Общие сведения о командах Ping и Traceroute

Содержание

Общие сведения

В данном документе рассматривается использование команд **ping** и **traceroute**. Приведенный в данном документе более подробный обзор результатов выполнения этих команд, был получен с помощью некоторых команд **debug**.

Примечание. Включение возможности использования команд **debug** на производственном маршрутизаторе может привести к серьезным проблемам. Перед использованием команд **debug** рекомендуется тщательно ознакомиться с разделом Использование команды Debug.

Предварительные условия

Требования

Для данного документа нет особых требований.

Используемые компоненты

Сведения, содержащиеся в данном документе, не ограничены определенными версиями программного и аппаратного обеспечения.

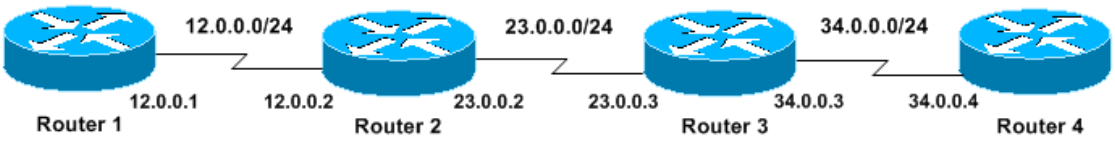
Сведения для данного документа были получены на тестовом оборудовании в специально созданных лабораторных условиях. Все устройства, описанные в данном документе, начинали работу с очищенной (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Условные обозначения

Дополнительные сведения об условных обозначениях см. в документе Технические рекомендации Cisco. Условные обозначения.

Базовые 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 хост активным или неактивным;
- Задержки приема-передачи при взаимодействии с хостом;
- Потери пакетов.

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

- эхо-запрос достигает места назначения;
- опрашиваемое устройство может отправить эхо-ответ обратно источнику в рамках заданного времени, называемого временем ожидания (тайм-аутом). Значение тайм-аута по умолчанию для маршрутизаторов Cisco равно двум секундам.

Для получения информации по всем параметрам данной команды см. раздел "Ping" главы Команды устранения неполадок.

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

Предупреждение. Использование команды **debug ip packet detail** на производственном маршрутизаторе может привести к большой нагрузке на процессор. Это может привести к серьезному падению производительности или выходу сети из строя. Перед использованием команд **debug** рекомендуется внимательно ознакомиться с разделом Использование команды Debug.

```
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

!--- Это ICMP пакет, отправленный из 12.0.0.1 в 12.0.0.2.
!--- ICMP тип=8 соответствует эхо-сообщению.
```

```
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

!--- Это ответ, полученный от 12.0.0.2.
!--- ICMP тип=0 соответствует эхо-ответу.
!--- По умолчанию число повторов равно 5, поэтому будет пять
!--- эхо-запросов и пять эхо-ответов.
```

В приведенной ниже таблице перечислены возможные значения типа ICMP-протокола.

Тип ICMP-протокола	**Символьная константа**
0	Эхо-ответ
3	цель назначения недостижима код 0 = сеть недостижима 1 = хост недостижим 2 = протокол недостижим 3 = порт недостижим 4 = необходима фрагментация и установка DF-бита 5 = исходный маршрут недоступен
4	отключение источника сообщения при перегрузке с предварительным возвратом сообщения
5	перенаправление код 0 = перенаправление дейтаграмм для сети 1 = перенаправление дейтаграмм для хоста 2 = перенаправление дейтаграмм для типа службы или сети 3 = перенаправление дейтаграмм для типа службы и хоста
6	альтернативный адрес
8	эхо
9	объявление маршрутизатора

10	запрос маршрутизатора
11	истечение времени ожидания код 0 = истекло время существования пакета во время его передачи 1 = превышено время сборки фрагментов
12	проблема параметра
13	запрос временной метки
14	ответ на запрос о временной метке
15	информационный запрос
16	ответ на информационный запрос
17	запрос маски
18	ответ на запрос маски
31	ошибка преобразования
32	мобильное перенаправление

В нижеприведенной таблице содержатся сведения о символах, которые могут содержаться в результатах выполнения команды ping:

Символ	**Описание**
!	Каждый восклицательный знак указывает на получение ответа.
.	Каждая точка указывает на тайм-аут сетевого сервера во время ожидания ответа.
U	Получена ошибка о невозможности достижения цели протокольным блоком данных.
Q	Отключение источника сообщения при перегрузке с предварительным возвратом сообщения (цель назначения перегружена).
M	Фрагментация не может быть выполнена.

?	Неизвестный тип пакета.
&	Превышено время существования пакета.

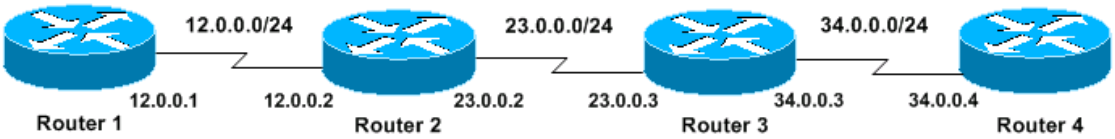
Причины неудачного выполнения команды ping

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

Проблема маршрутизации

Далее приведен пример неудачного выполнения команды ping, определения проблемы и мер, необходимых для ее устранения.

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
```

 **Предупреждение.** Использование команды **debug ip packet** на производственном маршрутизаторе может привести к большой нагрузке на процессор. Это может привести к серьезному падению производительности или выходу сети из строя. Перед использованием команд **debug** рекомендуется внимательно ознакомиться с разделом Использование команды Debug.

```
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 невозможности маршрутизации.

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 маршрутизатор 2, но маршрутизатор 2 не знает, как получить доступ к адресу 34.0.0.4. Маршрутизатор 2 отправляет Маршрутизатору 1 сообщение "unreachable ICMP" ("недоступный ICMP-протокол").

Теперь разрешите использование протокола маршрутизации информации (RIP-протокол) на маршрутизаторе 2 и маршрутизаторе 3.

```
Router2#
router rip
network 12.0.0.0
network 23.0.0.0
Router3#
router rip
network 23.0.0.0
network 34.0.0.0
```

Получаем:

```
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 улучшает ситуацию. Маршрутизатор 1 отправляет пакеты на маршрутизатор 4, но не получает никакого ответа от него.

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 ICMP-пакеты и пытается отправить ответ на адрес 12.0.0.1, но так как у него нет маршрута в эту сеть, эта попытка 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
```

Недоступность интерфейса

Эта ситуация возникает в случаях, когда интерфейс перестает работать. 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:
U.U.U
Success rate is 0 percent (0/5)
```

Поскольку маршрутизация исправна, то 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
```

Из приведенных выше данных видно, что источник проблемы находится между маршрутизатором 2 и маршрутизатором 3. Одной из возможных причин может являться то, что последовательный интерфейс на маршрутизаторе 3 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 только трафику telnet поступать на 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
```

Сведения об использовании списков контроля доступа с командами **debug** см. в разделе Использование команды Debug.

При попытке проверить доступность маршрутизатора 4 с помощью команды ping 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,
```

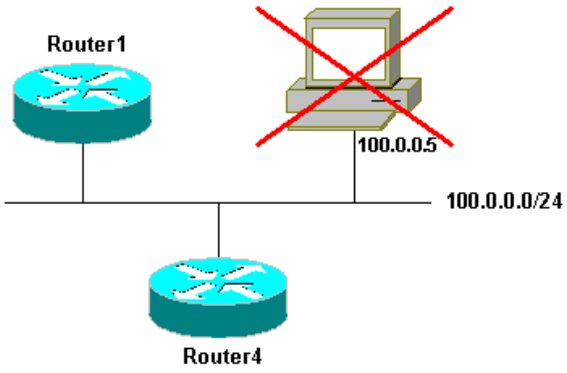
```
rcvd 3
Jan 20 16:34:49.291: ICMP: dst (12.0.0.1) administratively prohibited unreachable
rcv from 34.0.0.4
Jan 20 16:34:49.295: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
sending
Jan 20 16:34:51.295: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
sending
Jan 20 16:34:51.367: IP: 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 **access-list** всегда существует неявное условие "deny all" ("запретить всё"). Это означает, что ICMP-пакеты, поступающие на интерфейс Serial 0 маршрутизатора 4, отклоняются, а маршрутизатор 4, как показано в результате выполнения команды **debug**, отправляет источнику исходного пакета сообщение — "administratively prohibited unreachable" ("доступ запрещен административно"). Решением проблемы является добавление в команду **access-list** следующей строки:

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

Проблема с протоколом разрешения адресов (ARP-протокол)

В данном подразделе приведен пример подключения через протокол 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 отправлять 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.

В основном, ARP — это протокол, используемый для сопоставления адреса второго уровня (MAC-адрес) с адресом третьего уровня (IP-адрес). 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 об этой проблеме можно получить с помощью команды **debug**:

```
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 выполнения ARP-запроса, то он отсылает широковещательные кадры с интерфейса Ethernet 0 с адресом 0000.0000.000 в качестве шаблона и запрашивает, какой MAC-адрес соответствует IP-адресу 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 ответ от удаленного оконечного сетевого устройства не получен в течение двух секунд, то выполнение команды ping заканчивается неудачей:

```
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 выполнения расширенной команды ping:

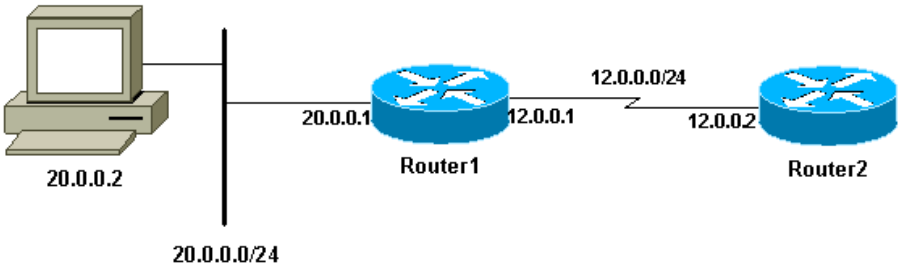
```
Router1#ping
Protocol [ip]:
Target IP address:  12.0.0.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]: 30
Extended commands [n]:
Sweep range of sizes [n]:

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 вышеприведенном примере увеличение времени ожидания привело к успешному выполнению команды ping.

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

Исправление адреса-источника

В данном подразделе приведен пример типичной ситуации:



На маршрутизаторе 1 добавлен интерфейс LAN:

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

Из узла локальной сети (LAN) можно выполнить команду ping для маршрутизатора 1. Команду ping можно направить с маршрутизатора 1 на маршрутизатор 2. Но к маршрутизатору 2 невозможно применить команду ping из узла локальной сети (LAN).

Можно посылать пакеты проверки связи с маршрутизатора 1 на маршрутизатор 2, так как по умолчанию IP-адрес исходящего интерфейса используется в качестве адреса источника в ICMP-пакете. Маршрутизатор 2 не располагает сведениями об этой новой локальной сети (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
```

Предупреждение. Использование команды **debug ip packet** на производственном маршрутизаторе может привести к большой нагрузке на процессор. Это может привести к серьезному падению производительности или выходу сети из строя. Перед использованием команд **debug** рекомендуется внимательно ознакомиться с разделом Использование команды Debug.

```
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 вышеприведенного примера работают, потому что адрес источника отправляемого пакета равен s = 12.0.0.1. Если необходимо смоделировать пакет, поступающий из локальной сети, то следует 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)
```

Теперь исходным адресом является IP-адрес 20.0.0.1, который не функционирует! Пакеты могут быть переданы, но ответа на них не поступает. Для устранения этой проблемы необходимо добавить маршрут к IP-адресу 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.

Команда traceroute

Команда **traceroute** используется для отыскания маршрутов, которые будут использоваться при передаче пакетов к сетевому узлу назначения. Устройство (например, маршрутизатор или персональный компьютер) отправляет последовательность UDP-дейтаграмм на недопустимый адрес порта на удаленном хосте.

Отправляются три дейтаграммы со значением поля TTL (время существования) равным единице. 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 со значением времени существования равным двум, что является причиной, по которой второй маршрутизатор возвращает ICMP-сообщение о превышении времени существования. 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-сообщение об истечении времени от маршрутизатора 2.

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 существования.

```
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-сообщение об истечении времени от маршрутизатора 3.

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 самый процесс происходит и для маршрутизатора 3 (23.0.0.3) с параметром 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

!--- Сообщение о недоступности порта от маршрутизатора 4.

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, маршрутизатор 4 оправляет обратно на маршрутизатор 1 ICMP-сообщение с типом равным 3, сообщение о недоступности места назначения и код равный 3, означающий, что порт недостижим.

В нижеприведенной таблице содержатся символы, которые могут отображаться в результате выполнения команды **traceroute**.

екстовые символы IP-трассировки

Символ	Описание
nn msec	Время приема-передачи в миллисекундах для заданного числа тестовых сообщений при проверке каждого узла сети
*	Время жизни тестового сообщения
A	Административно запрещено (например, список контроля доступа)
Q	Отключение источника сообщения при перегрузке с предварительным возвратом сообщения (цель назначения перегружена)
I	Пользовательская проверка на прерывание
U	Порт недостижим
H	Хост недостижим
N	Сеть недостижима
P	Протокол недостижим
T	Тайм-аут

?	Неизвестный тип пакета

Пропускная способность

С помощью команд **ping** и **traceroute** можно получить время приема-передачи (RTT). Это время, необходимое для отправки эхо-пакета и получения ответа. Полезно иметь приблизительное представление о задержке в канале. Однако получаемые значения недостаточны 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 спроектирован для маршрутизации пакетов. Ответ на запрос эхо-теста предлагается в качестве службы негарантированной доставки (best-effort – по возможности).

В качестве примера приведем результат выполнения команды ping между маршрутизатором 1 и маршрутизатором 2:

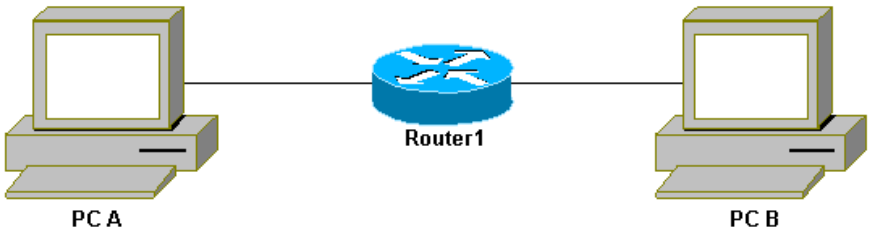
```
Router1#ping 12.0.0.2

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, а ответ на запрос проверки связи не является его первостепенной задачей.

Лучшим способом проверки пропускной способности маршрутизатора является использование трафика, проходящего **через** него:



После чего маршрутизатором производится быстрое перенаправление пакетов и их обработка с наивысшим приоритетом. Для иллюстрации этого вернемся 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 **debug**, ознакомьтесь с разделом Важные сведения о командах Debug.

Использованные до сих пор команды **debug** давали нам понимание того, что происходит при использовании команды **ping** или **traceroute**. Они также могут оказаться полезными при поиске и устранении 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 контроля доступа для ограничения трафика, который требуется контролировать. Ниже представлен пример этого:

```
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#
Jan 20 16:51:16.911: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
rcvd 3
Jan 20 16:51:17.003: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
rcvd 3
Jan 20 16:51:17.095: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
rcvd 3
Jan 20 16:51:17.187: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
rcvd 3
Jan 20 16:51:17.279: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
rcvd 3
```

Пакеты, не соответствующие списку контроля доступа, маршрутизатором 4 не отображаются. Для их просмотра необходимо добавить следующие команды:

```
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 следующие результаты:

```
Jan 20 16:53:16.527: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
rcvd 3
Jan 20 16:53:16.531: IP: s=34.0.0.4 (local), d=12.0.0.1 (Serial0), len 100,
sending
Jan 20 16:53:16.627: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
rcvd 3
Jan 20 16:53:16.635: IP: s=34.0.0.4 (local), d=12.0.0.1 (Serial0), len 100,
sending
Jan 20 16:53:16.727: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
rcvd 3
Jan 20 16:53:16.731: IP: s=34.0.0.4 (local), d=12.0.0.1 (Serial0), len 100,
sending
Jan 20 16:53:16.823: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
rcvd 3
Jan 20 16:53:16.827: IP: s=34.0.0.4 (local), d=12.0.0.1 (Serial0), len 100,
sending
Jan 20 16:53:16.919: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
rcvd 3
Jan 20 16:53:16.923: IP: s=34.0.0.4 (local), d=12.0.0.1 (Serial0), len 100,
sending
```

Другим способом минимизации воздействия команды **debug** является помещение отладочных сообщений в буфер и их вывод после выключения отладки с помощью команды **show log**:

```
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

- **Использование команд Extended ping и Extended traceroute**
- **Техническая поддержка — Cisco Systems**

Дата генерации PDF файла: Jan 05, 2010

http://www.cisco.com/support/RU/customer/content/9/92090/ping_traceroute.shtml