

Общие сведения о командах ping и traceroute

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Общие сведения](#)

[Команда ping](#)

[Почему не работает команда ping?](#)

[Проблема маршрутизации](#)

[Интерфейс недоступен](#)

[Команда access-list](#)

[Проблемы с протоколом разрешения адресов \(ARP\)](#)

[Задержка](#)

[Исправьте исходный адрес](#)

[Высокое отбрасывание входящей очереди](#)

[Команда трассировки](#)

[Производительность](#)

[Использование команды Debug](#)

[Дополнительные сведения](#)

Введение

В данном документе рассматривается использование команд ping и traceroute. Некоторые команды отладки обеспечивают более подробный обзор функционирования этих команд, содержащийся в данном документе.

Примечание: Включение возможности использования команд отладки на производственном маршрутизаторе может привести к серьезным проблемам. [Перед использованием команд debug](#) рекомендуется тщательно ознакомиться с разделом [Использование команды Debug](#).

Предварительные условия

Требования

Для этого документа отсутствуют особые требования.

Используемые компоненты

Настоящий документ не имеет жесткой привязки к каким-либо конкретным версиям программного обеспечения и оборудования.

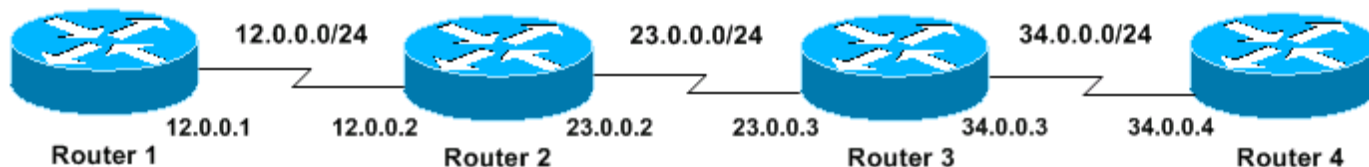
Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Условные обозначения

[Дополнительные сведения об условных обозначениях см. в документе Технические рекомендации Cisco. Условные обозначения.](#)

Общие сведения

В данном документе используется простая конфигурация, представленная ниже и используемая в качестве примера:



Команда ping

Команда ping является общим методом для поиска и устранения неисправностей устройств, к которым имеется доступ. Эта команда использует серию эхо-пакетов протокола управляющих сообщений в сети Интернет (ICMP-протокол) для определения:

- Является ли удаленный главный компьютер активным или неактивным.
- Задержка RTT при взаимодействии с хостом.
- Потеря пакета.

Команда ping сначала посылает пакет эхо-запроса на адрес, а затем ожидает ответа. Эхо-запрос может быть успешным, только если:

- эхо-запрос достигает цели, и
- место назначения может получить эхо-ответ обратно на источник в пределах предустановленного времени, которое называют временем ожидания. Для маршрутизаторов Cisco стандартное значение времени ожидания равно двум секундам.

[Для получения информации по всем параметрам данной команды см. раздел "Ping" главы "Команды устранения неполадок".](#)

Значение TTL ping - пакета не может быть изменено.

Пример выходных данных команды ping после разрешения использования команды debug ip packet detail выглядит следующим образом:

% Warning: Использование команды `debug ip packet detail` на производственном маршрутизаторе может вызвать высокую загрузку ЦП. Это может привести к серьезному падению производительности или выходу сети из строя. [Рекомендуется тщательно ознакомиться с разделом "Использование команды Debug перед использованием команд отладки."](#)

```
Router1#debug ip packet detail
IP packet debugging is on (detailed)
```

```
Router1#ping 12.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms
```

```
Router1#
Jan 20 15:54:47.487: IP: s=12.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
  sending
Jan 20 15:54:47.491: ICMP type=8, code=0
!--- This is the ICMP packet 12.0.0.1 sent to 12.0.0.2. !--- ICMP type=8 corresponds to the echo
message. Jan 20 15:54:47.523: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 100, rcvd 3
Jan 20 15:54:47.527: ICMP type=0, code=0
!--- This is the answer we get from 12.0.0.2. !--- ICMP type=0 corresponds to the echo reply
message. !--- By default, the repeat count is five times, so there will be five !--- echo
requests, and five echo replies.
```

В приведенной ниже таблице перечислены возможные значения типа ICMP.

Т и п И С М Р	Литерал
0	эхо-ответ
3	целевой недостижимый код 0 = сетевой недостижимый 1 = недостижимый узел 2 = протокол недостижимые 3 = порт недостижимые 4 = необходимая фрагментация и DF установил 5 = подведенный исходный маршрут
4	source Quench
5	код 0 перенаправления = перенаправляет дейтаграммы для сети 1 = дейтаграммы перенаправления для хоста 2 = дейтаграммы перенаправления для типа сервиса и сети 3 = дейтаграммы перенаправления для типа сервиса и хоста
6	альтернативно-адресный
8	эхо
9	объявление маршрутизатора
1 0	запрос маршрутизатора
1 1	код 0 time-exceeded = время жизни превысило в транзите 1 =, время повторной сборки фрагмента превысило

1 2	проблема с параметром
1 3	запрос о метке времени
1 4	отклик со штампом времени
1 5	запрос информации
1 6	информационный ответ
1 7	запрос маски
1 8	mask-reply
3 1	ошибка преобразования
3 2	mobile-redirect

В приведенной ниже таблице перечислены возможные символы выходных данных, полученных от средства проведения эхо-теста:

Символ	Описание
!	Каждый восклицательный знак указывает на получение ответа.
.	. Каждая точка указывает на тайм-аут сетевого сервера во время ожидания ответа.
U	Был получен целевой недостижимый ошибочный PDU.
Вопрос.	Гашение источников (направление перегружено).
M	Не мог фрагментировать.
?	Неизвестный тип пакета.
&	& Превышено время существования пакета.

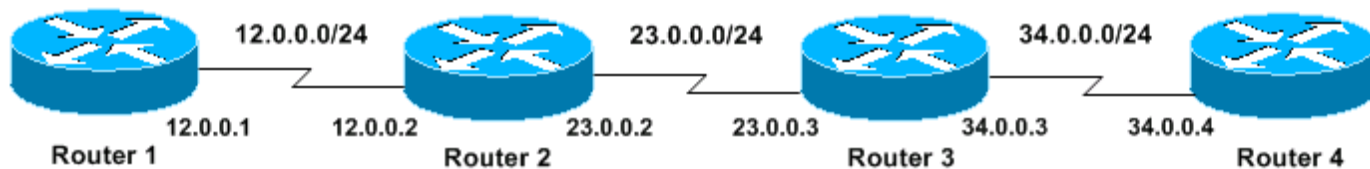
[Почему не работает команда ping?](#)

Если не удастся успешно выполнить команду ping, то причиной этому могут быть:

[Проблема маршрутизации](#)

Вот примеры неуспешных попыток эхо-запроса, определяя проблему, и что сделать для решения проблемы.

Данный сценарий объясняется с помощью схемы топологии сети, приведенной ниже:



Router1#

```

!
!
interface Serial0
ip address 12.0.0.1 255.255.255.0
no fair-queue
clockrate 64000
!
!
  
```

Router2#

```

!
!
interface Serial0
ip address 23.0.0.2 255.255.255.0
no fair-queue
clockrate 64000
!
interface Serial1
ip address 12.0.0.2 255.255.255.0
!
!
  
```

Router3#

```

!
!
interface Serial0
ip address 34.0.0.3 255.255.255.0
no fair-queue
!
interface Serial1
ip address 23.0.0.3 255.255.255.0
!
!
  
```

Router4#

```

!
!
interface Serial0
ip address 34.0.0.4 255.255.255.0
no fair-queue
clockrate 64000
!
!
  
```

В приведенном ниже примере производится опрос маршрутизатора 4 с маршрутизатора 1 с помощью команды ping:

Router1#ping 34.0.0.4

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:
.....
  
```

Success rate is 0 percent (0/5)

Давайте внимательнее посмотрим на произошедшее:

```
Router1#debug ip packet
IP packet debugging is on
```

% Warning: Использование команды debug ip packet на производственном маршрутизаторе может привести к большой нагрузке процессора. Это может привести к серьезному падению производительности или выходу сети из строя. [Рекомендуется тщательно ознакомиться с разделом "Использование команды Debug перед использованием команд отладки."](#)

```
Router1#ping 34.0.0.4
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:
```

```
Jan 20 16:00:25.603: IP: s=12.0.0.1 (local), d=34.0.0.4, len 100, unroutable.
Jan 20 16:00:27.599: IP: s=12.0.0.1 (local), d=34.0.0.4, len 100, unroutable.
Jan 20 16:00:29.599: IP: s=12.0.0.1 (local), d=34.0.0.4, len 100, unroutable.
Jan 20 16:00:31.599: IP: s=12.0.0.1 (local), d=34.0.0.4, len 100, unroutable.
Jan 20 16:00:33.599: IP: s=12.0.0.1 (local), d=34.0.0.4, len 100, unroutable.
Success rate is 0 percent (0/5)
```

Поскольку протоколы маршрутизации не используются в маршрутизаторе 1, он не знает, куда посылать пакеты и создает сообщение о немаршрутизации.

Добавим маршрутизатору 1 статический маршрут:

```
Router1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router1(config)#ip route 0.0.0.0 0.0.0.0 Serial0
```

Теперь у нас имеется:

```
Router1#debug ip packet detail
IP packet debugging is on (detailed)
```

```
Router1#ping 34.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

```
Jan 20 16:05:30.659: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
    sending
Jan 20 16:05:30.663:      ICMP type=8, code=0
Jan 20 16:05:30.691: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 56,
    rcvd 3
Jan 20 16:05:30.695:      ICMP type=3, code=1
Jan 20 16:05:30.699: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
    sending
Jan 20 16:05:30.703:      ICMP type=8, code=0
Jan 20 16:05:32.699: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
    sending
Jan 20 16:05:32.703:      ICMP type=8, code=0
Jan 20 16:05:32.731: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 56,
    rcvd 3
Jan 20 16:05:32.735:      ICMP type=3, code=1
Jan 20 16:05:32.739: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
    sending
Jan 20 16:05:32.743:      ICMP type=8, code=0
```

Теперь оценим неисправности, возникающие на маршрутизаторе 2:

```
Router2#debug ip packet detail
```

```
IP packet debugging is on (detailed)
```

```
Router2#
```

```
Jan 20 16:10:41.907: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:41.911: ICMP type=8, code=0
Jan 20 16:10:41.915: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:41.919: ICMP type=3, code=1
Jan 20 16:10:41.947: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:41.951: ICMP type=8, code=0
Jan 20 16:10:43.943: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:43.947: ICMP type=8, code=0
Jan 20 16:10:43.951: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:43.955: ICMP type=3, code=1
Jan 20 16:10:43.983: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:43.987: ICMP type=8, code=0
Jan 20 16:10:45.979: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:45.983: ICMP type=8, code=0
Jan 20 16:10:45.987: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:45.991: ICMP type=3, code=1
```

Маршрутизатор1 правильно отправляет свои пакеты на маршрутизатор2, но маршрутизатор2 не знает, как получить доступ к адресу 34.0.0.4. Маршрутизатор 1 отправляет Маршрутизатору 1 сообщение "unreachable ICMP".

Теперь включите протокол информации маршрутизации (RIP) в маршрутизаторе 2 и маршрутизаторе 3:

```
Router2#debug ip packet detail
```

```
IP packet debugging is on (detailed)
```

```
Router2#
```

```
Jan 20 16:10:41.907: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:41.911: ICMP type=8, code=0
Jan 20 16:10:41.915: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:41.919: ICMP type=3, code=1
Jan 20 16:10:41.947: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:41.951: ICMP type=8, code=0
Jan 20 16:10:43.943: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:43.947: ICMP type=8, code=0
Jan 20 16:10:43.951: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:43.955: ICMP type=3, code=1
Jan 20 16:10:43.983: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:43.987: ICMP type=8, code=0
Jan 20 16:10:45.979: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:45.983: ICMP type=8, code=0
Jan 20 16:10:45.987: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:45.991: ICMP type=3, code=1
```

Теперь мы имеем:

```
Router1#debug ip packet
```

```
IP packet debugging is on
```

```
Router1#ping 34.0.0.4
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:
```

```
Jan 20 16:16:13.367: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
sending.
Jan 20 16:16:15.363: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
sending.
Jan 20 16:16:17.363: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
```

```
sending.  
Jan 20 16:16:19.363: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,  
sending.  
Jan 20 16:16:21.363: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,  
sending.  
Success rate is 0 percent (0/5)
```

Это немного лучше. Маршрутизатор 1 отправляет пакеты на маршрутизатор 4, но не получает ответа.

Рассмотрим, какие проблемы могли возникнуть на маршрутизаторе 4:

```
Router4#debug ip packet  
IP packet debugging is on
```

```
Router4#  
Jan 20 16:18:45.903: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,  
rcvd 3  
Jan 20 16:18:45.911: IP: s=34.0.0.4 (local), d=12.0.0.1, len 100, unroutable  
Jan 20 16:18:47.903: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,  
rcvd 3  
Jan 20 16:18:47.907: IP: s=34.0.0.4 (local), d=12.0.0.1, len 100, unroutable  
Jan 20 16:18:49.903: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,  
rcvd 3  
Jan 20 16:18:49.907: IP: s=34.0.0.4 (local), d=12.0.0.1, len 100, unroutable  
Jan 20 16:18:51.903: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,  
rcvd 3  
Jan 20 16:18:51.907: IP: s=34.0.0.4 (local), d=12.0.0.1, len 100, unroutable  
Jan 20 16:18:53.903: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,  
rcvd 3  
Jan 20 16:18:53.907: IP: s=34.0.0.4 (local), d=12.0.0.1, len 100, unroutable
```

Маршрутизатор 4 получает пакеты ICMP и пытается ответить 12.0.0.1, но так как у него нет маршрута в эту сеть, он просто терпит неудачу.

Добавим маршрутизатору 4 статический маршрут:

```
Router4(config)#ip route 0.0.0.0 0.0.0.0 Serial0
```

Теперь он работает правильно и обе стороны имеют доступ друг к другу:

```
Router1#ping 34.0.0.4
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/35/36 ms
```

[Интерфейс недоступен](#)

Это - ситуация, где интерфейс прекращает работать. В приведенном выше примере производится ping маршрутизатора 4 с маршрутизатора 1:

```
Router1#ping 34.0.0.4
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:  
U.U.U  
Success rate is 0 percent (0/5)
```

Поскольку маршрутизация в порядке, устранение неполадок будет выполняться в пошаговом режиме. В начале попытаемся применить команду ping к маршрутизатору 2:


```
Router1#ping 12.0.0.2
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

В приведенных выше данных видно, что проблема лежит между маршрутизатором 2 и маршрутизатором 3. Одна из возможных причин – то, что последовательный интерфейс на Router3 был отключен:

```
Router3#show ip interface brief
```

```
Serial0  34.0.0.3    YES manual up          up
Serial1  23.0.0.3    YES manual administratively down  down
```

Эта проблема легко устранима:

```
Router3#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router3(config)#interface s1
```

```
Router3(config-if)#no shutdown
```

```
Router3(config-if)#
```

```
Jan 20 16:20:53.900: %LINK-3-UPDOWN: Interface Serial1, changed state to up
```

```
Jan 20 16:20:53.910: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1,
changed state to up
```

[Команда access-list](#)

В этом сценарии необходимо разрешить трафику telnet поступать в маршрутизатор 4 через интерфейс Serial0 .

```
Router4(config)# access-list 100 permit tcp any any eq telnet
```

```
Router4(config)#interface s0
```

```
Router4(config-if)#ip access-group 100 in
```

```
Router1#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router1(config)#access-list 100 permit ip host 12.0.0.1 host 34.0.0.4
```

```
Router1(config)#access-list 100 permit ip host 34.0.0.4 host 12.0.0.1
```

```
Router1(config)#end
```

```
Router1#debug ip packet 100
```

```
IP packet debugging is on
```

```
Router1#debug ip icmp
```

```
ICMP packet debugging is on
```

[Список доступа команд отладки см. раздел Использование команды debug.](#)

В случае выполнения попытки проверить доступность маршрутизатора 4 будет получен следующий результат:

```
Router1#ping 34.0.0.4
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:
```

```
U.U.U
```

```
Success rate is 0 percent (0/5)
```

```
Jan 20 16:34:49.207: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
sending
```

```
Jan 20 16:34:49.287: IP: s=34.0.0.4 (Serial0), d=12.0.0.1 (Serial0), len 56,
rcvd 3
```

```
Jan 20 16:34:49.291: ICMP: dst (12.0.0.1) administratively prohibited unreachable
```

```

rcv from 34.0.0.4
Jan 20 16:34:49.295: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
  sending
Jan 20 16:34:51.295: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
  sending
Jan 20 16:34:51.367: IP: s=34.0.0.4 (Serial0), d=12.0.0.1 (Serial0), len 56,
  rcvd 3
Jan 20 16:34:51.371: ICMP: dst (12.0.0.1) administratively prohibited unreachable
  rcv from 34.0.0.4
Jan 20 16:34:51.379: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
  sending

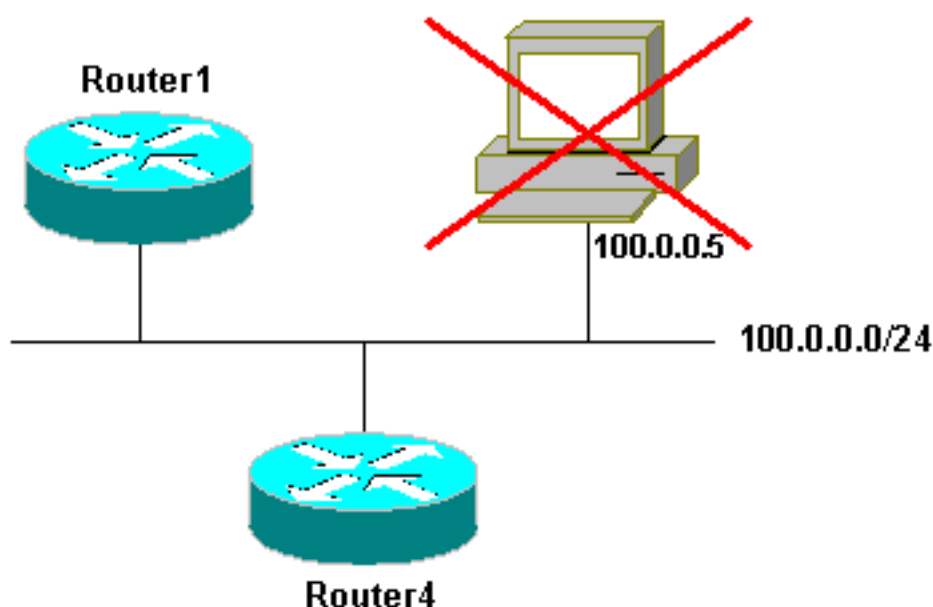
```

В конце команды списка доступа всегда существует неявное условие "deny all". Это означает, что ICMP пакеты, поступающие в интерфейс серии 0 маршрутизатора 4, отклонены, и маршрутизатор 4 посылает сообщения "административно запрещен недоступен" источнику оригинального пакета, как показано в сообщении отладки. Решением проблемы является добавление в команду access-list следующей строки:

```
Router4(config)#access-list 100 permit icmp any any
```

[Проблемы с протоколом разрешения адресов \(ARP\)](#)

Вот сценарий с Подключением по технологии Ethernet:



```
Router4#ping 100.0.0.5
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 100.0.0.5, timeout is 2 seconds:
```

```

Jan 20 17:04:05.167: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:05.171: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:07.167: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:07.171: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:09.175: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,

```

```

sending
Jan 20 17:04:09.183: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
encapsulation failed.
Jan 20 17:04:11.175: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
sending
Jan 20 17:04:11.179: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
encapsulation failed.
Jan 20 17:04:13.175: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
sending
Jan 20 17:04:13.179: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
encapsulation failed.
Success rate is 0 percent (0/5)
Router4#

```

В данном примере команда ping не работает из-за "неудачной инкапсуляции". Это означает, что маршрутизатору известно, на какой интерфейс следует отправить пакет, но неизвестно, каким образом это сделать. В этом случае необходимо понять принцип функционирования ARP-протокола. [Дополнительные сведения см. в документе Настройка методов разрешения адресов.](#)

ARP, в основном, - это протокол, используемый для сопоставления адреса второго уровня (MAC-адреса) с адресом третьего уровня (IP-адресом). Для проверки этого отображения можно использовать команду show arp:

```

Router4#show arp
Protocol Address          Age (min)  Hardware Addr  Type   Interface
Internet 100.0.0.4              -          0000.0c5d.7a0d  ARPA   Ethernet0
Internet 100.0.0.1             10         0060.5cf4.a955  ARPA   Ethernet0

```

Вернемся к проблеме неудачной инкапсуляции. Более подробные сведения о проблеме можно получить с помощью следующей команды отладки:

```

Router4#debug arp
ARP packet debugging is on

```

```

Router4#ping 100.0.0.5

```

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.0.0.5, timeout is 2 seconds:

Jan 20 17:19:43.843: IP ARP: creating incomplete entry for IP address: 100.0.0.5
interface Ethernet0
Jan 20 17:19:43.847: IP ARP: sent req src 100.0.0.4 0000.0c5d.7a0d,
dst 100.0.0.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:45.843: IP ARP: sent req src 100.0.0.4 0000.0c5d.7a0d,
dst 100.0.0.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:47.843: IP ARP: sent req src 100.0.0.4 0000.0c5d.7a0d,
dst 100.0.0.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:49.843: IP ARP: sent req src 100.0.0.4 0000.0c5d.7a0d,
dst 100.0.0.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:51.843: IP ARP: sent req src 100.0.0.4 0000.0c5d.7a0d,
dst 100.0.0.5 0000.0000.0000 Ethernet0.
Success rate is 0 percent (0/5)

```

В представленном выше результате выполнения команды показано, что маршрутизатор 4 транслирует пакеты, пересылая их на широковещательный Ethernet-адрес FFFF.FFFF.FFFF. В данном случае 0000.0000.0000 означает, что маршрутизатор 4 ищет MAC-адрес целевого устройства 100.0.0.5. Поскольку в этом примере он не знает MAC-адреса во время запроса ARP, он отправляет широковещательные кадры с интерфейса Ethernet 0 с 0000.0000.0000 в качестве заполнителя и спрашивает, какой MAC-адрес соответствует 100.0.0.5. Если маршрутизатор не получает ответа, то соответствующий адрес в результате выполнения команды show arp помечается как неполный:

```
Router4#show arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 100.0.0.4 - 0000.0c5d.7a0d ARPA Ethernet0
Internet 100.0.0.5 0 Incomplete ARPA
Internet 100.0.0.1 2 0060.5cf4.a955 ARPA Ethernet0
```

По прошествии определенного периода времени сведения о неполноте удаляются из ARP-таблицы. Пока соответствующий MAC-адрес отсутствует в ARP-таблице, выполнение команды ping будет заканчиваться неудачей в результате "неудачной инкапсуляции".

Задержка

По умолчанию, если вы не получаете ответ от удаленного конца в течение двух секунд, сброс эхо-запроса:

```
Router1#ping 12.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

В сетях с низкой скоростью передачи данных или с большой задержкой двух секунд времени ожидания может оказаться недостаточным. Это значение по умолчанию можно изменить с помощью расширенной проверки связи:

```
Router1#ping
Protocol [ip]:
Target IP address: 12.0.0.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]: 30
Extended commands [n]:
Sweep range of sizes [n]:

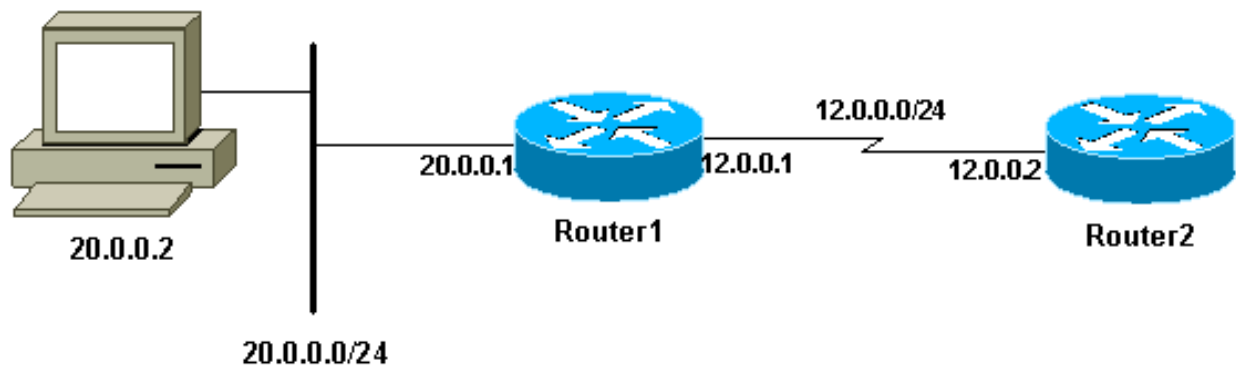
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 30 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1458/2390/6066 ms
```

В приведенном примере увеличение времени ожидания позволило успешно выполнить проверку связи.

Примечание: Среднее время приема-передачи составляет более двух секунд.

Исправьте исходный адрес

Вот пример типичной ситуации:



Добавление интерфейса LAN на маршрутизатор 1:

```
Router1(config)#interface e0
Router1(config-if)#ip address
Router1(config-if)#ip address 20.0.0.1 255.255.255.0
```

Со станции сети LAN можно выполнить эхо-тест маршрутизатора 1. С маршрутизатора 1 можно выполнить эхо-тест маршрутизатора 2. Со станции сети LAN невозможно выполнить эхо-тест маршрутизатора 2.

Можно посылать пакеты проверки связи с Router1 на Router2, потому что по умолчанию IP-адрес исходящего интерфейса используется в качестве адреса источника в пакете ICMP. Маршрутизатор 2 не располагает сведениями об этой новой LAN. Если маршрутизатор должен ответить на пакет приходящий из этой сети, то он не знает, как обрабатывать этот пакет.

```
Router1#debug ip packet
IP packet debugging is on
```

% Warning: Использование команды debug ip packet на производственном маршрутизаторе может привести к большой нагрузке процессора. Это может привести к серьезному падению производительности или выходу сети из строя. [Рекомендуется тщательно ознакомиться с разделом "Использование команды Debug перед использованием команд отладки."](#)

```
Router1#ping 12.0.0.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/7/9 ms
Router1#
```

```
Jan 20 16:35:54.227: IP: s=12.0.0.1 (local), d=12.0.0.2 (Serial0), len 100, sending
Jan 20 16:35:54.259: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 100, rcvd 3
```

Выходные данные из примера выше работают, так как адрес источника посылаемого нами пакета =12.0.0.1. Если необходимо смоделировать пакет, пришедший из локальной сети, нужно использовать расширенную команду ping:

```
Router1#ping
Protocol [ip]:
Target IP address: 12.0.0.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 20.0.0.1
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 12.0.0.2, timeout is 2 seconds:
```

```
Jan 20 16:40:18.303: IP: s=20.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
sending.
Jan 20 16:40:20.303: IP: s=20.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
sending.
Jan 20 16:40:22.303: IP: s=20.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
sending.
Jan 20 16:40:24.303: IP: s=20.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
sending
Jan 20 16:40:26.303: IP: s=20.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
sending.
Success rate is 0 percent (0/5)
```

В данное время исходный адрес - 20.0.0.1, и он не работает! Мы передаем пакеты, но ничего не получаем. Для устранения этой проблемы необходимо добавить маршрут к 20.0.0.0 в маршрутизаторе 2.

Основное правило состоит в том, что устройству, получившему запрос о проверке доступности, должен быть известен способ отправки ответа источнику этого запроса.

Высокое отбрасывание входящей очереди

Когда пакет входит на маршрутизатор, маршрутизатор пытается переслать его на уровне прерывания. Если соответствие не может быть найдено в соответствующей таблице кэш-памяти, пакет помещен в очередь во входной очереди входящего интерфейса, который будет обработан. Некоторые пакеты всегда обрабатываются, но с соответствующей конфигурацией и в стабильных сетях, скорость обработанных пакетов никогда не должна переполнять входную очередь. Если входящая очередь заполнена, пакет отбрасывается.

Хотя интерфейс подключен, и вы не можете пропинговать устройство к высокому отбрасыванию входящей очереди. Можно проверить отбрасывание ввода с командой **show interface**.

```
Router1#show interface Serial0/0/0
```

```
Serial0/0/0 is up, line protocol is up
```

```
MTU 1500 bytes, BW 1984 Kbit, DLY 20000 usec,
  reliability 255/255, txload 69/255, rxload 43/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:02, output 00:00:00, output hang never
Last clearing of "show interface" counters 01:28:49
Input queue: 76/75/5553/0 (size/max/drops/flushes);
  Total output drops: 1760
Queueing strategy: Class-based queueing
Output queue: 29/1000/64/1760 (size/max total/threshold/drops)
  Conversations 7/129/256 (active/max active/max total)
  Reserved Conversations 4/4 (allocated/max allocated)
  Available Bandwidth 1289 kilobits/sec
```

```
!--- Output suppressed
```

Как замечено по выходным данным, Отбрасывание входящей очереди высоко. См. [Устранение проблем Отбрасывания входящей очереди и Удалений из очереди вывода](#) для устранения проблем отбрасывания очереди ввода/вывода.

Команда трассировки

Команда `traceroute` используется для отыскания маршрутов, которые будут использоваться при передаче пакетов к сетевому узлу назначения. Устройство (например, маршрутизатор или PC) посылает последовательность диаграмм протокола дейтаграмм пользователя (UDP) на недопустимый адрес порта на удаленном сайте.

Отправлены три дейтаграммы, каждая со значением поля TTL равным одному. Значение времени существования равное единице означает, что дейтаграмма перестанет существовать после достижения первого маршрутизатора на маршруте, а затем этот маршрутизатор отправит ICMP-сообщение о превышении времени существования, указывающее на истечение времени существования.

Теперь отправляются другие три UDP-сообщения, каждое со значением TTL равным 2, что заставляет второй маршрутизатор вернуть ICMP TЕМ. Этот процесс продолжается до тех пор, пока пакеты не достигают другого пункта назначения. Так как эти дейтаграммы пытаются получить доступ к неверному порту на узле назначения, то этот узел возвращает ICMP-сообщения о недоступном порте. Это событие сигнализирует о необходимости завершить выполнение программы Traceroute.

После этого необходимо записать источник каждого ICMP-сообщения о превышении времени существования для обеспечения трассировки пути, по которому пакет попадает к адресату. [Для получения сведений обо всех параметрах данной команды см. раздел Трассировка \(привилегированный режим\).](#)

```
Router1#traceroute 34.0.0.4
```

```
Type escape sequence to abort.  
Tracing the route to 34.0.0.4
```

```
 1 12.0.0.2 4 msec 4 msec 4 msec  
 2 23.0.0.3 20 msec 16 msec 16 msec  
 3 34.0.0.4 16 msec * 16 msec
```

```
Jan 20 16:42:48.611: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 28,  
  sending  
Jan 20 16:42:48.615:      UDP src=39911, dst=33434  
Jan 20 16:42:48.635: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 56,  
  rcvd 3  
Jan 20 16:42:48.639:      ICMP type=11, code=0  
!--- ICMP Time Exceeded Message from Router2. Jan 20 16:42:48.643: IP: s=12.0.0.1 (local),  
d=34.0.0.4 (Serial0), len 28, sending Jan 20 16:42:48.647: UDP src=34237, dst=33435 Jan 20  
16:42:48.667: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 56, rcvd 3 Jan 20  
16:42:48.671: ICMP type=11, code=0 Jan 20 16:42:48.675: IP: s=12.0.0.1 (local), d=34.0.0.4  
(Serial0), len 28, sending Jan 20 16:42:48.679: UDP src=33420, dst=33436 Jan 20 16:42:48.699:  
IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 56, rcvd 3 Jan 20 16:42:48.703: ICMP  
type=11, code=0
```

Это первая последовательность пакетов отправляемых с параметром TTL = 1. Первый маршрутизатор (в данном случае маршрутизатор 2 (12.0.0.2)) игнорирует пакет и посылает назад отправителю (12.0.0.1) ICMP-сообщение типа 11. Это соответствует сообщению о превышении времени.

```

Jan 20 16:42:48.707: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 28,
  sending
Jan 20 16:42:48.711:      UDP src=35734, dst=33437
Jan 20 16:42:48.743: IP: s=23.0.0.3 (Serial0), d=12.0.0.1 (Serial0), len 56,
  rcvd 3
Jan 20 16:42:48.747:      ICMP type=11, code=0
!--- ICMP Time Exceeded Message from Router3. Jan 20 16:42:48.751: IP: s=12.0.0.1 (local),
d=34.0.0.4 (Serial0), len 28, sending Jan 20 16:42:48.755: UDP src=36753, dst=33438 Jan 20
16:42:48.787: IP: s=23.0.0.3 (Serial0), d=12.0.0.1 (Serial0), len 56, rcvd 3 Jan 20
16:42:48.791: ICMP type=11, code=0 Jan 20 16:42:48.795: IP: s=12.0.0.1 (local), d=34.0.0.4
(Serial0), len 28, sending Jan 20 16:42:48.799: UDP src=36561, dst=33439 Jan 20 16:42:48.827:
IP: s=23.0.0.3 (Serial0), d=12.0.0.1 (Serial0), len 56, rcvd 3 Jan 20 16:42:48.831: ICMP
type=11, code=0

```

Тот же процесс происходит и для Router3 (23.0.0.3) с TTL=2:

```

Jan 20 16:42:48.839: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 28,
  sending
Jan 20 16:42:48.843:      UDP src=34327, dst=33440
Jan 20 16:42:48.887: IP: s=34.0.0.4 (Serial0), d=12.0.0.1 (Serial0), len 56,
  rcvd 3
Jan 20 16:42:48.891:      ICMP type=3, code=3
!--- Port Unreachable message from Router4. Jan 20 16:42:48.895: IP: s=12.0.0.1 (local),
d=34.0.0.4 (Serial0), len 28, sending Jan 20 16:42:48.899: UDP src=37534, dst=33441 Jan 20
16:42:51.895: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 28, sending Jan 20 16:42:51.899:
UDP src=37181, dst=33442 Jan 20 16:42:51.943: IP: s=34.0.0.4 (Serial0), d=12.0.0.1 (Serial0),
len 56, rcvd 3 Jan 20 16:42:51.947: ICMP type=3, code=3

```

Маршрутизатор 4 можно достичь с параметром TTL = 3. На этот раз, поскольку порт недоступен, Router4 отправляет сообщение ICMP обратно на Router1 с типом=3, сообщением недоступности места назначения и кодом=3, означающим что порт недостижим.

В нижеприведенной таблице содержатся символы, которые могут отображаться в результате выполнения команды `tracert`.

IP-трассировка текстовых символов

Символ	Описание
pp мс	Для каждого узла, круговая задержка в миллисекундах для заданного номера зондов
*	Время зонда вышло
О	Административно запрещено (например, список доступа)
Вопрос.	Гашение источников (направление перегружено)
Я	Пользователь прервал тест
U	Порт недоступен
H	Узел недоступен
N	Сеть unreachable
P	Протокол недоступен
T	Таймаут
?	Неизвестный тип пакета

С помощью команд `ping` и `tracert` получаем время приема-передачи (RTT). Это время, необходимое для пересылки эхо-пакета и получения ответа. Полезно иметь приблизительное представление о задержке в канале. Однако получаемые значения недостаточно точны для оценки пропускной способности.

Если адресом назначения является адрес самого маршрутизатора, то для этих пакетов должно быть произведено перенаправление. Процессор должен обработать данные из такого пакета и отправить ответ. Это не основная цель маршрутизатора. По определению, маршрутизатор создан для маршрутизации пакетов. Ответ на запрос эхо-теста предлагается в качестве службы негарантированной доставки.

Для примера приведены результаты эхо-теста соединения между маршрутизатором 1 и маршрутизатором 2:

```
Router1#ping 12.0.0.2
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

Время приема-передачи (RTT) равно примерно четырем миллисекундам. После разрешения некоторых ресурсозатратных функций на маршрутизаторе 2 попытайтесь отправить команду `ping` с маршрутизатора 2 на маршрутизатор 1.

```
Router1#ping 12.0.0.2
```

```
Type escape sequence to abort.
```

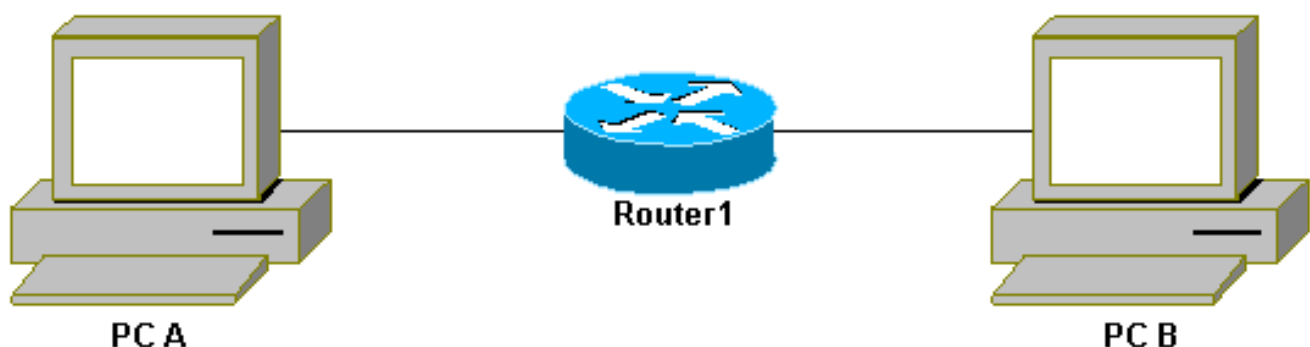
```
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/25/28 ms
```

Время приема-передачи (RTT) значительно выросло. Маршрутизатор 2 очень занят, и ответ на запрос проверки связи не является его первостепенной задачей.

Для проверки производительности маршрутизатора можно использовать проходящий через него трафик:



После чего маршрутизатором производится быстрое перенаправление пакетов и их обработка с наивысшим приоритетом. Для иллюстрации вернитесь к основной сети:



Произведем опрос маршрутизатора 3 с маршрутизатора 1 с помощью команды ping:

```
Router1#ping 23.0.0.3
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 23.0.0.3, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
```

Трафик проходит через маршрутизатор 2 и быстро коммутируется.

Теперь разрешим ресурсозатратные функции на маршрутизаторе 2:

```
Router1#ping 23.0.0.3
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 23.0.0.3, timeout is 2 seconds:
```

```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
```

Никаких различий почти не существует. Это связано с тем, что теперь на маршрутизаторе 2 пакеты обрабатываются на уровне прерывания.

Использование команды Debug

Прежде чем применять команды отладки, ознакомьтесь с разделом "Важные сведения о командах отладки".

Использованные до сих пор команды debug давали нам понимание того, что происходит при использовании команды ping или traceroute. Они также могут использоваться для устранения неполадок. Однако в реальных условиях отладка должна производиться с осторожностью. Если процессор не достаточно производительный или существует много перенаправляемых пакетов, то это может легко привести к падению производительности сетевого устройства. Существует пара методов для минимизации влияния выполнения команды debug на маршрутизатор. Один из способов - использование списков доступа для ограничения трафика, который требуется контролировать. Например:

```
Router4#debug ip packet ?
```

```
<1-199>      Access list
```

```
<1300-2699>  Access list (expanded range)
```

```
detail       Print more debugging detail
```

```
Router4#configure terminal
```

```
Router4(config)#access-list 150 permit ip host 12.0.0.1 host 34.0.0.4
```

```
Router4(config)#^Z
```

```
Router4#debug ip packet 150
```

```
IP packet debugging is on for access list 150
```

```
Router4#show debug
Generic IP:
  IP packet debugging is on for access list 150
```

```
Router4#show access-list
Extended IP access list 150
  permit ip host 12.0.0.1 host 34.0.0.4 (5 matches)
```

При этой конфигурации маршрутизатор 4 печатает сообщения отладки, соответствующие только списку контроля доступа 150. Команда ping, поступающая от маршрутизатора 1, приводит к отображению следующего сообщения:

```
Router4#debug ip packet ?
<1-199>      Access list
<1300-2699>  Access list (expanded range)
detail       Print more debugging detail
```

```
Router4#configure terminal
Router4(config)#access-list 150 permit ip host 12.0.0.1 host 34.0.0.4
Router4(config)#^Z
```

```
Router4#debug ip packet 150
IP packet debugging is on for access list 150
```

```
Router4#show debug
Generic IP:
  IP packet debugging is on for access list 150
```

```
Router4#show access-list
Extended IP access list 150
  permit ip host 12.0.0.1 host 34.0.0.4 (5 matches)
```

Мы больше не видим ответа от маршрутизатора Router4, так как эти пакеты не соответствуют списку управления доступом. Для их просмотра нужно добавить следующее:

```
Router4(config)#access-list 150 permit ip host 12.0.0.1 host 34.0.0.4
Router4(config)#access-list 150 permit ip host 34.0.0.4 host 12.0.0.1
```

После этого отобразятся такие результаты:

```
Router4(config)#access-list 150 permit ip host 12.0.0.1 host 34.0.0.4
Router4(config)#access-list 150 permit ip host 34.0.0.4 host 12.0.0.1
```

Другой способ свести к минимуму воздействие команды debug - помещать отладочные сообщения в буфер и выводить их с помощью команды show log после выключения отладки:

```
Router4#configure terminal
Router4(config)#no logging console
Router4(config)#logging buffered 5000
Router4(config)#^Z
```

```
Router4#debug ip packet
IP packet debugging is on
Router4#ping 12.0.0.1
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/36/37 ms
```

```
Router4#undebug all
All possible debugging has been turned off
```

```
Router4#show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
```

```
Console logging: disabled
Monitor logging: level debugging, 0 messages logged
Buffer logging: level debugging, 61 messages logged
Trap logging: level informational, 59 message lines logged
```

Log Buffer (5000 bytes):

```
Jan 20 16:55:46.587: IP: s=34.0.0.4 (local), d=12.0.0.1 (Serial0), len 100,
  sending
Jan 20 16:55:46.679: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
  rcvd 3
```

Как показано выше, команды ping и traceroute являются очень полезными при поиске и устранении проблем доступа к сети. Кроме того, они очень просты в применении. Так как эти две команды широко используются сетевыми инженерами, то понимание принципов их функционирования является очень важным при поиске и устранении неисправностей подключения к сети.

[Дополнительные сведения](#)

- [Используйте команды Extended ping и Extended traceroute](#)
- [Техническая поддержка - Cisco Systems](#)