Entender os comandos ping e traceroute

Contents

Introduction

Este documento descreve o uso dos comandos **ping** e **traceroute** nos roteadores Cisco.

Prerequisites

Requirements

Não existem requisitos específicos para este documento.

Componentes Utilizados

Este documento não se restringe a versões de software e hardware específicas.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

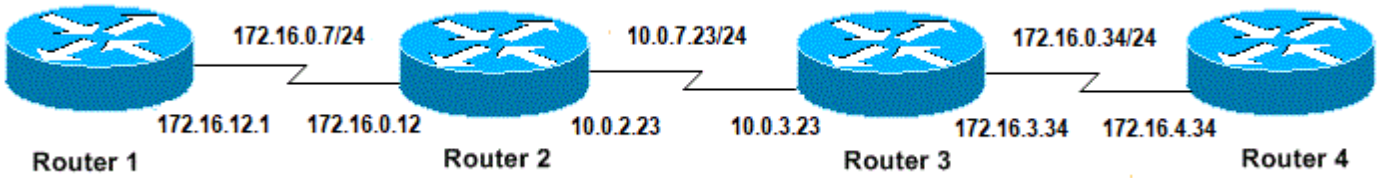
Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Informações de Apoio

> **Note**: Qualquer comando **debug** usado em um roteador de produção pode causar problemas sérios. Leia a seção Use the Debug Command antes de executar os comandos **debug** .

Neste documento, esta configuração básica é usada para exemplos neste artigo:



Configuração básica de IPs e roteadores

O comando ping

O comando **ping** é um método muito comum usado para solucionar problemas de acessibilidade de dispositivos. Usa uma série de mensagens de eco do protocolo Protocolo de controle de mensagens de Internet (ICMP) (ICMP) para determinar:

- Se um host remoto está ativo ou inativo.
- O atraso de ida e volta usado para se comunicar com o host.
- Perda de pacotes.

O **comando ping primeiro envia um pacote de requisição de eco a um endereço e depois aguarda uma resposta.** O ping será bem-sucedido somente se:

- a solicitação de eco chega ao destino e
- o destino é capaz de devolver uma resposta de eco para a origem, em um período predeterminado, denominado intervalo. O valor padrão desse timeout é dois segundos em Cisco routers.

O valor TTL de um pacote de ping não pode ser mudado.

Este próximo exemplo de código mostra o comando **ping** depois que o comando **debug ip packet detail** é habilitado.

> **aviso:** Quando o comando **debug ip packet detail** é usado em um roteador de produção, ele pode causar alta utilização da CPU. Isso pode resultar em uma grave degradação do desempenho ou uma interrupção da rede.

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

Router1#ping 172.16.0.12
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.12, 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=172.16.12.1 (local), d=172.16.0.12 (Serial0), len 100,
  sending
Jan 20 15:54:47.491: ICMP type=8, code=0

!--- This is the ICMP packet 172.16.12.1 sent to 172.16.0.12.
!--- ICMP type=8 corresponds to the echo message. Jan 20 15:54:47.523: IP: s=172.16.0.12
(Serial0), d=172.16.12.1 (Serial0), len 100, rcvd 3 Jan 20 15:54:47.527: ICMP type=0, code=0

!--- This is the answer we get from 172.16.0.12. !--- 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.
```

Valores possíveis do tipo ICMP

Tipo de ICMP	Literal
0	resposta de eco
3	código de destino inalcançável 0 = rede inalcançável 1 = host inalcançável 2 = protocolo inalcan 3 = porta inalcançável 4 = fragmentação necessária e DF definido 5 = falha na rota de origem
4	source-quench
5	reoriente o código 0 = reorienta datagramas para a rede 1 = reorienta datagramas para o host 2 reorienta datagramas para o tipo de serviço e a rede 3 = reorienta datagramas para o tipo de se e o host
6	endereço alternativo
8	eco
9	router-advertisement
10	router-solicitation
11	código de tempo excedido 0 = Time to Live excedido no trânsito 1 = tempo de remontagem de fragmento excedido
12	problema de parâmetro
13	timestamp-request
14	timestamp-reply
15	requisição de informações
16	resposta de informação
17	mask-request
18	mask-reply
31	conversion-error
32	mobile-redirect

Possíveis caracteres de saída do recurso de ping

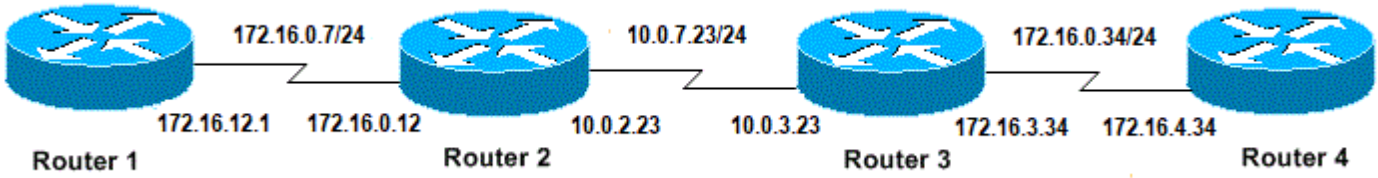
Caractere	Descrição
!	Cada ponto de exclamação indica o recibo de uma resposta.
.	Cada período indica que o servidor de rede atingiu o tempo limite enquanto aguardav uma resposta.
U	Um erro de destino inalcançável PDU foi recebido.
P	A fonte extingue (destino demasiado ocupado).
M	Não foi possível fragmentar.
?	Tipo de pacote desconhecido.
e	Duração de pacote excedida.

Não é possível fazer ping

Se você não conseguir fazer **ping** com êxito em um endereço IP, considere as causas listadas nesta seção.

Problema do roteador

Aqui estão exemplos de tentativas de ping malsucedidas, que podem determinar o problema, e o que fazer para resolver o problema. Este exemplo é mostrado com este diagrama de topologia de rede:



Problemas do roteador

```
Router1#
!
interface Serial0
ip address 172.16.12.1 255.255.255.0
no fair-queue
clockrate 64000
!
```

```
Router2#
!
interface Serial0
ip address 10.0.2.23 255.255.255.0
no fair-queue
clockrate 64000
!
interface Serial1
ip address 172.16.0.12 255.255.255.0
!
```

```
Router3#
!
interface Serial0
ip address 172.16.3.34 255.255.255.0
no fair-queue
!
interface Serial1
ip address 10.0.3.23 255.255.255.0
!
```

```
Router4#
!
interface Serial0
ip address 172.16.4.34 255.255.255.0
no fair-queue
clockrate 64000
!
```

Tente fazer ping de Router4 de Router1:

```
Router1#ping 172.16.4.34
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.34, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Resultados:

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

> **aviso:** Quando o comando **debug ip packet** é usado em um roteador de produção, ele pode causar alta utilização da CPU. Isso pode resultar em uma grave degradação do desempenho ou uma interrupção da rede.

```
Router1#ping 172.16.4.34

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.34, timeout is 2 seconds:

Jan 20 16:00:25.603: IP: s=172.16.12.1 (local), d=172.16.4.34, len 100, unroutable.
Jan 20 16:00:27.599: IP: s=172.16.12.1 (local), d=172.16.4.34, len 100, unroutable.
Jan 20 16:00:29.599: IP: s=172.16.12.1 (local), d=172.16.4.34, len 100, unroutable.
Jan 20 16:00:31.599: IP: s=172.16.12.1 (local), d=172.16.4.34, len 100, unroutable.
Jan 20 16:00:33.599: IP: s=172.16.12.1 (local), d=172.16.4.34, len 100, unroutable.
Success rate is 0 percent (0/5)
```

Como nenhum protocolo de roteamento é executado no Router1, ele não sabe para onde enviar seu pacote e causa uma mensagem "não roteável".

Adicione uma rota estática ao Roteador 1:

```
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
```

Resultados:

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

Router1#ping 172.16.4.34
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.34, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

Jan 20 16:05:30.659: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
  sending
Jan 20 16:05:30.663:     ICMP type=8, code=0
Jan 20 16:05:30.691: IP: s=172.16.0.12 (Serial0), d=172.16.12.1 (Serial0), len 56,
  rcvd 3
Jan 20 16:05:30.695:     ICMP type=3, code=1
Jan 20 16:05:30.699: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
  sending
Jan 20 16:05:30.703:     ICMP type=8, code=0
Jan 20 16:05:32.699: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
```

```
  sending
Jan 20 16:05:32.703:     ICMP type=8, code=0
Jan 20 16:05:32.731: IP: s=172.16.0.12 (Serial0), d=172.16.12.1 (Serial0), len 56,
  rcvd 3
Jan 20 16:05:32.735:     ICMP type=3, code=1
Jan 20 16:05:32.739: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
  sending
Jan 20 16:05:32.743:     ICMP type=8, code=0
```

Examine o que está errado no Roteador2:

```
Router2#debug ip packet detail
IP packet debugging is on (detailed)

Router2#
Jan 20 16:10:41.907: IP: s=172.16.12.1 (Serial1), d=172.16.4.34, len 100, unroutable
Jan 20 16:10:41.911:     ICMP type=8, code=0
Jan 20 16:10:41.915: IP: s=172.16.0.12 (local), d=172.16.12.1 (Serial1), len 56, sending
Jan 20 16:10:41.919:     ICMP type=3, code=1
Jan 20 16:10:41.947: IP: s=172.16.12.1 (Serial1), d=172.16.4.34, len 100, unroutable
Jan 20 16:10:41.951:     ICMP type=8, code=0
Jan 20 16:10:43.943: IP: s=172.16.12.1 (Serial1), d=172.16.4.34, len 100, unroutable
Jan 20 16:10:43.947:     ICMP type=8, code=0
Jan 20 16:10:43.951: IP: s=172.16.0.12 (local), d=172.16.12.1 (Serial1), len 56, sending
Jan 20 16:10:43.955:     ICMP type=3, code=1
Jan 20 16:10:43.983: IP: s=172.16.12.1 (Serial1), d=172.16.4.34, len 100, unroutable
Jan 20 16:10:43.987:     ICMP type=8, code=0
Jan 20 16:10:45.979: IP: s=172.16.12.1 (Serial1), d=172.16.4.34, len 100, unroutable
Jan 20 16:10:45.983:     ICMP type=8, code=0
Jan 20 16:10:45.987: IP: s=172.16.0.12 (local), d=172.16.12.1 (Serial1), len 56, sending
Jan 20 16:10:45.991:     ICMP type=3, code=1
```

O Roteador 1 enviou corretamente seus pacotes ao Roteador 2, mas o Roteador 2 não sabe como acessar o endereço 172.16.4.34. O Roteador 2 retorna uma mensagem "ICMP inalcançável" ao Roteador 1.

Ative o Routing Information Protocol (RIP) nos roteadores 2 e 3:

```
Router2#
router rip
 network 172.16.0.7
 network 10.0.7.23
Router3#
router rip
 network 10.0.7.23
 network 172.16.0.34
```

Resultados:

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

Router1#ping 172.16.4.34

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.34, timeout is 2 seconds:

Jan 20 16:16:13.367: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
 sending.
Jan 20 16:16:15.363: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
 sending.
```

```
Jan 20 16:16:17.363: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
 sending.
Jan 20 16:16:19.363: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
 sending.
Jan 20 16:16:21.363: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
 sending.
Success rate is 0 percent (0/5)
```

O Roteador 1 envia pacotes ao Roteador 4, mas o Roteador 4 não envia uma resposta de volta.

Possível problema no Router4:

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

Router4#
Jan 20 16:18:45.903: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:45.911: IP: s=172.16.4.34 (local), d=172.16.12.1, len 100, unroutable
Jan 20 16:18:47.903: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:47.907: IP: s=172.16.4.34 (local), d=172.16.12.1, len 100, unroutable
Jan 20 16:18:49.903: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:49.907: IP: s=172.16.4.34 (local), d=172.16.12.1, len 100, unroutable
Jan 20 16:18:51.903: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:51.907: IP: s=172.16.4.34 (local), d=172.16.12.1, len 100, unroutable
Jan 20 16:18:53.903: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:53.907: IP: s=172.16.4.34 (local), d=172.16.12.1, len 100, unroutable
```

O Roteador 4 recebe os pacotes ICMP e tenta responder a 172.16.12.1, mas como não tem uma rota para essa rede, ele falha.

Adicione uma rota estática ao Roteador4:

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

Agora ambos os lados podem acessar um ao outro:

```
Router1#ping 172.16.4.34

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.34, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/35/36 ms
```

Interface down

Essa é uma situação em que a interface pára e não funciona mais. Neste próximo exemplo, há uma tentativa de fazer ping do Router1 para o Router4:

```
Router1#ping 172.16.4.34

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.34, timeout is 2 seconds:
```

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

Como o roteamento está correto, faça um troubleshooting passo a passo do problema. Tente fazer ping de Router2:

```
Router1#ping 172.16.0.12

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.12, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

A partir do exemplo anterior, o problema está entre o Roteador2 e o Roteador3. Uma possibilidade é que a interface serial no Roteador3 tenha sido desativada:

```
Router3#show ip interface brief
Serial0   172.16.3.34    YES manual up                    up
Serial1   10.0.3.23    YES manual administratively down   down
```

Isso é simples de corrigir:

```
Router3#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router3(config)#interface serial1
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
```

Comando da lista de acesso

Neste cenário, somente o tráfego telnet tem permissão para entrar no Roteador 4 através da interface Serial0.

```
Router4(config)# access-list 100 permit tcp any any eq telnet
Router4(config)#interface serial0
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 172.16.12.1 host 172.16.4.34
Router1(config)#access-list 100 permit ip host 172.16.4.34 host 172.16.12.1
Router1(config)#end
Router1#debug ip packet 100
IP packet debugging is on
Router1#debug ip icmp
ICMP packet debugging is on
```

Tente fazer ping no Router4:

```
Router1#ping 172.16.4.34

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.34, timeout is 2 seconds:
U.U.U
```

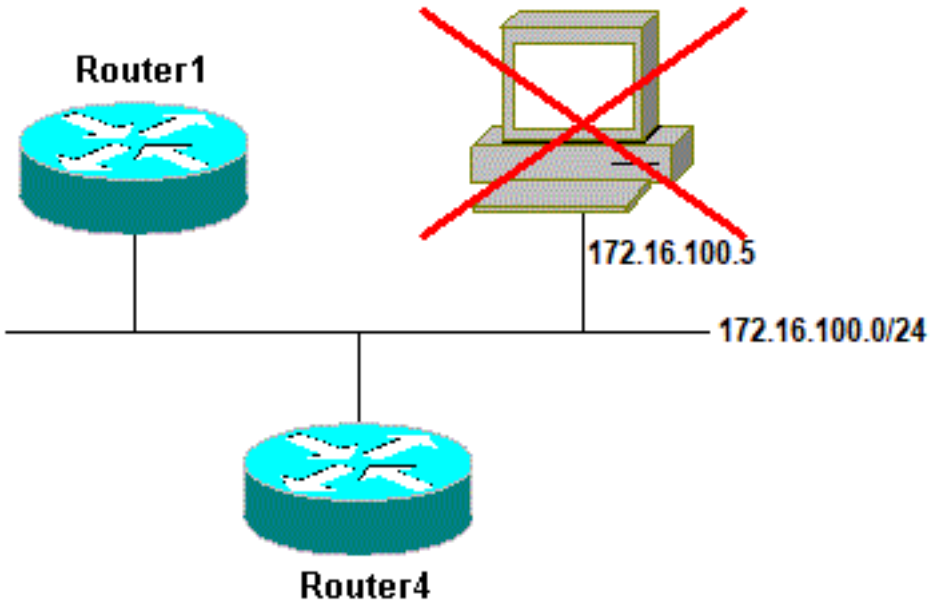
```
Success rate is 0 percent (0/5)

Jan 20 16:34:49.207: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
 sending
Jan 20 16:34:49.287: IP: s=172.16.4.34 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:34:49.291: ICMP: dst (172.16.12.1) administratively prohibited unreachable
 rcv from 172.16.4.34
Jan 20 16:34:49.295: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
 sending
Jan 20 16:34:51.295: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
 sending
Jan 20 16:34:51.367: IP: s=172.16.4.34 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:34:51.371: ICMP: dst (172.16.12.1) administratively prohibited unreachable
 rcv from 172.16.4.34
Jan 20 16:34:51.379: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 100,
 sending
```

No final de um comando **access-list** , há sempre um **deny all** implícito. Isso significa que os pacotes ICMP que entram na interface Serial 0 no Router4 são negados, e o Router 4 envia uma mensagem ICMP "administratively bidden unreachable" para a origem do pacote original, como mostrado na mensagem **debug** . A solução é adicionar esta linha no comando **access-list**:

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

Problema do Address Resolution Protocol (ARP)

Neste cenário, esta é a conexão Ethernet:



Problema de Address Resolution Protocol

```
Router4#ping 172.16.100.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.100.5, timeout is 2 seconds:

Jan 20 17:04:05.167: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
```

```
Jan 20 17:04:05.171: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:07.167: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:07.171: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:09.175: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:09.183: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:11.175: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:11.179: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:13.175: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:13.179: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Success rate is 0 percent (0/5)
Router4#
```

Neste exemplo, o ping não funciona devido à mensagem "encapsulation failed". Isso significa que o roteador sabe em que interface deve enviar o pacote, mas não sabe como fazê-lo. Nesse caso, você precisa entender como o Address Resolution Protocol (ARP) funciona.

O ARP é um protocolo usado para mapear o endereço de Camada 2 (endereço MAC) para um endereço de Camada 3 (endereço IP). Você pode verificar isso com o comando **show arp**:

```
Router4#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.16.100.4               -   0000.0c5d.7a0d  ARPA   Ethernet0
Internet  172.16.100.7              10   0060.5cf4.a955  ARPA   Ethernet0
```

Retorne ao problema "encapsulation failed", mas desta vez habilite o comando **debug arp**:

```
Router4#debug arp
ARP packet debugging is on

Router4#ping 172.16.100.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.100.5, timeout is 2 seconds:

Jan 20 17:19:43.843: IP ARP: creating incomplete entry for IP address: 172.16.100.5
  interface Ethernet0
Jan 20 17:19:43.847: IP ARP: sent req src 172.16.100.4 0000.0c5d.7a0d,
                 dst 172.16.100.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:45.843: IP ARP: sent req src 172.16.100.4 0000.0c5d.7a0d,
                 dst 172.16.100.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:47.843: IP ARP: sent req src 172.16.100.4 0000.0c5d.7a0d,
                 dst 172.16.100.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:49.843: IP ARP: sent req src 172.16.100.4 0000.0c5d.7a0d,
                 dst 172.16.100.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:51.843: IP ARP: sent req src 172.16.100.4 0000.0c5d.7a0d,
                 dst 172.16.100.5 0000.0000.0000 Ethernet0.
Success rate is 0 percent (0/5)
```

A saída anterior mostra que o Roteador4 envia pacotes por broadcast e os envia para o endereço de broadcast Ethernet FFFF.FFFF.FFFF. Aqui, 0000.0000.0000 significa que o Roteador4 procura o endereço MAC do destino 172.16.100.5. Como ele não sabe o endereço MAC enquanto o ARP é solicitado neste exemplo, ele usa 000.0000.000 como um espaço reservado

nos quadros de broadcast enviados da interface Ethernet 0 e pergunta qual endereço MAC corresponde a 172.16.100.5. Se não houver resposta, o endereço MAC que corresponde ao endereço IP na saída de **show arp** é marcado como incompleto:

```
Router4#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.16.100.4               -   0000.0c5d.7a0d  ARPA   Ethernet0
Internet  172.16.100.5               0   Incomplete      ARPA
Internet  172.16.100.7               2   0060.5cf4.a955  ARPA   Ethernet0
```

Após um período predeterminado, esta entrada incompleta é removida da tabela ARP. Contanto que o endereço MAC não esteja na tabela ARP, o ping falhará como resultado de "falha de encapsulamento".

Retardo

Por padrão, se você não recebe uma resposta da extremidade remota dentro de dois segundos, o ping falha:

```
Router1#ping 172.16.0.12

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.12, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Em redes com um enlace lento ou um retardo longo, dois segundos não são bastante. Você pode alterar esse padrão com um ping estendido:

```
Router1#ping
Protocol [ip]:
Target IP address:  172.16.0.12
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 172.16.0.12, timeout is 30 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1458/2390/6066 ms
```

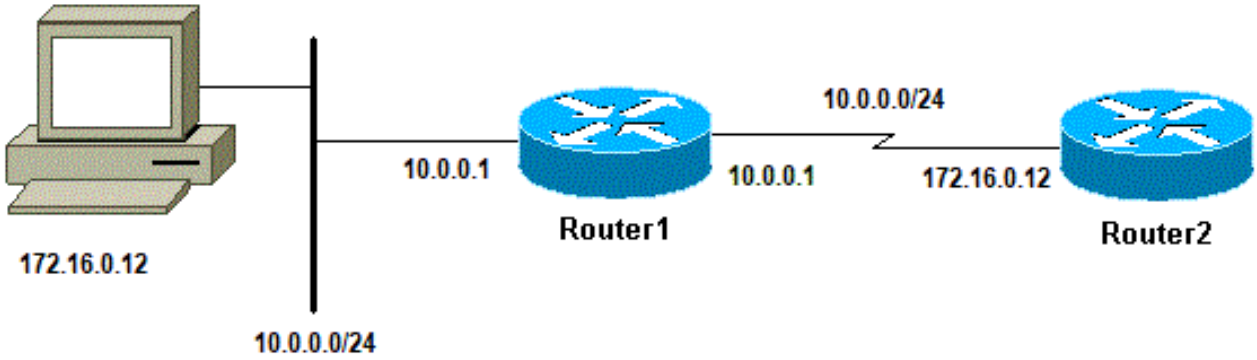
Para obter mais informações sobre o comando ping estendido, consulte Compreender os Comandos Ping Estendido e Traceroute Estendido .

No exemplo anterior, quando o tempo limite foi aumentado, o ping teve êxito.

> **Note**: O tempo médio de round-trip é maior do que dois segundos.

Endereço de origem correto

Este exemplo é um cenário comum:



Endereço de origem correto

Adicione uma interface LAN ao Router1:

```
Router1(config)#interface ethernet0
Router1(config-if)#ip address 10.0.0.1 255.255.255.0
```

A partir de uma estação na LAN, você pode fazer ping no Roteador 1. A partir do Roteador 1 você pode fazer ping no Roteador 2. Mas a partir de uma estação na LAN, você não pode fazer ping no Roteador 2.

A partir do Roteador 1, é possível executar o ping no Roteador 2 pois, por padrão, você usa o endereço IP da interface de saída como o endereço de origem no seu pacote ICMP. O Roteador2 não tem informações sobre essa nova LAN. Se tiver que responder a um pacote dessa rede, ele não saberá como tratá-lo.

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

> **aviso:** Quando o comando **debug ip packet** é usado em um roteador de produção, ele pode causar alta utilização da CPU. Isso pode resultar em uma grave degradação do desempenho ou uma interrupção da rede.

```
Router1#ping 172.16.0.12

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.12, 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=172.16.12.1 (local), d=172.16.0.12 (Serial0), len 100, sending
Jan 20 16:35:54.259: IP: s=172.16.0.12 (Serial0), d=172.16.12.1 (Serial0), len 100, rcvd 3
```

O exemplo de saída anterior funciona porque o endereço de origem do pacote enviado é 172.16.12.1. Para simular um pacote da LAN, você precisa usar um ping estendido:

```
Router1#ping
Protocol [ip]:
Target IP address: 172.16.0.12
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
```

```
Extended commands [n]: y
Source address or interface: 10.0.0.1
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 172.16.0.12, timeout is 2 seconds:

Jan 20 16:40:18.303: IP: s=10.0.0.1 (local), d=172.16.0.12 (Serial0), len 100,
 sending.
Jan 20 16:40:20.303: IP: s=10.0.0.1 (local), d=172.16.0.12 (Serial0), len 100,
 sending.
Jan 20 16:40:22.303: IP: s=10.0.0.1 (local), d=172.16.0.12 (Serial0), len 100,
 sending.
Jan 20 16:40:24.303: IP: s=10.0.0.1 (local), d=172.16.0.12 (Serial0), len 100,
 sending
Jan 20 16:40:26.303: IP: s=10.0.0.1 (local), d=172.16.0.12 (Serial0), len 100,
 sending.
Success rate is 0 percent (0/5)
```

Desta vez, o endereço de origem é 10.0.0.1 e não funciona. Os pacotes são enviados, mas nenhuma resposta é recebida. Para corrigir esse problema, adicione uma rota a 10.0.0.0 no Roteador 2. A regra básica é que o dispositivo que recebeu o ping também deve saber como enviar a resposta para a origem do ping.

Quedas de fila de entrada altas

Quando um pacote entra no roteador, o roteador tenta encaminhá-lo a um nível de interrupção. Se uma combinação não pode ser encontrada em uma tabela de cache apropriada, o pacote está enfileirado na fila de entrada da interface de entrada a ser processada. Alguns pacotes sempre são processados, mas com a configuração apropriada e nas redes estáveis, a taxa de pacotes processados nunca deve congestionar a fila de entrada. Se a fila de entrada estiver cheia, o pacote será descartado

Embora a interface esteja ativa e você não possa fazer ping no dispositivo devido a quedas altas na fila de entrada. Você pode verificar as quedas de entrada com o comando **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 supressed
```

Como considerado do para output, a queda de fila de entrada é alta. Consulte Solucionar Problemas de Perdas de Filas de Entrada e Saída para solucionar problemas de perdas de filas de Entrada/Saída.

O comando Traceroute

O comando **traceroute** é usado para descobrir as rotas que os pacotes realmente fazem quando trafegam até seu destino. O dispositivo (por exemplo, um roteador ou um PC) envia uma seqüência de datagramas de Protocolo UDP para um endereço de porta inválido no host remoto.

Três datagramas são enviados, cada um com um valor de campo Time-To-Live (TTL) definido como um. O valor TTL de 1 provoca "timeout" no datagrama assim que este bate o primeiro roteador no trajeto; esse roteador responde com uma mensagem de tempo excedido (TEM) ICMP que indica que o datagrama expirou.

Outras três mensagens de UDP são agora enviadas, cada uma com o valor de TTL definido como 2, que faz com que o segundo roteador retorne ICMP TEMs. Este processo continua até que os pacotes realmente alcancem o outro destino. Como esses datagramas tentam acessar uma porta inválida no host de destino, mensagens ICMP de porta inalcançável são retornadas e indicam uma porta inalcançável; este sinais de evento o programa Traceroute que está terminado.

A finalidade atrás desta é gravar a fonte de cada Time Exceeded Message ICMP para fornecer um traço do trajeto que o pacote tomou para alcançar o destino.

```
Router1#traceroute 172.16.4.34

Type escape sequence to abort.
Tracing the route to 172.16.4.34

  1 172.16.0.12 4 msec 4 msec 4 msec
  2 10.0.3.23 20 msec 16 msec 16 msec
  3 172.16.4.34 16 msec *  16 msec

Jan 20 16:42:48.611: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 28,
 sending
Jan 20 16:42:48.615:     UDP src=39911, dst=33434
Jan 20 16:42:48.635: IP: s=172.16.0.12 (Serial0), d=172.16.12.1 (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=172.16.12.1 (local),
d=172.16.4.34 (Serial0), len 28, sending Jan 20 16:42:48.647: UDP src=34237, dst=33435 Jan 20
16:42:48.667: IP: s=172.16.0.12 (Serial0), d=172.16.12.1 (Serial0), len 56, rcvd 3 Jan 20
16:42:48.671: ICMP type=11, code=0 Jan 20 16:42:48.675: IP: s=172.16.12.1 (local), d=172.16.4.34
(Serial0), len 28, sending Jan 20 16:42:48.679: UDP src=33420, dst=33436 Jan 20 16:42:48.699:
IP: s=172.16.0.12 (Serial0), d=172.16.12.1 (Serial0), len 56, rcvd 3 Jan 20 16:42:48.703: ICMP
type=11, code=0
```

Essa é a primeira sequência de pacotes enviada com um TTL=1. O primeiro roteador, nesse caso, o Roteador2 (172.16.0.12), descarta o pacote e envia de volta à origem (172.16.12.1) uma mensagem ICMP tipo=11. Isso corresponde à Mensagem de tempo excedido.

```
Jan 20 16:42:48.707: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 28,
```

```
 sending
Jan 20 16:42:48.711:     UDP src=35734, dst=33437
Jan 20 16:42:48.743: IP: s=10.0.3.23 (Serial0), d=172.16.12.1 (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=172.16.12.1 (local),
d=172.16.4.34 (Serial0), len 28, sending Jan 20 16:42:48.755: UDP src=36753, dst=33438 Jan 20
16:42:48.787: IP: s=10.0.3.23 (Serial0), d=172.16.12.1 (Serial0), len 56, rcvd 3 Jan 20
16:42:48.791: ICMP type=11, code=0 Jan 20 16:42:48.795: IP: s=172.16.12.1 (local), d=172.16.4.34
(Serial0), len 28, sending Jan 20 16:42:48.799: UDP src=36561, dst=33439 Jan 20 16:42:48.827:
IP: s=10.0.3.23 (Serial0), d=172.16.12.1 (Serial0), len 56, rcvd 3 Jan 20 16:42:48.831: ICMP
type=11, code=0
```

O mesmo processo ocorre para o Roteador3 (10.0.3.23) com um TTL=2:

```
Jan 20 16:42:48.839: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 28,
 sending
Jan 20 16:42:48.843:     UDP src=34327, dst=33440
Jan 20 16:42:48.887: IP: s=172.16.4.34 (Serial0), d=172.16.12.1 (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=172.16.12.1 (local),
d=172.16.4.34 (Serial0), len 28, sending Jan 20 16:42:48.899: UDP src=37534, dst=33441 Jan 20
16:42:51.895: IP: s=172.16.12.1 (local), d=172.16.4.34 (Serial0), len 28, sending Jan 20
16:42:51.899: UDP src=37181, dst=33442 Jan 20 16:42:51.943: IP: s=172.16.4.34 (Serial0),
d=172.16.12.1 (Serial0), len 56, rcvd 3 Jan 20 16:42:51.947: ICMP type=3, code=3
```

Com um TTL=3, o Roteador4 é finalmente alcançado. Dessa vez, já que a porta não é válida, o roteador 4 envia de volta para o roteador 1 uma mensagem ICMP com tipo=3, uma mensagem de destino inalcançável e um código=3 de porta inalcançável.

A próxima tabela lista os caracteres que podem aparecer na saída do comando **traceroute**.

Caracteres de texto traceroute de IP

Caractere	Descrição
nn msec	Para cada nó, o Round-Trip Time nos milissegundos para o número especific de pontas de prova
*	O tempo da prova esgotou
R	Administrativamente proibido (exemplo, lista de acesso)
P	Contenção de origem (destino muito ocupado)
I	Teste interrupção do usuário
U	Porta inalcançável
H	Host inalcançável
N	Rede inacessível
P	Protocolo inacessível
T	Timeout
?	Tipo de pacote desconhecido

Desempenho

Você pode obter o tempo de ida e volta (RTT) com os comandos **ping** e **traceroute** . Esse é o tempo necessário para enviar um pacote de eco e obter uma resposta. Isso pode fornecer uma ideia aproximada do atraso no link. Contudo, estas figuras não são precisas bastante ser usadas para a avaliação de desempenho.

Quando um destino do pacote é o roteador próprio, este pacote tem que ser comutado por processamento. O processador precisa lidar com as informações desse pacote e enviar uma resposta. Este não é o principal objetivo de um roteador. Por definição, um roteador é construído para rotear pacotes. Um ping respondido é oferecido como um serviço de melhor esforço.

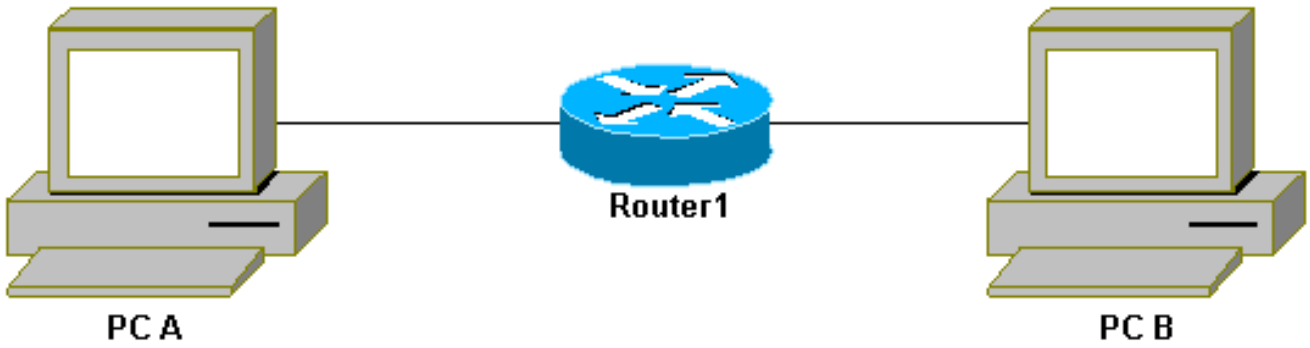
Para ilustrar isso, este é um exemplo de um ping do Roteador 1 para o Roteador 2:

```
Router1#ping 172.16.0.12

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.12, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

O RTT é aproximadamente quatro milissegundos. Depois que você permite alguns recursos de processo intensivos em Roteador 2, tente executar o ping no Roteador 2 do Roteador 1.

```
Router1#ping 172.16.0.12

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.12, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/25/28 ms
```

O RTT aumentou dramaticamente. Router2 está bastante ocupado e a prioridade é não responder ao ping. Uma maneira melhor de testar o desempenho do roteador é com o tráfego que passa pelo roteador.



Tráfego através do roteador

O tráfego é então comutado rapidamente e é tratado pelo roteador com a prioridade mais alta. A rede básica ilustra isso:

Roteadores de rede 3 básicos

Faça ping no Roteador 3 a partir do Roteador 1:

```
Router1#ping 10.0.3.23

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.3.23, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
```

O tráfego passa pelo Roteador 2 e agora é comutado rapidamente. Ative o recurso de processamento intensivo no Roteador 2:

```
Router1#ping 10.0.3.23

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.3.23, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
```

Não há quase nenhuma diferença. Isto porque, no Roteador2, os pacotes agora são tratados no nível de interrupção.

Utilizar o comando debug

Antes de utilizar **comandos debug, consulte Informações Importantes sobre Comandos Debug.**

Os diferentes comandos **debug** usados neste artigo mostram o que acontece quando um comando **ping** ou **traceroute** é usado. Esses comandos podem ajudá-lo a solucionar problemas. No entanto, em um ambiente de produção, as depurações devem ser usadas com cuidado. Se seu CPU não é poderoso, ou se você tem muitos pacotes comutados por processamento, estes podem facilmente parar seu dispositivo. Há diversas maneiras de minimizar o impacto do **comando debug no roteador.** Uma maneira é usar listas de acesso para restringir o tráfego específico a ser monitorado.

Aqui está um exemplo:

```
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 172.16.12.1 host 172.16.4.34
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 172.16.12.1 host 172.16.4.34 (5 matches)
```

Com essa configuração, o Roteador 4 imprime apenas a mensagem de depuração que corresponde à lista de acesso 150. Um ping do Roteador 1 faz com que esta mensagem seja exibida:

```
Router4#
Jan 20 16:51:16.911: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
 rcvd 3
Jan 20 16:51:17.003: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
 rcvd 3
Jan 20 16:51:17.095: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
 rcvd 3
Jan 20 16:51:17.187: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
 rcvd 3
Jan 20 16:51:17.279: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
 rcvd 3
```

A resposta para o problema não vem do Router4 porque esses pacotes não correspondem à lista de acesso. Para vê-los, adicione:

```
Router4(config)#access-list 150 permit ip host 172.16.12.1 host 172.16.4.34
Router4(config)#access-list 150 permit ip host 172.16.4.34 host 172.16.12.1
```

Resultados:

```
Jan 20 16:53:16.527: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.531: IP: s=172.16.4.34 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.627: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.635: IP: s=172.16.4.34 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.727: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.731: IP: s=172.16.4.34 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.823: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.827: IP: s=172.16.4.34 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.919: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.923: IP: s=172.16.4.34 (local), d=172.16.12.1 (Serial0), len 100,
 sending
```

Outra maneira de reduzir o impacto do comando **debug** é armazenar em buffer as mensagens de depuração e exibi-las com o comando **show log** depois que a depuração for desativada:

```
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 172.16.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, 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=172.16.4.34 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:55:46.679: IP: s=172.16.12.1 (Serial0), d=172.16.4.34 (Serial0), len 100,
 rcvd 3
```

Os comandos **ping** e **traceroute** são utilitários úteis que você pode usar para solucionar problemas de acesso à rede. Eles também são muito fáceis de utilizar. Esses dois comandos são amplamente usados pelos engenheiros de rede.

Informações Relacionadas

- Entender os comandos ping e Traceroute estendidos
- Suporte Técnico - Cisco Systems

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.