.*
Дизайн обложки *Мовчан А. Г.*

Формат 70×100 1/16.
Гарнитура «PT Serif». Печать офсетная.
Усл. печ. л. 29,625. Тираж 200 экз.

Веб-сайт издательства: **www.dmkpress.com**