



В этой главе...

- **Команды ping и traceroute.** Этот раздел посвящен командам `ping` и `traceroute`, а также нюансам их использования при поиске и устранении неисправностей в сетях.
- **Устранение неисправностей в процессах пересылки пакетов.** В этом разделе подробно описан процесс перенаправления пакетов, в частности — процессы маршрутизации в оконечных узлах и аппаратных маршрутизаторах. В нем также рассмотрены основные проблемы, возникающие при передаче пакетов в сети.
- **Советы относительно поиска и устранения неисправностей и необходимый инструментарий.** В этом разделе описаны вопросы, связанные с влиянием различных факторов на процесс пересылки пакетов. В нем также содержится множество советов по поиску и устранению неисправностей, а также описаны важные команды.

Поиск и устранение ошибок в IP-маршрутизации

Эта глава преследует две основные цели. Первая — подробно описать полезные инструменты и команды, которые не рассматривались в главах 4–6, в частности инструменты, которые помогут проанализировать проблемы в сети. Вторая цель — кратко описать концепции, которые рассматривались в части II этой книги. Повторение основ маршрутизации понадобится для уверенного и успешного устранения неисправностей в современных компьютерных сетях. В главе также даны разнообразные советы по методам поиска и устранения неисправностей в сетях с использованием технологий, которые рассматривались в этой части книги.

Процесс поиска и устранения неисправностей затрагивает несколько тем, описанных как в этой книге выше, так и в томе I. Из этой главы вы узнаете, как ответить на вопрос сертификационного экзамена CCNA, если в нем рассматривается какая-либо нештатная ситуация в сети. Несмотря на то что в списке тем экзамена поиск и устранение неисправностей в явном виде не указаны, эту главу следует прочитать. В ней нет контрольных вопросов, поскольку ее тема не входит в экзамен, тем не менее в лабораторном задании на экзамене CCNA описанные ниже инструменты и команды, а также приведенные полезные советы могут пригодиться. Если читатель внимательно изучил главы по устранению неисправностей в томе I книги и считает, что его знания по этой теме находятся на достаточно высоком уровне, он может сразу перейти к последнему разделу главы.

Введение

В этой главе подробно описан процесс поиска и устранения неисправностей в IP-маршрутизации. В первом разделе подробно описаны две наиболее часто используемые в таком процессе команды — **ping** и **tracert**. Во втором — рассмотрен процесс маршрутизации с точки зрения поиска и устранения неисправностей в сетях, в частности описано, как изолировать проблемы, связанные именно с маршрутизацией, и найти основную причину неправильной работы сети. В последнем, третьем, разделе главы представлены небольшие темы, которые пригодятся при поиске и устранении неисправностей в сетях.

ВНИМАНИЕ!

В текущей главе, как и в главе 15 тома I описан процесс поиска и устранения неисправностей в IP-маршрутизации. Эта тема важна для понимания принципов маршрутизации в сетях и пригодится при сдаче сертификационных экзаменов ICND1, ICND2 и CCNA. Поскольку темы этих трех экзаменов перекрываются, частично перекрываются и главы в двух томах книги. Тем не менее следует помнить, что в этой главе часть вопросов рассмотрена намного подробнее, чем в соответствующей главе тома I.

Команды ping и traceroute

В этом разделе описан рекомендуемый специалистами алгоритм поиска и устранения неисправностей в IP-маршрутизации, т.е. процесс анализа плоскости передачи данных (data plane) в маршрутизаторах компании Cisco. Прежде всего рассмотрены базовые средства анализа компьютерной сети: протокол ICMP, команды **ping** и **tracert**.

Протокол управляющих сообщений сети Интернет

В стек TCP/IP входит протокол ICMP (Internet Control Message Protocol — протокол управляющих сообщений сети Интернет), который предназначен для управления и контроля TCP/IP-сети. С помощью протокола ICMP можно получить много полезной информации о состоянии сети и ее работоспособности. В данном случае первая часть названия, а именно *протокол управляющих сообщений*, является самой информативной. Протокол ICMP используется прежде всего для управления IP-сетями посредством специализированных сообщений и процедур уровня IP, поэтому его относят к сетевому уровню стека TCP/IP. В действительности ICMP-сообщения содержатся внутри IP-пакета и транспортный уровень не используется, поэтому говорят, что это протокол третьего уровня эталонной модели взаимодействия открытых систем.

Протокол ICMP описан в RFC 792, в котором можно найти следующее определение.

- Иногда шлюзу (маршрутизатору) или узлу-получателю необходимо обмениваться сообщениями с узлом-отправителем, например, чтобы уведомить об ошибке при обработке полученной дейтаграммы. Для этой цели используется спе-

циализированный протокол, называемый *протоколом управляющих сообщений сети Интернет (ICMP)*, который в качестве своей основы использует протокол IP, хотя в действительности является его составной частью и должен присутствовать в любой реализации последнего.

В протоколе ICMP используется несколько разных типов сообщений для разных управляющих функций (табл. 7.1).



Таблица 7.1. Типы сообщений протокола ICMP

Сообщение	Описание
Получатель недоступен (Destination Unreachable)	Уведомляет узел-отправитель о том, что невозможно доставить пакет
Превышен интервал (Time Exceeded)	Превышен временной интервал для доставки пакета, пакет был уничтожен
Перенаправление (Redirect)	Маршрутизатор, приславший такое уведомление, получил сообщение от другого маршрутизатора о том, что у последнего есть более оптимальный маршрут. Уведомление указывает узлу-отправителю, что следует использовать оптимальный маршрут
Эхо-запрос, эхо-ответ (Echo Request, Echo Reply)	Эти сообщения используются в команде ping для проверки связи в сети

Команда ping и эхо-запросы и эхо-ответы протокола ICMP

Команда **ping** использует сообщения эхо-запроса (Echo Request) и эхо-ответа (Echo Reply) протокола ICMP. Зачастую сетевые специалисты говорят “отправь ping-пакет”, а в действительности подразумевается, что будет отправлен эхо-запрос соответствующей командой. Смысл обоих сообщений вполне очевиден из их названий. Эхо-запрос подразумевает, что получивший его узел должен отправить некоторый ответ на такой запрос, а эхо-ответ — это специализированный ответ протокола ICMP на запрос. В ответ включается определенная информация, задаваемая ключами или параметрами команды **ping**: в зависимости от того, какие флаги были получены в запросе, разные блоки информации будут включены в ответное сообщение.

С помощью команды **ping** можно использовать различные комбинации запросов и ответов и получать разную информацию. Например, в параметрах команды можно указать длину пакета, адрес отправителя и адрес получателя, а также установить определенные параметры в IP-заголовке. В главе 4, “Маршрутизация IP: статические и подключенные маршруты”, был показан пример расширенного варианта команды **ping**, в котором устанавливались различные дополнительные параметры.

Сообщение о недоступности получателя протокола ICMP

В этой книге основное внимание уделяется протоколу IP как протоколу, на котором основаны маршрутизация и адресация сети. Тем не менее основная задача протоколов стека TCP/IP гораздо шире — доставка пакетов от приложения-отправителя приложению-получателю. Узлы и маршрутизаторы пересылают уведомление о недоступности получателя (Destination Unreachable) протокола ICMP в том случае, когда они не могут доставить данные приложению или узлу-получателю.

В сообщении о недостижимости узла протокола ICMP могут содержаться пять независимых кодов, позволяющих точно указать причину, по которой пакет не может быть доставлен. Каждый из пяти кодов описывает определенную функцию протокола IP, TCP или UDP.

В качестве примера использования различных кодов рассмотрим сеть, показанную на рис. 7.1. Предположим, что компьютер Фреда пытается соединиться с веб-сервером. (Веб-сервер использует протокол HTTP, который, в свою очередь, в качестве транспортного механизма использует протокол TCP.) Три ICMP-сообщения о недостижимости узла-получателя из пяти могут быть использованы маршрутизаторами А и Б. Два оставшихся сообщения могут быть сгенерированы веб-сервером. Такие сообщения могут быть пересланы компьютеру Фреда в ответ на его запрос к веб-серверу.

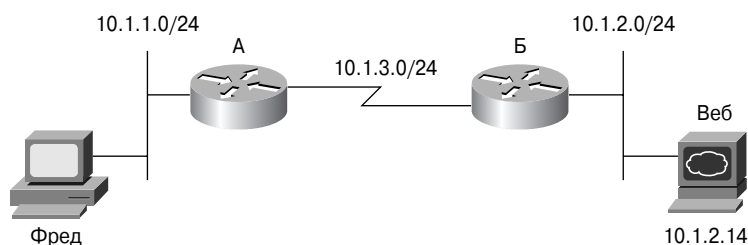


Рис. 7.1. Пример сети для обсуждения ICMP-сообщений

В табл. 7.2 описаны упоминавшиеся выше сообщения о недостижимости узла протокола ICMP, а ниже описано использование таких сообщений для примера сети, представленной на рис. 7.1.

Таблица 7.2. Коды сообщений о недостижимости узла протокола ICMP

Код	Пример использования	Отправитель
Сеть недостижима (Network unreachable)	В таблице маршрутизации нет записи для сети узла-получателя	Маршрутизатор
Узел недостижим (Host unreachable)	Пакет может быть успешно маршрутизирован в сеть-получатель, но узел-получатель в такой сети недоступен	Маршрутизатор
Невозможно фрагментировать пакет (Can't fragment)	В пакете установлен бит, указывающий, что его фрагментация запрещена (Don't Fragment bit), но маршрутизатору необходимо разбить пакет на блоки меньшего размера	Маршрутизатор
Протокол недостижим (Protocol unreachable)	Пакет был успешно доставлен узлу-получателю, но соответствующий протокол транспортного уровня в узле не работает	Узел
Порт недостижим (Port unreachable)	Пакет был успешно доставлен узлу-получателю, но запрашиваемый порт получателя или приложение недоступно	Узел

Ниже описаны коды, перечисленные в табл. 7.2, на примере схемы, показанной на рис. 7.1.

- **Сеть недостижима.** Маршрутизатор А будет устанавливать этот код в сообщениях, если у него нет маршрута к сети-получателю. В рассматриваемом примере сети маршрутизатору нужно отправить пакет в сеть 10.1.2.0/24. Если в таблице маршрутизации нет записи для такой сети, маршрутизатор А отправит компьютеру Фреда ICMP-сообщение о недостижимости получателя с кодом "сеть недостижима" в ответ на пакет от компьютера Фреда, предназначенный узлу с адресом 10.1.2.14.
- **Узел недостижим.** Этот код в сообщении сигнализирует о том, что недоступен один узел из всей сети или подсети. Например, если у маршрутизатора А есть маршрут к сети 10.1.2.0/24, пакет будет переслан маршрутизатору Б. Если соответствующий интерфейс локальной сети маршрутизатора Б находится в рабочем состоянии, то у него также есть маршрут к сети 10.1.2.0/24 (она напрямую подключена к устройству), следовательно, маршрутизатор посредством протокола ARP попытается получить MAC-адрес веб-сервера. Если же веб-сервер не работает (например, выключен), ответ по протоколу ARP не будет получен и маршрутизатор Б перешлет компьютеру Фреда ICMP-сообщение о недостижимости узла, свидетельствующее о том, что маршрут к сети есть, но пакет не может быть переслан узлу с IP-адресом 10.1.2.14.
- **Невозможно фрагментировать пакет.** Это третий код, который может быть установлен маршрутизаторами в ICMP-сообщении. Фрагментация используется, если маршрутизатор пересылает пакет через канал, максимально допустимый размер блока данных в котором меньше, чем во входящем интерфейсе. Маршрутизатору в таком случае нужно разбить пакет на более мелкие блоки, но в заголовке пакета может быть установлен флаг, запрещающий фрагментацию (Do Not Fragment). В таком случае, если маршрутизатору А или Б нужно разбить пакет, но флаг установлен, устройство отправит ICMP-сообщение компьютеру Фреда о том, оно не может фрагментировать пакет.
- **Протокол недостижим.** Если пакет был успешно доставлен веб-серверу, но есть какая-либо проблема в самом сервере, то могут быть указаны еще два кода в ICMP-ответах. Первый код свидетельствует о том, что протоколы более высокого уровня (относительно IP), обычно TCP или UDP, не запущены на сервере. Такая ситуация маловероятна, поскольку в большинстве операционных систем используется стек TCP/IP, а три основных протокола этого стека, IP, TCP и UDP, реализованы в программном обеспечении. Если же веб-сервер получит IP-пакет, а службы протоколов TCP и UDP по какой-либо причине не работают, то компьютеру Фреда будет отправлено сообщение с кодом "протокол недостижим" в ответ на пакет с адресом получателя 10.1.2.14.
- **Порт недостижим.** Последний из рассматриваемых кодов ошибок встречается в сетях намного чаще, чем предыдущий. Если сервер включен и исправно функционирует (т.е. компьютер работает), а программное обеспечение веб-сервера на нем не запущено, пакет будет успешно доставлен серверу, но не сможет быть обработан веб-службой. Фактически такой сервер не прослушивает стандартный порт веб-службы, поэтому узел с адресом 10.1.2.14 отправит компьютеру Фреда сообщение с кодом "порт недостижим" в ответ на его пакет.

ВНИМАНИЕ!

Согласно политикам безопасности современных сетей перечисленные выше сообщения обычно отфильтровываются, чтобы повысить их уровень защиты.

Команда **ping** выдает различные коды ответов, соответствующие разным ошибкам. В табл. 7.3 перечислены коды ответов этой команды в операционной системе Cisco IOS.

Таблица 7.3. Коды ответов команды ping

Код	Описание
!	Получен обычный эхо-ответ
.	Эхо-ответ от запрашиваемого узла не был получен
U	Получено сообщение о недостижимости узла-получателя
N	Получено сообщение о недостижимости сети или подсети
M	Получено сообщение о невозможности фрагментации
?	Получен неизвестный пакет

ICMP-сообщение о перенаправлении пакетов

ICMP-сообщение о перенаправлении пакетов представляет собой средство информирования маршрутизаторами пользовательских узлов, позволяющее указать последним, что для определенных адресов получателей необходимо изменить адрес стандартного шлюза. В большинстве узлов используется такой параметр, как IP-адрес стандартного маршрутизатора подсети, которому пересылаются пакеты, получатель которых находится не в локальной подсети. Тем не менее в сети может быть несколько граничных маршрутизаторов и стандартный шлюз узла может быть не самым оптимальным транзитным переходом к какой-либо сети-получателю. Стандартный шлюз может обнаружить такую ситуацию и отправить соответствующее ICMP-сообщение узлу, чтобы указать, что пакеты к определенным получателям лучше всего пересылать через другой маршрутизатор.

Например, в сети, показанной на рис. 7.2, ПК использует маршрутизатор Б в качестве своего стандартного маршрутизатора. Тем не менее маршрут к сети 10.1.4.0 через маршрутизатор А оптимальнее (предполагается, что для всех подсетей на рис. 7.2 используется маска 255.255.255.0). Компьютер (ПК) пересылает пакет маршрутизатору Б (этап 1 на рис. 7.2). Маршрутизатор Б на основании записи в своей таблице маршрутизации пересылает пакет дальше (этап 2), т.е. маршрутизатору А, через который пролегает оптимальный маршрут. Обнаружив такую ситуацию, маршрутизатор Б посылает ICMP-сообщение о перенаправлении пакетов для ПК (этап 3), чтобы указать, что все последующие пакеты в сеть 10.1.4.0 следует направлять через маршрутизатор А. Вполне вероятно, что ПК проигнорирует такое сообщение и продолжит слать все пакеты через маршрутизатор Б, но в данном примере компьютер правильно обрабатывает такое сообщение и уже следующий пакет к тому же самому получателю отправляет через маршрутизатор А (этап 4).

ICMP-сообщение о превышении временного интервала на пересылку пакета

Это сообщение используется для уведомления узла о том, что пакет был отброшен из-за истечения интервала на его передачу. В действительности время как параметр при передаче данных не используется, но, чтобы предотвратить возникновение кольцевых маршрутов в сети, которые могут существовать сколь угодно долго, в пакетах используется специальный счетчик транзитных узлов. Такой счетчик называют *временем существования пакетов* (Time to Live — TTL). Маршрутизаторы при передаче пакета уменьшают значение TTL на 1; если значение счетчика достигло 0, пакет отбрасывается устройством. Данный механизм позволяет избежать заикливания пакетов при передаче данных. На рис. 7.3 проиллюстрирован описанный процесс.

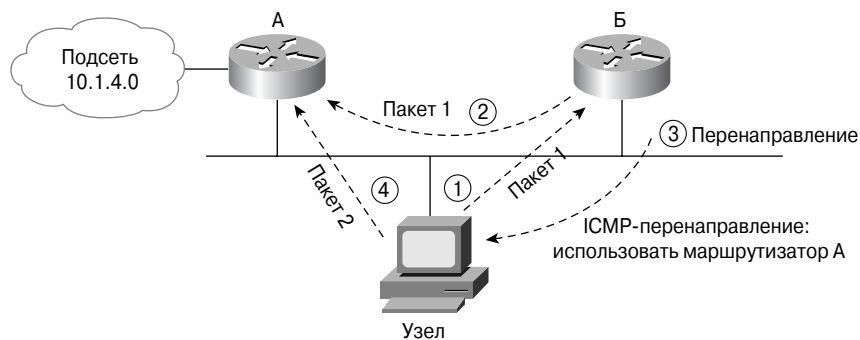


Рис. 7.2. ICMP-сообщение о перенаправлении пакетов

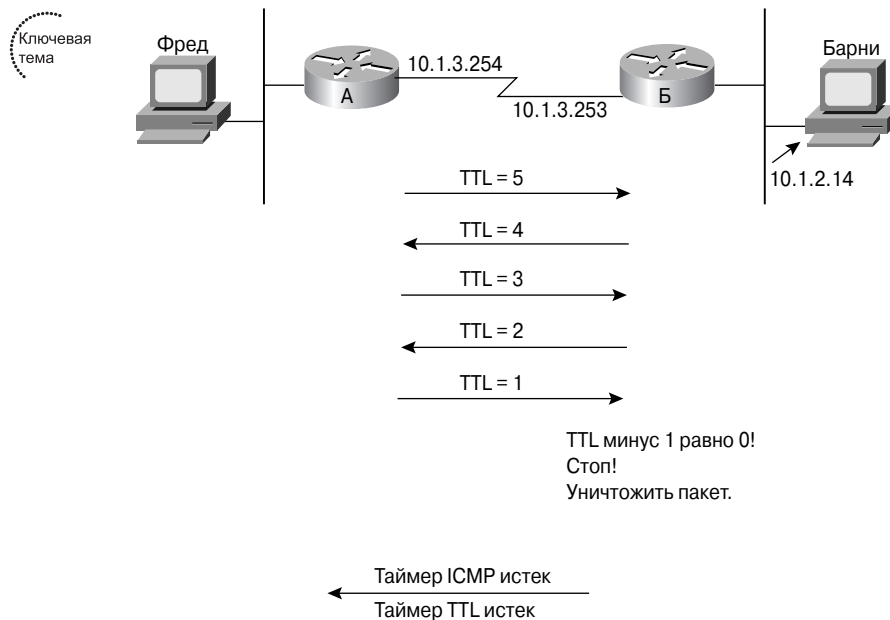


Рис. 7.3. Счетчик TTL уменьшается до нуля

Как показано на рис. 7.3, маршрутизатор уничтожает пакет и пересылает ICMP-сообщение об истечении интервала времени на пересылку пакета с соответствующим кодом узлу-отправителю. Благодаря такому уведомлению отправитель знает о том, что пакет не был доставлен получателю. Сообщения об истечении интервала времени пересылки пакета могут быть полезны при поиске и устранении неисправностей в сетях. Если таких сообщений много, это может свидетельствовать о проблемах с маршрутизацией в сети.

Команда `traceroute`

Команда `ping` является мощным инструментом для поиска и устранения неисправностей в сетях, тем не менее она может дать ответ только на вопрос “Работает ли маршрут из одной точки в другую?” Команда `traceroute` с точки зрения процесса устранения проблем в сети намного лучше, поскольку она не только позволяет убедиться в том, что маршрут работает, но и показывает IP-адреса всех транзитных маршрутизаторов на маршруте. Фактически эта команда отображает полный маршрут, по которому проследует пакет от отправителя к получателю.

В качестве основного механизма в операционной системе Cisco IOS используется счетчик TTL (т.е. соответствующее поле в заголовке IP-пакета), посредством которого определяется последовательность транзитных устройств на маршруте. При запуске команды `traceroute` в заголовке пакета устанавливается значение TTL, равное 1, для второго пакета запроса значение увеличивается на 1 и т.д. Когда значение счетчика TTL в запросе становится равным 0 (а каждый промежуточный маршрутизатор уменьшает значение счетчика на 1), промежуточное устройство генерирует ICMP-сообщение о превышении интервала на передачу пакета. IP-адрес отправителя такого сообщения и будет идентификатором транзитного устройства, т.е. маршрутизатора, который отбросил пакет.

Чтобы разобраться в механизме работы команды `traceroute`, рассмотрим небольшой пример. Предположим, инженер выполнил команду `traceroute` в интерфейсе командной строки маршрутизатора. Команда пересылает первый набор IP-пакетов (три пакета) с использованием протокола UDP в качестве механизма транспортного уровня, значение счетчика TTL которых равно 1. Когда пакеты достигают ближайшего промежуточного маршрутизатора, он уменьшает значение TTL на 1 и значение счетчика становится равным 0. Такие пакеты отбрасываются промежуточным устройством и отправителю пересылаются сообщения об истечении времени на передачу пакетов. Узел-отправитель определяет IP-адрес отправителя ICMP-сообщений, и он выводится в выводе команды `traceroute`.

На втором этапе команда `traceroute` пересылает следующий набор пакетов, счетчик TTL которых равен 2. Первый промежуточный маршрутизатор уменьшает значение счетчика на 1, и теперь TTL равно 2. Далее, следующий промежуточный маршрутизатор (второй на маршруте), в свою очередь, уменьшает значение счетчика на 1 и TTL снова принимает значение, равное 0. Второй на маршруте маршрутизатор уничтожает пакеты и генерирует сообщение об истечении времени на передачу пакета для отправителя пакетов, таким образом, отправитель узнает адрес второго маршрутизатора на маршруте.

Команда `traceroute` отслеживает прибытие пакетов к окончательному узлу-получателю (т.е. финальной точке маршрута) с помощью ICMP-сообщения о недости-

жимости порта. В пакете, отправляемом операционной системой Cisco IOS, используется номер порта получателя протокола UDP, который заведомо не должен использоваться принимающим узлом. Причем номер порта выбирается достаточно большим, чтобы быть уверенным в том, что никакие приложения его не используют. Как только в ответ будет получено сообщение о недостижимости порта, команда **traceroute** прекратит свою работу и трассировка маршрута завершится.

На рис. 7.4 проиллюстрирован механизм работы команды. Обратите внимание, что для экономии места показан только один пакет из трех для заданного значения TTL. В маршрутизаторе А выполняется команда **traceroute**, чтобы определить маршрут к компьютеру Барни. В примере 7.1 показан вывод команды с включенными отладочными сообщениями (debug messages) в маршрутизаторе Б.

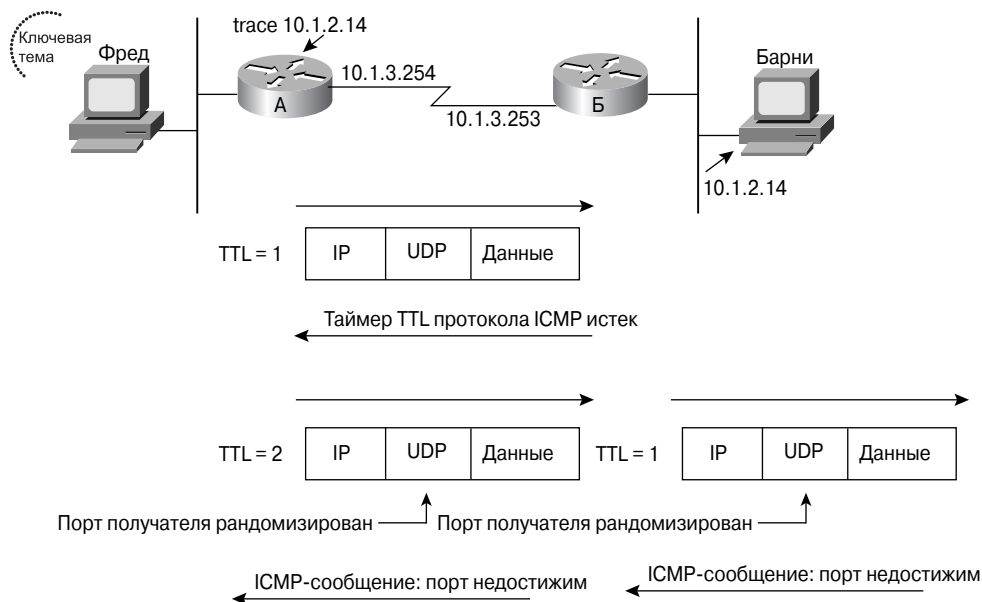


Рис. 7.4. Сообщения, генерируемые командой **traceroute**

Пример 7.1. Отладочные сообщения протокола ICMP в маршрутизаторе Б при использовании команды **traceroute** в маршрутизаторе А

```
RouterA#traceroute 10.1.2.14
Type escape sequence to abort.
Tracing the route to 10.1.2.14
 1 10.1.3.253 8 msec 4 msec 4 msec
 2 10.1.2.14 12 msec 8 msec 4 msec
! Здесь мы переходим к маршрутизатору Б
! Выводимые ниже сообщения появляются после выполнения команды
traceroute
! в маршрутизаторе А
RouterB#debug ip icmp
RouterB#
ICMP: time exceeded (time to live) sent to 10.1.3.254 (dest was 10.1.2.14)
ICMP: time exceeded (time to live) sent to 10.1.3.254 (dest was 10.1.2.14)
ICMP: time exceeded (time to live) sent to 10.1.3.254 (dest was 10.1.2.14)
```

В первой строке вывода команды **tracert** указан IP-адрес маршрутизатора Б, а во второй — IP-адрес узла-получателя. IP-адрес устройства в выводе команды показан слева. Обратите внимание, что это адрес ближайшего к отправителю интерфейса маршрутизатора, поскольку при формировании ICMP-сообщения об истечении интервала на передачу пакета всегда используется ближайший к отправителю порт устройства. Если бы такой адрес в службе DNS был связан с каким-либо именем узла, то оно отображалось бы вместо IP-адреса.

Аналогично расширенному варианту команды **ping**, описанному в главе 4, существует расширенный вариант команды **tracert**, в котором можно выполнять дополнительные манипуляции пересылаемыми пакетами. Как видно из примера 7.1 команда **tracert** в качестве IP-адреса отправителя использует 10.1.3.254, т.е. адрес исходящего интерфейса маршрутизатора А, поэтому в рассматриваемом примере команда **tracert** проверяет как маршрут к узлу 10.1.2.14, так и обратный маршрут к узлу 10.1.3.254. С помощью расширенного варианта команды можно, например, проверить не только такой маршрут, но и маршрут между двумя локальными сетями — сетями маршрутизаторов А и Б. В примере 7.2 проиллюстрирован расширенный вариант этой команды.

ВНИМАНИЕ!

В операционных системах компании Microsoft есть аналогичная команда, только называется она по-другому — **tracert**. В этой команде есть и другие отличия: она пересылает эхо-запросы протокола ICMP и не использует UDP-пакеты. Следовательно, при создании списков ACL может возникнуть такая ситуация, что команда операционной системы Cisco IOS **tracert** работает, а команда **tracert** в операционных системах компании Microsoft — нет. Или наоборот.

Устранение неисправностей в процессах пересылки пакетов

Поиск и устранение неисправностей в IP-маршрутизации — это одна из самых сложных задач, с которыми приходится сталкиваться сетевому специалисту. Как обычно, структурированный подход к процессу значительно упрощает поиск и устранение проблем. В главах 4–6 уже подробно объяснялись теоретические основы маршрутизации и рассказывалось, что должно происходить в сети в режиме нормальной работы. В этом разделе подробно описан следующий этап процесса восстановления работоспособности сети: изолирование проблем. (Формализованный алгоритм поиска и устранения неисправностей подробно описан в главе 3, "Поиск и устранение неисправностей коммутации в локальной сети".)

ВНИМАНИЕ!

Устранение неисправностей в протоколах маршрутизации подробно описано в главе 11, "Устранение неисправностей в протоколах маршрутизации".

Изолирование проблем маршрутизации в узлах

В алгоритме поиска и устранения неисправностей, который описан в этом разделе, есть два основных этапа: этап анализа проблем в сетевых узлах и этап анализа проблем в маршрутизаторах. Вполне очевидно, что, когда два компьютера в сети не могут установить между собой связь, прежде всего выполняется проверка сетевых настроек самих узлов, а именно — может ли каждый из них связаться со своим стандартным шлюзом (default gateway) и переслать ему пакеты. Второй этап процесса поиска и устранения неисправностей связан уже непосредственно с маршрутизаторами и протоколами маршрутизации.

Ниже перечислены этапы процесса поиска и устранения неисправностей на участке между окончательным узлом и ближайшим к нему маршрутизатором.

- Этап 1** Проверьте, отправляет ли узел пакеты получателям вне своей подсети. Укажите в команде `ping` в качестве адреса получателя стандартный шлюз и запустите эту команду со стандартного шлюза, указав в качестве получателя IP-адрес узла. Если команда `ping` выдает отрицательный результат, выполните следующие действия.
- а) Проверьте состояние интерфейса стандартного шлюза. Коды состояния интерфейса должны быть “up and up” (рабочее состояние).
 - б) Проверьте настройки IP-адреса узла и маски, сравните их с настройками стандартного шлюза. Убедитесь, что оба адреса находятся в одной подсети и маски обоих устройств одинаковы. Для последней проверки нужно рассчитать диапазон адресов подсети.
 - в) Если в маршрутизаторе используется магистральный канал сетей VLAN (VLAN trunking), проверьте его работоспособность и убедитесь, что сеть VLAN узла правильно сконфигурирована.
 - г) Если проблема до сих пор существует, проверьте сеть на уровне 1/2: правильность кабеля или, как было рассказано в главе 3, проблемы с настройками сетей VLAN
- Этап 2** Проверьте настройки стандартного шлюза, выполнив команду `ping` для получателя вне локальной подсети. Можно также использовать расширенный вариант команды и в качестве адреса-отправителя пакетов подставить любой из интерфейсов маршрутизатора



На рис. 7.5 показан пример, в котором компьютер ПК1 не может соединиться с веб-сервером ПК4. Чтобы проверить, может ли ПК1 пересылать пакеты в локальную сеть, в нем запущена команда `ping 10.1.1.1`, проверяющая наличие связи со стандартным шлюзом. Аналогично сетевой инженер согласно рекомендациям этапа 1 алгоритма по поиску и устранению неисправностей может выполнить в маршрутизаторе команду `ping 10.1.1.10`. Можно использовать или первый, или второй вариант проверки, поскольку они эквивалентны. Если команда выдает отрицательный результат, тогда следует выполнить пп. а–в этапа 1.

На этапе 2 описанного выше алгоритма устранения неисправностей проверяется еще одна настройка устройства, которой обычно уделяют мало внимания. В действительности команда `ping` на этапе 1 алгоритма не требует, чтобы у узла был настроен (или правильно настроен) стандартный шлюз, поскольку адреса отправителя

и получателя находятся в одной подсети. Если инженер выполняет команду **ping** для стандартного шлюза, а не удаленного узла, то он фактически не проверяет работоспособность протоколов маршрутизации, а только тестирует наличие связи в локальной сети. Например, если в сети, представленной на рис. 7.5, выполнить команду **ping 10.1.13.1** в компьютере ПК1, то будет проверяться работоспособность стандартного шлюза, поскольку адрес 10.1.13.1 находится в другой подсети (не в подсети узла — 10.1.1.0/24). Тем не менее, поскольку адрес просто установлен на другом интерфейсе того же самого шлюза, работоспособность маршрутизации в остальной сети не тестируется.

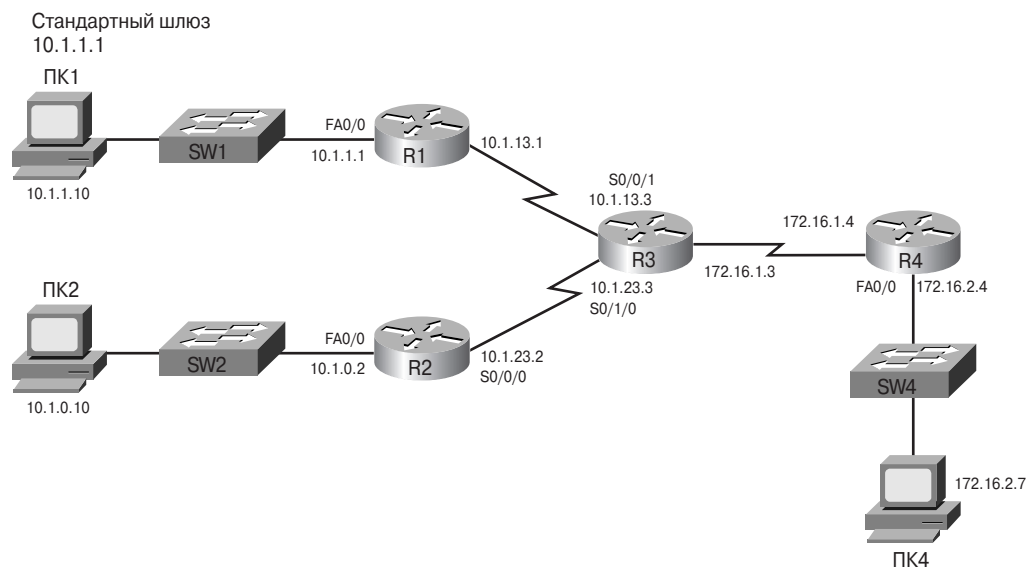



Рис. 7.5. Схема сети для примера алгоритма поиска и устранения неисправностей

Изолирование проблем маршрутизации в маршрутизаторах

Если процесс изолирования проблем в узле завершен и команда **ping** выдает положительный результат для стандартного шлюза, но связи между двумя удаленными узлами нет, необходимо устранить неисправности в маршрутизации. Ниже описан алгоритм поиска и устранения неисправностей в маршрутизации с помощью команды **tracert** в маршрутизаторе. (Следует помнить, что в операционных системах компании Microsoft используется команда **tracert**).

ВНИМАНИЕ!

Приведенный ниже алгоритм полезен в качестве справочника, но он достаточно длинный. Не следует запоминать его во всех деталях; нужно прочитать алгоритм и проанализировать следующие за ним примеры, которые помогут разобраться в этапах алгоритма. Этот формализованный алгоритм предназначен для облегчения процесса поиска и устранения неисправностей в сетях. Обратите внимание, что указанные ниже этапы являются продолжением алгоритма поиска ошибок в сети, рассмотренного выше.

- Этап 3** Проверьте наличие связи с узлом-получателем с помощью расширенного варианта команды `tracroute` в стандартном шлюзе, подставив в качестве адреса отправителя IP-адрес интерфейса из сегмента локальной сети оконечного узла. 
- а) Если команда дает положительный результат, то проблем с прямым или обратным маршрутом нет.
 - б) Если пользовательский трафик все равно не передается между узлами (а команда дала положительный результат!), следует проверить списки контроля доступа (ACL) на всех маршрутизаторах на маршруте, на всех интерфейсах в обоих направлениях
- Этап 4** Если команда `tracroute` на этапе 3 не дала положительного результата, проверьте *маршрут в исходящем направлении* следующим образом.
- а) Установите telnet-сеанс с последним указанным в выводе команды `tracroute` маршрутизатором (т.е. вышестоящим).
 - б) Определите, какой маршрут в маршрутизаторе с каким интерфейсом связан на пути к IP-адресу получателя пакета (с помощью команд `show ip route`, `show ip route ip-адрес`).
 - в) Если искомый маршрут отсутствует, попытайтесь установить причину этого; обычно ошибка связана с тем, что есть неправильный статический маршрут или неправильно сконфигурирован протокол динамической маршрутизации.
 - г) Если маршрут в таблице есть, и это стандартный маршрут (default route), убедитесь, что он будет использоваться устройством (т.е. проверьте команды `ip classless/no ip classless`).
 - д) Если искомый маршрут в таблице есть, выполните команду `ping` с адресом следующего транзитного маршрутизатора на маршруте в качестве параметра. Если же маршрут представляет собой напрямую подключенную сеть, выполните эту же команду с адресом узла-получателя в качестве параметра.
 - Если команда `ping` выдает отрицательный результат, поищите неисправности в соединении между маршрутизатором и узлом на уровне 2, а также проверьте конфигурацию списков доступа (ACL), если они имеются.
 - Если команда `ping` выдает положительный результат, проверьте конфигурацию списков доступа (ACL).
 - е) Если искомый маршрут есть в таблице и каких-либо других ошибок нет, проверьте, правильный ли интерфейс указывает такой маршрут.
- Этап 5** Если на этапе 4 не были выявлены проблемы и ошибки, проверьте *правильность маршрута в обратном направлении*.
- а) Если маршрут в исходящем направлении указывает на какой-либо маршрутизатор и трасса на этом маршрутизаторе прерывается, следует подключиться к этому устройству и выполнить действия, описанные

в этапе 3, но для нижестоящего маршрутизатора. Проанализируйте маршрут в обратном направлении — к узлу-отправителю пакета.

б) Если исходящий маршрут обрывается на интерфейсе, к которому напрямую подключена сеть получателя пакетов, проверьте IP-настройки узла-получателя. Прежде всего, следует убедиться в правильности указанных IP-адреса, маски и стандартного шлюза.

Представим, что ПК1 не может связаться с ПК4 в схеме на рис. 7.5, но у обоих узлов есть связь со своими стандартными шлюзами. На этапе 3 описанного выше алгоритма устранения неисправностей в маршрутизаторах нужно запустить команду **tracert 172.16.2.7** с использованием IP-адреса интерфейса Fa0/0 маршрутизатора R1 в качестве адреса отправителя. Если эта команда выдает IP-адрес 10.1.13.3 в последней строке вывода, то следует перейти к этапу 4 и проверить маршрут в исходящем направлении в маршрутизаторе R3 для получателя с адресом 172.16.2.7. Если на этапе 4 не удастся изолировать проблему, нужно перейти к этапу 5 и переместиться на следующий транзитный маршрутизатор, R4 в данном случае, и проверить маршрут в обратном направлении — от этого маршрутизатора к узлу с адресом 10.1.1.1.

Ниже описаны два сценария событий в сети, в которых используется описанный выше алгоритм поиска и устранения неисправностей в маршрутизации.

Сценарий 1. Проблемы с маршрутом в исходящем направлении

В этом примере используется сеть, показанная на рис. 7.5. В данном случае ПК1 не может посредством веб-браузера подключиться к веб-серверу, запущенному на ПК4. Более глубокое исследование проблемы позволило определить, что ПК1 вообще не может связаться с узлом 172.16.2.7 (ПК4). В примере 7.2 показаны команды, использовавшиеся в маршрутизаторах R1 и R4 для изоляции проблем на этапах 1 и 2, а также частично и на этапе 3.

Пример 7.2. Отладочные сообщения протокола ICMP в маршрутизаторе Б при использовании команды **tracert** в маршрутизаторе А

```
R1#ping 10.1.1.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.10, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R1#ping
Protocol [ip]:
Target IP address: 10.1.1.10
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.13.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose [none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
```

```

Sending 5, 100-byte ICMP Echos to 10.1.1.10, timeout is 2 seconds:
Packet sent with a source address of 10.1.13.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R1#
! Теперь повторим тот же тест на маршрутизаторе R4
R4#ping 172.16.2.7
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.7, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R4#show ip interface brief
Interface      IP-Address      OK? Method Status Protocol
FastEthernet0/0 172.16.2.4     YES manual administratively down down
FastEthernet0/1 172.16.1.4     YES manual up up
Serial0/0/0     unassigned     YES unset administratively down down
Serial0/0/1     unassigned     YES unset administratively down down
Serial0/1/0     unassigned     YES unset administratively down down

```

Стандартный и расширенный варианты команды **ping** в маршрутизаторе R1 позволяют выполнить этапы 1 и 2 алгоритма поиска и устранения неисправностей, а именно — проверить, правильно ли работает маршрутизация в компьютере ПК1. Тем не менее, как видно из примера, маршрутизатор R4 не может установить связь с компьютером ПК4, поскольку его соответствующий интерфейс для локальной сети выключен. Приведенный пример достаточно прост, однако с его помощью можно продемонстрировать общий подход к устранению неисправностей в маршрутизации.

Усложним немного задачу в рамках того же самого сценария развития событий в сети. Предположим, что причина отказа та же самая, но теперь у сетевого инженера нет доступа к маршрутизатору R4. В таком случае можно выполнить только этапы 1 и 2 и убедиться, что сеть в ПК1 исправно работает, но нельзя изолировать проблему для ПК4. В примере 7.3 проиллюстрированы этапы 3 и 4 алгоритма поиска и устранения неисправностей для рассматриваемого случая. В первой части примера проиллюстрирован этап 3, в котором используется команда **traceroute 172.16.2.7** с подстановкой IP-адреса отправителя 10.1.1.1. Трассировка маршрута в данном случае обрывается на узле с адресом 10.1.13.3 (R3). На этапе 4 исследуется процесс маршрутизации в устройстве R3 пакетов к получателю с адресом 172.16.2.7.

Пример 7.3. Иллюстрация алгоритма поиска и устранения неисправностей: этапы 1–4

```

R1#traceroute
Protocol [ip]:
Target IP address: 172.16.2.7
Source address: 10.1.1.1
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose [none]:
Type escape sequence to abort.
Tracing the route to 172.16.2.7

```



```

1 10.1.13.3 0 msec 4 msec 0 msec
2 10.1.13.3 !H * !H
! Обратите внимание, что выполнение команды завершилось, но узел-
получатель
! с адресом 172.16.2.7 не был достигнут
R3#show ip route 172.16.2.7
% Subnet not in table
R3#show 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

172.16.0.0/24 is subnetted, 1 subnets
C    172.16.1.0 is directly connected, FastEthernet0/0
10.0.0.0/24 is subnetted, 4 subnets
C    10.1.13.0 is directly connected, Serial0/0/1
R    10.1.1.0 [120/1] via 10.1.13.1, 00:00:04, Serial0/0/1
R    10.1.0.0 [120/1] via 10.1.23.2, 00:00:01, Serial0/1/0
C    10.1.23.0 is directly connected, Serial0/1/0

```

В выводе расширенного варианта команды **traceroute** в начале примера можно увидеть, что трассировка маршрута прерывается на маршрутизаторе R3 (10.1.13.3) — этап 3. Далее, на этапе 4, исследуется маршрут к IP-адресу 172.16.2.7 в маршрутизаторе R3 с помощью команды **show ip route 172.16.2.7**. Сообщение “subnet not in table” (подсеть отсутствует в таблице) в выводе команды свидетельствует о том, что у данного маршрутизатора нет маршрута к указанному адресу. С помощью команды **show ip route** без параметров можно также убедиться, что в таблице маршрутизации нет маршрутов к искомой сети.

Как только в процессе изолирования проблемы обнаружено, что отсутствует какой-либо маршрут, следует попытаться определить, из какого источника маршрут должен быть получен устройством. В рассматриваемом примере в маршрутизаторе R3 запущен протокол RIP-2 и маршрут должен быть получен через него, поэтому на следующем этапе нужно искать неисправности уже в протоколе динамической маршрутизации.

Причина неисправности в данном случае не изменилась — у маршрутизатора R4 выключен интерфейс Fa0/0, но симптомы проблемы более интересные. Поскольку интерфейс не работает, маршрутизатор R4 не анонсирует маршрут к подсети 172.16.2.0/24 маршрутизатору R3. Маршрутизатор R3 анонсирует автоматически просуммированный маршрут 172.16.0.0/16 устройствам R1 и R2, поскольку в протоколе RIP-2 автоматическое суммирование маршрутов изначально включено и эти два маршрутизатора будут пересылать устройству R3 пакеты, предназначенные для сети 172.16.2.0/24. Именно благодаря такой настройке при выполнении команды **traceroute** в маршрутизаторе R1 пакеты достигают маршрутизатора R3.

Сценарий 2. Проблемы с маршрутом в обратном направлении

В этом примере используется та же самая сеть, которая показана на рис. 7.5. Тем не менее ситуация несколько изменилась, в частности причина отказа и проблемы в сети теперь другие — более интересные. Попробуем найти и устранить неисправность в такой сети согласно описанному выше алгоритму.

В данном сценарии компьютер ПК1, как и в предыдущем, не может связаться с адресом 172.16.2.7 (ПК4). Проверка настроек сети и стандартного шлюза для ПК1 согласно этапам 1 и 2 показывает, что они правильные и связь с маршрутизатором существует. Тем не менее проблемы в сети есть, поскольку установить связь с ПК4 невозможно и доступа как к компьютеру ПК4, так и к маршрутизатору R4 нет. В примере 7.4 сначала проиллюстрирован этап 3 алгоритма поиска и устранения неисправностей, для первичной проверки используется расширенный вариант команды **traceroute** в маршрутизаторе R1. В данном случае в выводе команды не отображается IP-адрес даже маршрутизатора R3 — 10.1.13.3. Далее в примере показаны действия, выполняемые на этапе 4.

Пример 7.4. Поиск и устранение неисправностей, сценарий 2: этапы 3 и 4

```
R1#traceroute ip 172.16.2.7 source fa0/0
Type escape sequence to abort.
Tracing the route to 172.16.2.7
 1  *  *  *
 2  *  *  *
 3  *
R1#show ip route 172.16.2.7
Routing entry for 172.16.0.0/16
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 10.1.13.3 on Serial0/1/0, 00:00:05 ago
  Routing Descriptor Blocks:
    * 10.1.13.3, from 10.1.13.3, 00:00:05 ago, via Serial0/1/0
      Route metric is 1, traffic share count is 1
R1#ping 10.1.13.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.13.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R1#show ip access-lists
```

Переключаемся на маршрутизатор R3

```
R3#show ip access-lists
```

```
R3#
```

Сначала в примере 7.4 выполняется этап 3 алгоритма, для него используется команда **traceroute**, в выводе которой отображаются только символы "*". Они означают, что команда не смогла идентифицировать даже ближайший транзитный маршрутизатор на маршруте.

Далее переходим к проверке этапа 4.

Этап 4, а В рассматриваемом примере с маршрутизатором R1 уже был установлен telnet-сеанс, поэтому заново устанавливать его не нужно

- Этап 4, б** Следующая команда в примере, `show ip route 172.16.2.7`, позволяет определить, что у маршрутизатора R1 есть нестандартный маршрут к сети 172.16.0.0, пролегающий через маршрутизатор R3 (10.1.13.3)
- Этап 4, в** В рассматриваемой ситуации данный этап не нужен, поскольку маршрут был определен на этапе 4, б
- Этап 4, г** В рассматриваемой ситуации данный этап не нужен, поскольку найденный маршрут не является стандартным — 0.0.0.0/0
- Этап 4, д** С помощью следующей команды, `ping 10.1.13.3`, выполняется проверка, может ли маршрутизатор R1 пересылать пакеты следующему на маршруте транзитному маршрутизатору, адрес которого был определен на этапе 4, б. Команда возвращает положительный результат
- Этап 4, е** С помощью команды `show ip access-lists` мы можем убедиться, что в обоих маршрутизаторах, R1 и R3, не установлены списки контроля доступа (ACL)

Итак, все вышеперечисленные действия позволили определить, что маршрут в исходящем направлении есть, поэтому переходим к этапу 5. В команде `traceroute` в примере 7.4 IP-адрес интерфейса Fa0/0 маршрутизатора R1, т.е. 10.1.1.1, использовался в качестве адреса отправителя. На этапе 5 необходимо проверить наличие маршрута от устройства R3 к этому же адресу, но теперь он будет использоваться в качестве адреса получателя. В примере 7.5 проиллюстрирован последний этап алгоритма.

Пример 7.5. Сценарий 2: этап 5

```
! Приведенная ниже команда показывает маршрут к подсети 10.1.1.0/26,
следующим
! транзитным узлом для которой является устройство с адресом
10.1.23.2.
R3#show ip route 10.1.1.1
Routing entry for 10.1.1.0/26
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
    * 10.1.23.2
      Route metric is 0, traffic share count is 1
! С помощью следующей команды мы обнаружим, что две подсети -
10.1.1.0/26
! и 10.1.1.0/24 — перекрываются.
R3#show 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

172.16.0.0/24 is subnetted, 2 subnets
```

```
C      172.16.1.0 is directly connected, FastEthernet0/0
R      172.16.2.0 [120/1] via 172.16.1.4, 00:00:18, FastEthernet0/0
      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C      10.1.13.0/24 is directly connected, Serial0/0/1
S      10.1.1.0/26 [1/0] via 10.1.23.2
R      10.1.1.0/24 [120/1] via 10.1.13.1, 00:00:10, Serial0/0/1
R      10.1.0.0/24 [120/1] via 10.1.23.2, 00:00:11, Serial0/1/0
C      10.1.23.0/24 is directly connected, Serial0/1/0
```

Мы обнаружили, что в маршрутизаторе R3 неправильно сконфигурирован статический маршрут к подсети 10.1.1.0/26. В эту сеть входят все адреса из диапазона 10.1.1.0–10.1.1.63, следовательно, и IP-адрес 10.1.1.1 — тоже. Когда маршрутизатор R3 пытается переслать пакет по обратному маршруту к получателю с адресом 10.1.1.1, он обнаруживает, что у него есть два маршрута к одной и той же подсети. Согласно правилам маршрутизации устройство R3 выбирает наиболее специфичный маршрут (т.е. маршрут с наиболее длинным префиксом), следовательно, он пересылает пакет согласно маршрутной записи для сети 10.1.1.0/26. В итоге маршрутизатор R3 пересылает все пакеты для получателя с адресом 10.1.1.1 маршрутизатору R2, а не R1.

В рассмотренном примере невозможно определить, с какой целью был создан именно такой статический маршрут, но после выполнения этапов алгоритма поиска и устранения неисправностей была обнаружена причина отказа — статический маршрут к сети 10.1.1.0/26 в маршрутизаторе R3. Если в локальной сети маршрутизатора R1 используются адреса из диапазона 10.1.1.0–10.1.1.255, то такой маршрут вполне можно удалить.

Альтернативный алгоритм изолирования проблемы для этапов 3–5

Этапы поиска и устранения неисправностей, связанные с изолированием и устранением ошибок в настройках маршрутизаторов, прежде всего, основываются на информации, получаемой от команды **traceroute**. С ее помощью можно найти отправную точку для изучения проблемы. В качестве альтернативы можно использовать две другие команды: **ping** и **telnet**. Тем не менее с их помощью нельзя так быстро определить наиболее вероятные источники проблем в сети. При использовании команд **ping** и **telnet** нужно выполнить больше действий на первом маршрутизаторе (т.е. стандартном шлюзе узла в локальной сети), затем выполнить те же действия на вышестоящем маршрутизаторе, на следующем и так до тех пор, пока не будет найдено проблемное устройство.

Итак, с помощью двух указанных команд можно выполнить проверки этапов 4 и 5. Например, чтобы получить ту же самую информацию, что и в рассмотренных выше примерах, следует подключиться к маршрутизатору R1, который является стандартным шлюзом компьютера ПК1. С помощью команды **ping** можно проверить, есть ли маршрут в исходящем направлении в маршрутизаторе R1 к получателю 172.16.2.7 (ПК4), использовать команду для проверки связи с вышестоящим маршрутизатором и проверить, есть ли в устройстве списки контроля доступа (ACL). Далее, убедившись, что в исходящем направлении маршрут есть, подключиться к маршрутизатору R3 и обнаружить, что у него нет маршрута к искомой сети.

Советы относительно поиска и устранения неисправностей и необходимый инструментарий

В этом разделе описаны разнообразные инструменты для поиска и устранения неисправностей, а также даны полезные советы, которые помогут облегчить и ускорить процесс восстановления работоспособности сети. Часть представленной информации напрямую связана с сертификационным экзаменом CCNA, а часть просто пригодится в практической работе.

Средства для проверки маршрутизации в оконечных узлах

В этом разделе кратко описаны две темы, в которых рассматривается процесс IP-маршрутизации в оконечных узлах. В первой теме перечислены некоторые полезные советы по поиску и устранению неисправностей в узлах, а вторая тема посвящена IP-настройкам коммутаторов локальных сетей.


Советы по устранению неисправностей в оконечных узлах

В табл. 7.4 перечислены ключевые признаки и наиболее распространенные причины неисправностей, которые с ними связаны. Эту таблицу можно распечатать и первое время использовать в практической работе. Следует помнить, что в таблице перечислены не абсолютно все возможные причины отказа, а только наиболее вероятные.

Таблица 7.4. Часто встречающиеся неисправности в оконечных узлах и их наиболее вероятные причины

Неисправность	Вероятная причина
Узел успешно пересылает пакеты другим узлам в той же самой подсети, но не может установить связь с удаленными подсетями	В настройках узла не указан стандартный шлюз (default gateway) или IP-адрес стандартного шлюза неправильный
Узел успешно пересылает пакеты другим узлам в той же самой подсети, но не может установить связь с удаленными подсетями	Стандартный шлюз для этого узла находится в другой логической подсети, т.е. адрес или маска в настройках узла указана неправильно
Часть узлов в какой-либо подсети может установить связь с узлами из других подсетей, а часть — нет	Причиной такого поведения могут быть настройки стандартного шлюза-маршрутизатора, в котором для интерфейса локальной сети указана маска, отличающаяся от маски, заданной узлам. В такой ситуации маршрут к напрямую подключенной сети в маршрутизаторе будет включать в себя не все узлы подсети
Часть узлов в какой-либо сети VLAN может установить связь с узлами из других подсетей, а часть — нет	Маски в настройках сети узлов разные

В процессе поиска и устранения неисправностей в практической работе первое, что нужно сделать, — определить симптомы. В сертификационных экзаменах компании Cisco решить проблему отсутствия связи у оконечного узла можно с помощью следующего алгоритма.


- Этап 1** Проверьте, находятся ли все маршрутизаторы и узлы в сегменте локальной сети в одной и той же логической подсети и одинаковые ли у них установлены маски 
- Этап 2** Сравните настройки стандартного шлюза каждого узла с конфигурацией соответствующего интерфейса маршрутизатора и убедитесь, что указаны правильные IP-адреса
- Этап 3** Если на первых двух этапах ошибок не выявлено, ищите неисправности на уровнях 1 и 2, как описано в главах 1–3

IP-настройки коммутатора локальной сети

Коммутаторы Ethernet для пересылки фреймов не используют функций уровня 3. Тем не менее для использования нескольких полезных технологий, например telnet- и SSH-служб, в коммутаторе устанавливается IP-адрес.

Коммутатор с IP-конфигурацией работает точно так же, как и обычный пользовательский узел, только в отличие от последнего в нем не используется отдельная сетевая карта. Вместо этого в устройстве создается внутренний виртуальный интерфейс, связанный с сетью VLAN 1, к которому и осуществляются все подключения. Для этого виртуального интерфейса конфигурируются все сетевые настройки, похожие на настройки персонального компьютера: IP-адрес, маска и стандартный шлюз. Дополнительно можно также указать IP-адрес DNS-сервера.

Ниже перечислены основные этапы конфигурирования IP-параметров коммутатора локальной сети, которые подробно описаны в томе I. В примере 7.6 проиллюстрирован процесс конфигурирования для коммутатора SW1 в схеме сети (см. рис. 7.5).

- Этап 1** Перейдите в режим конфигурирования сети VLAN 1 с помощью команды `interface vlan 1` в режиме глобальной конфигурации устройства 
- Этап 2** Задайте устройству IP-адрес с помощью команды `ip address ip-адрес маска` в режиме конфигурирования интерфейса
- Этап 3** Включите интерфейс сети VLAN 1 с помощью команды `no shutdown` в режиме конфигурирования интерфейса
- Этап 4** Укажите в режиме глобального конфигурирования стандартный шлюз с помощью команды `ip default-gateway ip-адрес`

Пример 7.6. Конфигурирование IP-адреса коммутатора

```
SW1#configure terminal
SW1(config)#interface vlan 1
SW1(config-if)#ip address 10.1.1.200 255.255.255.0
SW1(config-if)#no shutdown
00:25:07: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
00:25:08: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1,
changed
state to up
SW1(config-if)#exit
SW1(config)#ip default-gateway 10.1.1.1
```

ВНИМАНИЕ!

Следует помнить, что виртуальный интерфейс сети VLAN 1 находится в выключенном состоянии и обязательно нужно вводить команду **no shutdown**; в противном случае коммутатор не сможет обрабатывать предназначенные ему IP-пакеты.

Чаще всего проблемы со связью в коммутируемых IP-сетях связаны с магистральными соединениями между коммутаторами (trunking). Компания Cisco рекомендует не включать устройства конечных пользователей в сеть VLAN 1, но IP-адрес устройства устанавливается именно в этой сети. Чтобы получить доступ к устройству через telnet-сеанс, в вышестоящем маршрутизаторе должна быть правильно сконфигурирована маршрутизация между сетями VLAN.

Команда show ip route

Эта команда очень важна для успешного поиска и устранения неисправностей в IP-маршрутизации и связанных с ней протоколов. Команда **show ip route** неоднократно рассматривалась или упоминалась в предыдущих главах, а в текущем разделе все ее особенности и варианты использования сведены воедино.

На рис. 7.6 показан вывод команды **show ip route** (из примера 7.3), в котором наиболее важные элементы пронумерованы, а в табл. 7.5 приведено их описание.

```

      ①      ②      ③
  C  10.0.0.0/24 is subnetted, 4 subnets
  R   10.1.13.0 is directly connected, Serial0/0/1
  C   10.1.1.0 [120/1] via 10.1.13.1, 00:00:04, Serial0/0/1
  R   10.1.23.0 is directly connected, Serial0/1/0
  R   10.1.0.0 [120/1] via 10.1.23.2, 00:00:01, Serial0/1/0
  ④      ⑤      ⑥ ⑦      ⑧      ⑨      ⑩
  
```

Рис. 7.6. Вывод команды **show ip route**

Таблица 7.5. Описание вывода команды **show ip route**

Номер	Элемент	Значение элемента	Описание
1	Классовая сеть	10.0.0.0	В таблице маршрутизации отображаются классовые сети для подсетей, в первой строке в данном примере показана классовая сеть 10.0.0.0
2	Длина префикса	/24	Если в подсетях используется одна и та же маска подсети, то она показана в первой строке вывода. Обратите внимание, что используется префиксный формат
3	Количество подсетей	4 subnets	В первой строке также указано количество подсетей классовой сети, присутствующих в таблице маршрутизации устройства
4	Код маршрута	R, C	Короткий код, идентифицирующий источник маршрута. R означает, что маршрут был получен посредством протокола RIP, C — сеть напрямую подключена к устройству. На рис. 7.6 таблица кодов опущена, но она выводится в команде show ip route (см. пример 7.3)

Окончание табл. 7.5

Номер	Элемент	Значение элемента	Описание
5	Адрес подсети	10.1.0.0	Адрес конкретной подсети
6	Административное расстояние	120	Если маршрутизатор получает маршрут к одной и той же подсети из разных источников, в таблицу маршрутизации устанавливается маршрут с меньшим административным расстоянием
7	Метрика	1	Метрика полученного маршрута
8	Следующий на маршруте транзитный маршрутизатор	10.1.23.2	Это IP-адрес устройства, которому будут пересылаться пакеты, предназначенные указанной подсети
9	Таймер	00:00:01	Указывает время, прошедшее с момента последнего обновления данного маршрута
10	Выходной интерфейс	Serial0/1/0	Через этот интерфейс будут пересылаться пакеты, предназначенные указанной подсети

Вывод рассматриваемой команды будет немного отличаться при использовании маски переменной длины (VLSM). В рассмотренном примере в подсетях использовались одинаковые маски /24 для сети 10.0.0.0, поэтому маска указывается всего один раз в выводе команды. Если же используются маски VLSM, то в выводе команды будет указано, что сеть разбита на разные блоки (variably subnetted) и для каждого маршрута будет указана собственная маска. Подробнее маски VLSM описаны в главе 5, "Маски VLSM и суммирование маршрутов".

Коды состояния интерфейса

Выше было указано, что на одном из этапов поиска и устранения неисправностей в механизмах маршрутизации следует проверить состояние интерфейса маршрутизатора и убедиться в том, что он работает. Такая проверка означает, что с помощью соответствующей команды для интерфейса инженер должен увидеть два кода, свидетельствующие о том, что интерфейс включен ("up"), зачастую инженеры просто говорят "ап и ап", чтобы сократить фразу.

В этой главе подробности процесса устранения неисправностей в интерфейсах не рассматриваются. Обратитесь к главе 12, "Двухточечные каналы распределенных сетей", в которой подробно описаны последовательные интерфейсы и алгоритмы устранения неисправностей. При поиске неисправностей, если маршрутизатор подключен к коммутатору локальной сети, прежде всего нужно проверить, совпадают ли настройки дуплексности и скорости обоих устройств. Если же используется магистральное соединение (trunking), следует убедиться в том, что оно статически сконфигурировано как в маршрутизаторе, так и в коммутаторе, поскольку в маршрутизаторах нет динамических протоколов для установки и согласования магистрального соединения.

Проблемы в сетях VLAN

В этом разделе описаны три наиболее часто встречающиеся проблемы в сетях VLAN.

- Неправильное использование масок VLSM и протоколов маршрутизации
- Использование перекрывающихся подсетей в маршрутизаторах
- Неправильная маршрутизация при наличии перекрывающихся подсетей в сети

Неправильное использование масок VLSM и протоколов маршрутизации

Одна из основных ошибок в сети с виртуальными подсетями — это неправильная интерпретация масок VLSM или ошибки в них. Как было сказано в главе 5, VLSM подразумевает использование масок разной длины *в одной и той же классовой подсети*. Например, если в классовой сети 10.0.0.0 используется маска 255.255.240.0, а в сети 172.16.0.0 — 255.255.255.0 в одной и той же структуре сети, то это не означает, что для адресации применена технология VLSM. Если же в классовой сети 10.0.0.0 используются обе указанные маски, тогда можно говорить об использовании VLSM.

Следует помнить о том, что только в бесклассовых протоколах маршрутизации (RIP-2, EIGRP, OSPF) можно использовать маски VLSM; в классowych (RIP-1, IGRP) — нельзя. Поэтому прежде всего нужно определить, используются ли маски VLSM; после этого необходимо посмотреть, какой протокол маршрутизации используется, классовой или бесклассовой.

Использование перекрывающихся подсетей в маршрутизаторах

В правилах создания IP-подсетей требуется, чтобы диапазоны адресов подсетей в сети не перекрывались. Операционная система IOS распознает перекрывающиеся подсети, когда вводится команда **ip address**, но далеко не во всех случаях. В этом разделе описаны ситуации, в которых перекрывающиеся подсети могут быть сконфигурированы, а все возможные ситуации можно свести к следующим двум вариантам.

- **Перекрывающиеся подсети создать невозможно.** Операционная система Cisco IOS обнаруживает перекрывающиеся подсети при использовании команды **ip address**, если они конфигурируются *в одном и том же маршрутизаторе*. Если на каком-либо интерфейсе уже задана подсеть и пользователь пытается ее же указать на другом, а первый интерфейс находится в рабочем состоянии, операционная система IOS не примет вторую команду. Если первый интерфейс находится в нерабочем состоянии, то операционная система примет команду **ip address**, но включить интерфейс не удастся.
- **Перекрывающиеся сети создать можно.** Операционная система не сможет обнаружить перекрывающиеся подсети, если они заданы командой **ip address** *в разных маршрутизаторах*.

В примере 7.7 показано, что операционная система маршрутизатора не позволяет использовать перекрывающиеся VLSM-подсети в одном устройстве. В этом примере у маршрутизатора R3 для интерфейса Fa0/0 указан IP-адрес 172.16.5.1/24, а для ин-



терфейса Fa0/1 вводится адрес 172.16.5.193/26. Диапазоны адресов для таких подсетей будут равны.

- Подсеть 172.16.5.0/24: 172.16.5.1–172.16.5.254
- Подсеть 172.16.5.192/26: 172.16.5.193–172.16.5.254

Пример 7.7. Блокирование операционной системой попытки конфигурирования перекрывающихся подсетей

```
R3#configure terminal
R3 (config)#interface Fa0/0
R3 (config-if)#ip address 172.16.5.1 255.255.255.0
R3 (config-if)#interface Fa0/1
R3 (config-if)#ip address 172.16.5.193 255.255.255.192
% 172.16.5.192 overlaps with FastEthernet0/0
R3 (config-if)#
```

Операционная система IOS “знает”, что устанавливать на одном устройстве перекрывающиеся подсети запрещено правилами адресации. В рассматриваемом случае обе подсети представляют собой сегменты, напрямую подключенные к одному и тому же маршрутизатору, следовательно, они не должны пересекаться и операционная система блокирует вторую команду.

Тем не менее перекрывающиеся подсети можно сконфигурировать в разных устройствах. На рис. 7.7 показана схема сети, похожая на представленную на рис 5.2, которая использовалась для иллюстрации неправильной адресации. В примере 7.8 приведена конфигурация устройств для этой схемы, в которой в маршрутизаторах R2 и R3 создаются перекрывающиеся подсети, а ниже показана таблица маршрутизации устройства R2.

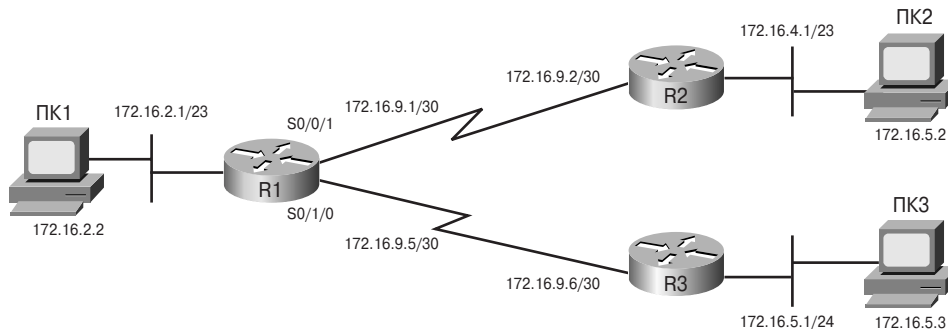


Рис. 7.7. Сеть с перекрывающимися подсетями

Пример 7.8. Перекрывающиеся подсети в двух маршрутизаторах

```
R2#configure terminal
R2 (config)#interface Fa0/0
R2 (config-if)#ip address 172.16.4.1 255.255.254.0
R3#configure terminal
R3 (config)#interface Fa0/0
R3 (config-if)# ip address 172.16.5.1 255.255.255.0
```

На сертификационном экзамене нужно помнить, что перекрывающиеся подсети могут быть непреднамеренно сконфигурированы на разных устройствах, но не на одном и том же. Поэтому, если в задании требуется выбрать адрес подсети и сконфигурировать интерфейс маршрутизатора, устройство может принять команду `ip address` с некорректным адресом, и маршрутизация при этом работать не будет.

В следующем разделе описаны признаки наличия в сети перекрывающихся подсетей.

Неправильная маршрутизация при наличии перекрывающихся подсетей в сети

ВНИМАНИЕ!

Этот раздел необходим для того, чтобы описание процесса поиска и устранения неисправностей в маршрутизации и адресации было наиболее полным. В сертификационном экзамене CCNA заданий, связанных с поиском и устранением перекрывающихся или дублирующихся подсетей, не будет.

Признаки проблем в сети могут несколько отличаться в зависимости от того, какая ошибка в конфигурации была допущена: были ли указаны два одинаковых адреса разным узлам или просто сконфигурированы перекрывающиеся подсети. Если в сети есть перекрывающиеся подсети, то коммуникация между узлами, находящимися в неперекрывающихся частях подсетей происходит как обычно. Обратимся к рассмотренному выше примеру (см. рис. 7.6). Подсети 172.16.4.0/23 и 172.16.5.0/24 перекрываются, а именно — перекрываются адреса из диапазона 172.16.5.0–172.16.5.255. При этом узлы в неперекрывающейся части сети, т.е. в диапазоне 172.16.4.0–172.16.4.255, будут взаимодействовать без проблем.

Адреса из меньшего блока в перекрывающихся подсетях, скорее всего, будут без проблем взаимодействовать с сетью, а адреса из большего блока (или диапазона) — не будут. Чтобы понять, почему для сети характерно такое поведение, представим себе, что компьютер ПК1 пытается обмениваться эхо-запросами с получателем 172.16.5.2 (ПК2), находящимся за маршрутизатором R2 и с получателем 172.16.5.3 (ПК3), расположенным за маршрутизатором R3 (см. рис. 7.7). Чтобы упростить задачу, предположим, что адреса компьютеров ПК2 и ПК3 не дублируются, т.е. есть только перекрытие подсетей. В примере 7.9 показан вывод команды `traceroute 172.16.5.2` и таблицы маршрутизации устройств R1 и R3, из которых видно, что пакеты, пересылаемые ПК1 маршрутизатору R2, в действительности сначала отправляются устройством R1 маршрутизатору R3, а затем пересылаются в локальную сеть последнего.

Пример 7.9. Перекрывающиеся подсети в двух маршрутизаторах

```
! Маршрут от устройства R1 к получателю 172.16.5.2 указывает на R3
R1#show ip route 172.16.5.2
Routing entry for 172.16.5.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 172.16.9.6 on Serial0/1/0, 00:00:25 ago
  Routing Descriptor Blocks:
    * 172.16.9.6, from 172.16.9.6, 00:00:25 ago, via Serial0/1/0
```

```
Route metric is 1, traffic share count is 1
! Маршрут от устройства R1 к получателю 172.16.5.3 указывает на R3
R1#show ip route 172.16.5.3
Routing entry for 172.16.5.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 172.16.9.6 on Serial0/1/0, 00:00:01 ago
  Routing Descriptor Blocks:
  * 172.16.9.6, from 172.16.9.6, 00:00:01 ago, via Serial0/1/0
    Route metric is 1, traffic share count is 1
! Команда tracert к узлу PC2 показывает в качестве следующего
! транзитного устройства R3, а не маршрутизатор R2, поэтому пакет никогда
! не достигнет узла PC2, а завершить команду можно, прервав ее выполнение.

R1#tracert 172.16.5.2

Type escape sequence to abort.
Tracing the route to 172.16.5.2

  1  172.16.9.6  4 msec  0 msec  4 msec
  2  * * *
  3  * * *
  4

R1#tracert 172.16.5.3

Type escape sequence to abort.
Tracing the route to 172.16.5.3

  1  172.16.9.6  0 msec  4 msec  0 msec
  2  172.16.5.3  4 msec  *  0 msec
```

В этом примере проиллюстрирована ситуация, в которой маршрутизатор R1 пересылает пакеты узлам с адресами 172.16.5.2 (ПК2) и 172.16.5.3 (ПК3) через маршрутизатор R3. Устройство R3 пытается переслать эти пакеты в свою локальную сеть. Для компьютера ПК3 это удается, а для ПК2 — нет. Компьютер ПК3 находится в меньшем блоке из двух перекрывающихся подсетей, а ПК2 — нет.

Симптомы могут быть более сложными, если в рассматриваемой сети есть дублирующиеся адреса узлов. Представим себе, что мы добавили в локальную сеть маршрутизатора R2 ПК2 с адресом 172.16.5.3, совпадающим с IP-адресом компьютера ПК3. Пользователь ПК2, вполне очевидно, будет жаловаться, что у него не работает сеть, но если сетевой инженер запустит команду **ping 172.16.5.3**, то она даст положительный результат! Команда будет обмениваться запросами и ответами с "неправильным" узлом, но результат будет выглядеть так, как будто все отлично работает! Теперь мы видим, что правильно диагностировать такую проблему неимоверно сложно.

Вторая сложность в обнаружении перекрывающихся VLSM-подсетей состоит в том, что проблема может не присутствовать постоянно или проявиться позже. Для той же самой схемы сети представим себе, что все адреса в обеих подсетях были назначены DHCP-сервером, причем адреса выдавались начиная с меньшего значения (стандартный режим работы). Первые 6 месяцев сервер присваивал узлам IP-адреса вида 172.16.4.x в локальной сети маршрутизатора R2. Наконец, количество узлов в сегменте локальной сети стало достаточно большим, и сервер начал выдавать адреса вида 172.16.5.x. В такой локальной сети устройства не смогут обмениваться пакетами

с внешними узлами. Поскольку проблема проявилась после того, как прошло, например, полгода после момента настройки сети, обнаружить ошибку и устранить ее может быть очень сложно.

Резюме по поиску и устранению неисправностей в VLSM-подсетях

Ниже приведены ключевые моменты процесса поиска и устранения неисправностей в VLSM-подсетях.

- Проверьте, действительно ли в сети используются маски VLSM. Если используются, проверьте, бесклассовый ли протокол маршрутизации сконфигурирован. Ключевая тема
- Проверьте, не сконфигурированы ли перекрывающиеся подсети в сети.
- Первый признак того, что в сети есть перекрывающиеся подсети, — сетевые службы в части узлов работают корректно, а часть узлов не может установить связь с устройствами вне своего сегмента локальной сети.
- С помощью команды **traceroute** проверьте, в правильном ли направлении отправляют маршрутизаторы пакеты. Если маршрут неправильный, то одна из наиболее вероятных причин — перекрывающиеся подсети.
- В сертификационных экзаменах часто встречаются вопросы, связанные с масками VLSM и IP-адресацией. Проанализируйте все подсети в таких вопросах, чтобы убедиться, что они не перекрываются, и помните, что не всегда команды **ping** и **traceroute** могут помочь найти и устранить проблему.

Несвязные сети и автоматическое суммирование маршрутов

В главе 5 описывалась концепция несвязных сетей и предлагалось решение, если с ними возникают проблемы: использовать бесклассовый протокол маршрутизации с отключенной функцией автоматического суммирования маршрутов. В этом разделе рассматривается пример, в котором используются две классовые сети: 10.0.0.0 и 172.16.0.0. Схема сети показана на рис. 7.8; обратите внимание, что с точки зрения маршрутизации, если все каналы работают, в такой структуре несвязных подсетей нет.

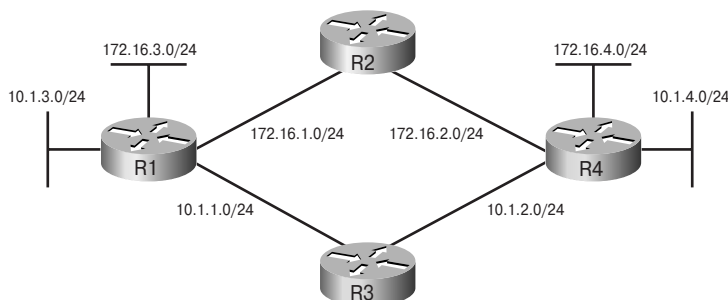


Рис. 7.8. Схема сети для примера несвязных подсетей

В сети, показанной на рис. 7.8, если все каналы находятся в рабочем состоянии и используется протокол маршрутизации с автоматическим суммированием маршрутов

тов (стандартное поведение), все узлы могут обмениваться друг с другом пакетами. В такой сети пакеты к получателям с адресами 172.16.0.0 будут пересылаться через маршрутизатор R2, а к сети 10.0.0.0 — через R1.

К несчастью, проблемы возникнут, если один из четырех каналов между маршрутизаторами откажет. Если хотя бы один из каналов в этой сети разрывается, то одна из двух классовых сетей становится несвязной. Предположим, что канал между маршрутизаторами R3 и R4 перестал функционировать, маршрут от маршрутизатора R1 к R4 проходит через подсети сети 172.16.0.0 и сеть 10.0.0.0 становится несвязной. Даже если в сети используется бесклассовый протокол маршрутизации, но включено автоматическое суммирование маршрутов, оба устройства, R1 и R4, будут анонсировать сеть 10.0.0.0/8 маршрутизатору R2. Последний, в свою очередь, “знает” два маршрута к сети 10.0.0.0: один — через маршрутизатор R1, другой — через маршрутизатор R4. Чтобы решить эту проблему, рекомендуется использовать бесклассовый протокол маршрутизации и отключить автоматическое суммирование маршрутов.

Несмотря на то что дизайн сети, показанный на рис. 7.8, кажется нелогичным и неестественным, он встречается намного чаще, чем может представить себе читатель (чаще всего как результат покупки, продажи или слияния компаний). Следует помнить о том, что в сетях могут быть несвязные подсети; эта информация пригодится как на сертификационном экзамене, так и в практической работе.

Советы по устранению ошибок в списках контроля доступа

Поиск и устранение проблем в сети, связанных со списками контроля доступа (ACL), — это одна из самых сложных задач, с которыми приходится сталкиваться инженеру в практической работе. Самая большая сложность состоит в том, что такие распространенные утилиты, как **ping** и **traceroute**, не пересылают пакеты, которые могут попадать под правила списков контроля доступа, в частности под правила расширенных списков. Поэтому, несмотря на то что, например, команда **ping** выдает положительный результат, компьютер пользователя все равно не имеет доступа к нужной ему службе или служба не может обратиться к какому-либо узлу.

Ниже описан алгоритм, с помощью которого следует искать и устранять ошибки в списках контроля доступа (ACL).

- Этап 1** Определите, на каких интерфейсах установлены списки контроля доступа и в каком направлении (с помощью команд **show running-config, show ip interfaces**)
- Этап 2** Определите, какие правила списков срабатывают (с помощью команд **show access-lists, show ip access-lists**)
- Этап 3** Проанализируйте списки ACL, чтобы предсказать, какие пакеты будут срабатывать согласно каким правилам, учитывая указанные ниже особенности работы списков.
 - а) В списках ACL проверка правил происходит до первого срабатывания.
 - б) Чтобы упростить процесс поиска соответствий в правилах, можно преобразовать записи адреса и инверсной маски в списках контроля доступа в стандартный формат адреса и маски, как было показано в главе 6, “Списки управления доступом”.



в) Определите направление потока пакетов относительно сервера (пакет пересылается серверу или от сервера). Убедитесь, что для соответствующего потока пакетов правильно указан IP-адрес отправителя и порт или IP-адрес получателя и порт получателя и направление списка ACL совпадает с направлением потока пакетов.

г) Проверьте, правильно ли используются ключи `tcp` и `udp` в расширенных списках, правила которых проверяют номера портов. (В табл. 6.5 перечислены наиболее популярные номера портов TCP и UDP.)

д) Помните, что ICMP-пакеты не используют службы TCP и UDP, поэтому для фильтрации таких пакетов нужно указывать ключ `icmp` (а не `ip`, `tcp` или `udp`).

е) Вместо неявного запрещающего правила в конце списков контроля доступа укажите запрещающее или разрешающее правило в явном виде, чтобы с помощью команды `show` можно было увидеть счетчики пакетов, отброшенных или переданных согласно такому правилу

В главе 6 описана теория, которая понадобится на этапе 3 приведенного выше алгоритма. Ниже перечислены некоторые полезные команды, которые помогут читателю найти ошибки в списках контроля доступа.

Итак, если при передаче IP-пакета возникают какие-либо проблемы и есть подозрение, что они связаны с установленными в конфигурации списками ACL, прежде всего, следует определить, на каком интерфейсе установлен список контроля доступа и в каком направлении. Быстрее всего можно выполнить такое действие с помощью команды `show running-config`, в выводе которой следует поискать команды `ip access-group` для интерфейсов. В некоторых случаях режим привилегированного пользователя может быть недоступен, поэтому придется использовать другие команды группы `show`. Как определить, в каком направлении, исходящем или входящем, сконфигурирован IP-список контроля доступа с помощью команды `show ip interfaces`, показано в примере 7.10.

Пример 7.10. Вывод команды `show ip interface`

```
R1>show ip interface s0/0/1
Serial0/0/1 is up, line protocol is up
  Internet address is 10.1.2.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.9
  Outgoing access list is not set
  Inbound access list is 102
```

! Около 26 строк вывода опущены для краткости

Обратите внимание, что показанная выше команда выводит информацию о том, включен ли список ACL, в каком направлении и его номер или название. В примере 7.10 демонстрируется вариант команды с указанием конкретного интерфейса `show ip interface S0/0/1`, чтобы уменьшить количество выводимой инфор-

мации. Команда **show ip interface** выводит ту же самую информацию, но для всех интерфейсов устройства.

На этапе 2 алгоритма поиска ошибок в списках контроля доступа следует посмотреть сам список. Опять же, быстрее всего можно выполнить такое действие с помощью команды **show running-config**. Если режим привилегированного пользователя недоступен, то аналогичный результат можно получить с помощью команд **show access-lists** и **show ip access-lists**. Единственное отличие этих двух команд: первая команда покажет все списки, для любых протоколов третьего уровня, вторая покажет только IP-списки контроля доступа. В рассматриваемом случае обе команды будут выводить ту же информацию, которая показана в примере 7.11. Обратите внимание, что в выводе команды также присутствует счетчик пакетов, которые попали под соответствующее правило списка контроля доступа.

Пример 7.11. Вывод команды show ip access-lists

```
R1#show ip access-lists
Extended IP access list 102
  10 permit ip 10.1.2.0 0.0.0.255 10.1.1.0 0.0.0.255 (15 matches)
```

После того как были определены интерфейс, к которому привязан список контроля доступа, и направление списка (т.е. выполнены этапы 1 и 2), начинается самая сложная часть процесса — интерпретация списка ACL. Всегда обращайтесь внимание на значения счетчиков в списках контроля доступа. Так, в примере 7.11 мы видим, что 15 пакетов по своим характеристикам совпали с указанным правилом в IP-списке с номером 102. Часть пакетов, вполне вероятно, была отброшена, поскольку в конце списка контроля доступа есть неявное запрещающее правило, которое в выводе команды не отображается. Сконфигурировав правило **access-list 102 deny ip any any** для такого списка ACL, которое выполняет те же самые функции, что и неявное правило, мы в выводе команды **show ip access-lists** увидели бы счетчик отброшенных пакетов. Компания Cisco зачастую рекомендует указывать явное запрещающее правило в конце каждого списка контроля доступа вместо неявного, чтобы упростить процесс поиска ошибок в списках ACL.

Подготовка к экзамену

Ключевые темы

Следует повторить ключевые темы данной главы, которые помечены пиктограммой на полях. В табл. 7.12 перечислены основные ключевые темы и соответствующие им страницы.



Таблица 7.12. Ключевые темы главы 7

Элемент ключевой темы	Описание	Страница
Табл. 7.1	Часто встречающиеся ICMP-сообщения и их назначение	331
Рис. 7.3	Принцип использования значения TTL в IP-заголовке пакета	335
Рис. 7.4	Иллюстрация использования поля TTL в команде <code>tracert</code>	337
Список	Два этапа изолирования проблем в маршрутизации для конечного узла	339
Список	Три этапа изолирования проблем с IP-маршрутизацией в сети	341
Список	Три этапа алгоритма поиска и устранения неисправностей в передаче пакетов конечным узлом	349
Список	Этапы конфигурирования IP-настроек коммутатора локальной сети	349
Список	Условия, при которых можно сконфигурировать перекрывающиеся подсети	352
Список	Ключевые моменты процесса поиска и устранения ошибок в конфигурации масок VLSM	356
Список	Три этапа алгоритма поиска и устранения ошибок в списках ACL	357

Заполните таблицы и списки по памяти

Распечатайте приложение К (Appendix J) с компакт-диска или его раздел, относящийся к этой главе, и заполните таблицы и списки по памяти. В приложении Л (Appendix K) приведены заполненные таблицы и списки для самопроверки.

Ключевые термины

Дайте определения перечисленных ниже терминов и проверьте правильность их написания в словаре терминов.

маршрут в исходящем направлении, обратный маршрут