Compreendendo os comandos ping and traceroute

Índice

Introdução

Este original ilustra o uso dos **comandos ping and traceroute**. Com o auxílio de alguns **comandos debug**, este capturas de documento mais vista detalhada de como estes comandos trabalham.

Note: Permitir todos os **comandos debug em um** roteador de produção pode causar problemas graves. Nós recomendamos que você leia com cuidado a seção do uso a comando Debug antes que você emita **comandos debug**.

Pré-requisitos

Requisitos

Não existem requisitos específicos para este documento.

Componentes Utilizados

Este original não é restringido à versão de software e hardware específica.

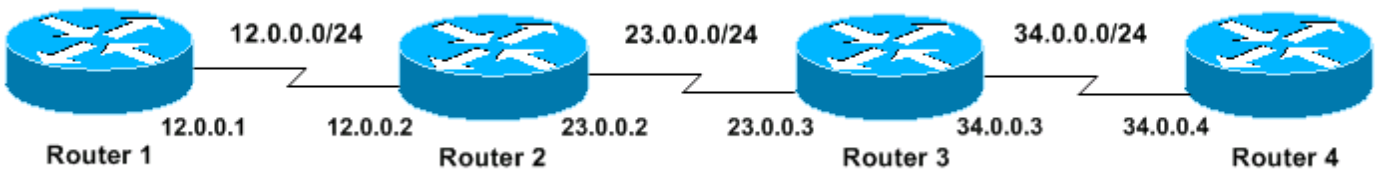
As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos usados neste original começaram com uma configuração cancelada (do padrão). Se sua rede está viva, certifique-se de que você compreende o impacto potencial do comando any.

Convenções

Para obter mais informações sobre das convenções de documento, refira as convenções dos dicas técnicas da Cisco.

Informações de fundo

Neste documento, nós usamos a configuração básica mostrada abaixo como base para nossos exemplos:



O comando Ping

O **comando ping** é um método muito comum para pesquisar defeitos a acessibilidade dos 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 é ativo ou inativo.
- O retardo round trip na comunicação com o host.
- Perda de pacotes.

O comando ping primeiramente envia um pacote de requisição de eco a um endereço, a seguir espera 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.

Para todas as opções sobre este comando, veja o “sibilo” sob comandos de Troubleshooting.

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

Está aqui um exemplo de emissor que mostra o **comando ping** após ter permitido o **comando debug ip packet detail**:

Aviso: Usar o **comando debug ip packet detail em um** roteador de produção pode causar a utilização elevada da CPU. Isso pode causar uma grave degradação do desempenho ou uma

parada de rede. Nós recomendamos que você lê com cuidado o uso o comando Debug antes de emitir **comandos 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
!--- 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.
```

A tabela a seguir lista os possíveis valores de tipo de ICMP.

Tipo de ICMP	Literal
0	resposta de eco
3	código de destino inacessível 0 = rede inacessível 1 = host inacessível 2 = protocolo inacessíveis 3 = porta inacessível 4 = fragmentação necessário e DF ajustado 5 = rota de origem falha
4	fonte-extinga
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 serviço e o host
6	alternativo-endereço
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-resposta
15	requisição de informações

16	informação-resposta
17	máscara-pedido
18	mask-reply
31	conversion-error
32	mobile-redirect

A tabela a seguir lista os possíveis caracteres de saída da instalação de ping:

Caráter	Descrição
!	Cada ponto de exclamação indica o recibo de uma resposta.
.	Cada período indica o servidor de rede cronometrado para fora ao esperar uma resposta.
U	Um erro de destino inalcançável PDU foi recebido.
Q	A fonte extingue (destino demasiado ocupado).
M	Não foi possível fragmentar.
?	Tipo de pacote desconhecido.
&	Duração de pacote excedida.

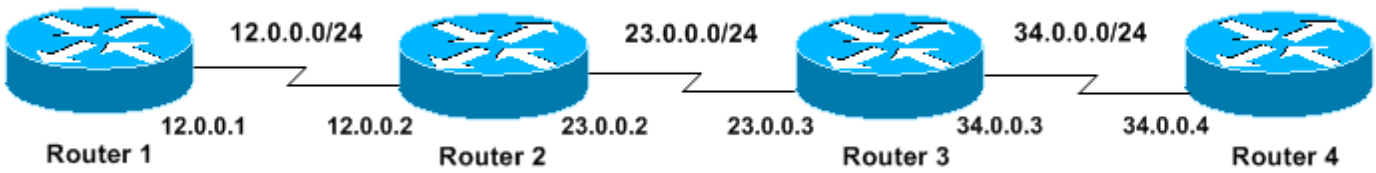
Por que não é possível executar ping?

Se você não pode executar o ping com sucesso em um endereço, considere estas causas:

Problema de Roteamento

Estão aqui os exemplos de tentativas mal sucedidas de execução do ping, determinando o problema, e que a fazer para resolver o problema.

Esse cenário é explicada no diagrama de topologia de rede a seguir:




```
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
!
!
```

Deixe-nos tentar executar o ping no Roteador4 a partir do Roteador 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:
.....
Success rate is 0 percent (0/5)
```

Deixe-nos ter um olhar mais atento no que aconteceu:

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

Aviso: Usar o **comando debug ip packet em um** roteador de produção pode causar a utilização da alta utilização da CPU. Isso pode causar uma grave degradação do desempenho ou uma parada de rede. Nós recomendamos que você lê com cuidado o uso o comando Debug antes de emitir **comandos 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)
```

Já que nenhum Routing Protocol está em execução no Router 1, ele não sabe para onde enviar seu pacote, portanto, recebemos uma mensagem unroutable.

Deixe-nos agora adicionar 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
```

Agora temos:

```
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
```

Deixe-nos agora examinam o que está errado no Roteador 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
```

O roteador 1 está enviando seus pacotes corretamente para o roteador 2, mas este não sabe como acessar o endereço 34.0.0.4. Router2 envia para trás “uma mensagem ICMP inacessível” a Router1.

Agora vamos habilitar o Routing Information Protocol (RIP) no Roteador 2 e no Roteador 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
```

Agora temos:

```
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)
```

Isso é um pouco melhor. O Roteador 1 está enviando pacotes ao Roteador 4, mas não está obtendo nenhuma resposta do Roteador 4.

Deixe-nos ver qual pode ser o problema no Roteador4:

```
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
```

O Roteador 4 recebe os pacotes de ICMP e tenta responder à rede 12.0.0.1, mas, como não tem uma rota nessa rede, ele simplesmente falha.

Deixe-nos adicionar uma rota estática ao Roteador4:

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

Agora ele funciona perfeitamente, e ambos os lados podem acessar um ao outro:

```
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
```

Interface down

Esta é uma situação onde a relação para de funcionar. No exemplo abaixo, tentamos enviar um ping de Router1 a Router4:

```
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)
```

Desde que o roteamento é muito bem, nós faremos o Troubleshooting ponto por ponto. Primeiramente, deixe-nos tentar executar o ping no Roteador 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
```

Pelo que vimos acima, o problema está entre o roteador 2 e o roteador 3. Uma possibilidade é que a interface serial no Router3 tenha sido fechada:

```
Router3#show ip interface brief
Serial0   34.0.0.3    YES manual up                    up
Serial1   23.0.0.3    YES manual administratively down   down
```

Isto é bastante simples para fixar:

```
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
```

Comando da lista de acesso

Nesta encenação, nós queremos permitir que somente o tráfego do telnet entre em Roteador4 através da relação 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
```

Refira o uso à seção de comando Debug usando Listas de acesso com **comandos debug**.

Quando nós tentamos agora sibilar Roteador4, nós temos o seguinte:

```
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
```

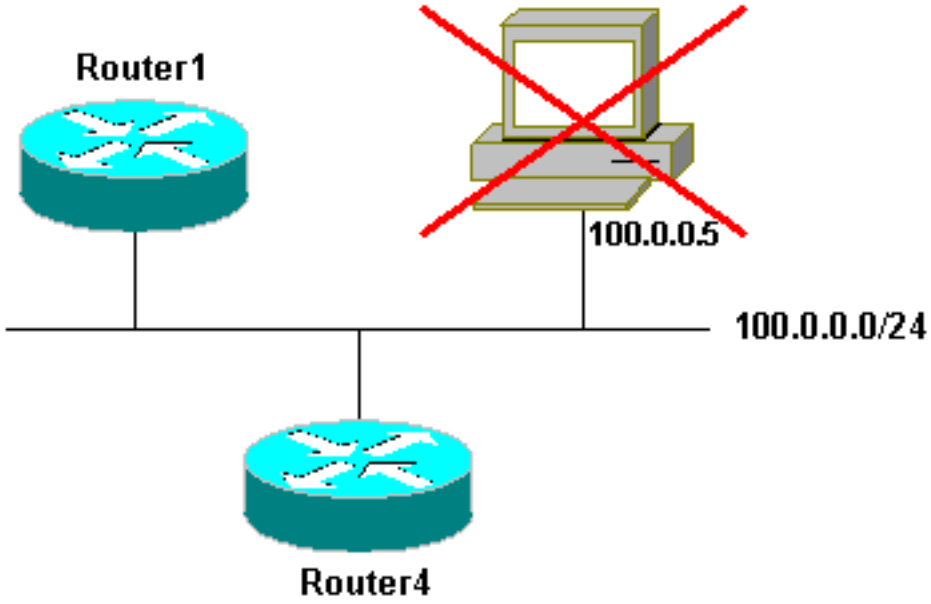
```
Jan 20 16:34:51.295: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
 sending
Jan 20 16:34:51.367: IP: s=34.0.0.4 (Serial0), d=12.0.0.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:34:51.371: ICMP: dst (12.0.0.1) administratively prohibited unreachable
 rcv from 34.0.0.4
Jan 20 16:34:51.379: IP: s=12.0.0.1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

Na extremidade de um **comando access-list**, nós mandamos sempre um implícito “negar tudo”. Isto significa que os pacotes ICMP que estão incorporando a relação da série 0 em Roteador4 estão negados, e Roteador4 envia uma mensagem “inacessível” administrativamente proibida ICMP à fonte do pacote original segundo as indicações da mensagem **debugar**. A solução é adicionar a seguinte linha no **comando access-list**:

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

Problema de Protocolo de resolução de endereços (ARP)

Está aqui uma encenação com uma conexão 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#
```

Neste exemplo, o ping não funciona devido à “falha de capsulagem”. Isso significa que o roteador sabe em que interface ele deve enviar o pacote, mas não sabe como fazê-lo. Neste caso, você precisa de compreender como o protocolo Protocolo de Resolução de Endereços (ARP) funciona. Veja configurar métodos de organização de endereços para uma explicação detalhada.

Basicamente, o ARP é um protocolo usado para mapear o endereço da camada 2 (endereço MAC) para o endereço da camada 3 (endereço IP). Você pode verificar este mapeamento usando o **comando 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
```

Retorne ao problema falhado “encapsulamento”. Nós obtemos uma ideia melhor do problema usando este **comando 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)
```

A saída acima mostra que o Roteador 4 está difundindo pacotes enviando-os para o endereço de difusão de Ethernet FFFF.FFFF.FFFF. Aqui, o 0000.0000.0000 significa que o Roteador4 está procurando o MAC address do destino 100.0.0.5. Já que ele não conhece o endereço MAC durante a requisição ARP neste exemplo, ele usa 0000.0000.000 como um espaço reservado nos quadros de broadcast enviados para fora da interface Ethernet 0, perguntando qual endereço MAC corresponde a 100.0.0.5. Se nós não obtemos uma resposta, o endereço correspondente nos outputis arp da mostra **marcados** como incompleto:

```
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
```

Após um período predeterminado, esta entrada incompleta é removida da tabela ARP. Enquanto o MAC address correspondente não está na tabela ARP, o ping falha em conseqüência da “falha de capsulagem”.

Atraso

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

```
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)
```

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

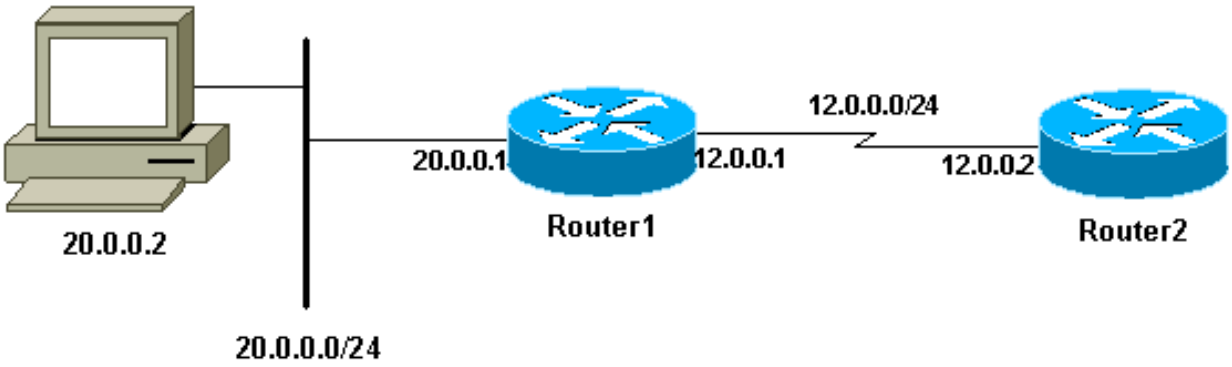
```
Router1#ping
Protocol [ip]:
Target IP address:  12.0.0.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]: 30
Extended commands [n]:
Sweep range of sizes [n]:

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
```

No exemplo acima, aumentar o intervalo levou a um ping bem-sucedido.

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

Endereço de origem correto

Está aqui um exemplo de uma situação típica:



Adicionar uma interface LAN ao Roteador1:

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

De uma estação no LAN, você pode sibilar Router1. Do Roteador 1 é possível executar o ping do Roteador 2. Mas, a partir de uma estação na LAN, você não pode efetuar ping no Router2.

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. Router2 não tem a informação sobre este LAN novo. Se tem que responder a um pacote que vem desta rede, não sabe segurá-la.

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

Aviso: Usar o **comando debug ip packet em um** roteador de produção pode causar a utilização da alta utilização da CPU. Isso pode causar uma grave degradação do desempenho ou uma parada de rede. Nós recomendamos que você lê com cuidado o uso o comando Debug antes de emitir **comandos 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
```

O exemplo de saída acima funciona porque o endereço de origem do pacote que estamos enviando é s=12.0.0.1. se quisermos simular um pacote recebido da LAN, temos que usar um ping estendido:

```
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)
```

Dessa vez, o endereço de origem é 20.0.0.1 e não está funcionando! Estamos enviando nossos pacotes, mas não estamos recebendo nada. Para corrigir esse problema, temos que adicionar uma rota a 20.0.0.0 no Roteador2.

A regra básica é que o dispositivo sibilado deve igualmente saber enviar a resposta à fonte do sibilo.

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 está completa, o pacote está deixado cair.

Embora a relação é ascendente e você não pode executar o ping do dispositivo devido às quedas de fila de entrada altas. Você pode verificar a queda da 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. Refira a pesquisando defeitos quedas de fila de entrada e quedas da fila de saída a fim pesquisar defeitos quedas da fila do entrada/saída.

O comando traceroute

O **comando traceroute** é usado para descobrir as rotas que os pacotes realmente tomam ao viajar a 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; em seguida, esse roteador responde com um Time Exceeded Message (TEM) ICMP, indicando 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. Desde que estes datagramas estão tentando alcançar uma porta inválida no host de destino, os mensagens inalcançadas da porta ICMP são retornados, indicando 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. Para todas as opções sobre este comando, veja Traçar (privilegiado).

```
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
```

Esta é a primeira seqüência de pacotes enviada com um TTL=1. O primeiro roteador, nesse caso, o Roteador 2 (12.0.0.2), desconecta o pacote e retorna à origem (12.0.0.1) uma mensagem ICMP

tipo=11. Isso corresponde à Mensagem de tempo excedido.

```
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
```

O mesmo processo ocorre para o Roteador3 (23.0.0.3) com um 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
```

Com um TTL=3, nós alcançamos finalmente Roteador4. 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.

As lista abaixo da tabela os caráteres que podem aparecer na saída do **comando traceroute**.

Caráteres do texto IP Traceroute

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

Desempenho

Usando os **comandos ping and traceroute**, nós obtemos o Round-Trip Time (RTT). Este é o tempo necessário para enviar um pacote de eco e obter uma resposta de volta. Isto pode ser útil ter uma ideia bruta do atraso na relação. 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 tem que segurar a informação deste pacote, e envia uma resposta. Este não é o principal objetivo de um roteador. Por definição, um roteador é construído para rotear pacotes. A resposta a um ping é oferecida como um empenho máximo de serviço.

Para ilustrar isto, é aqui um exemplo de um sibilo de Router1 a Router2:

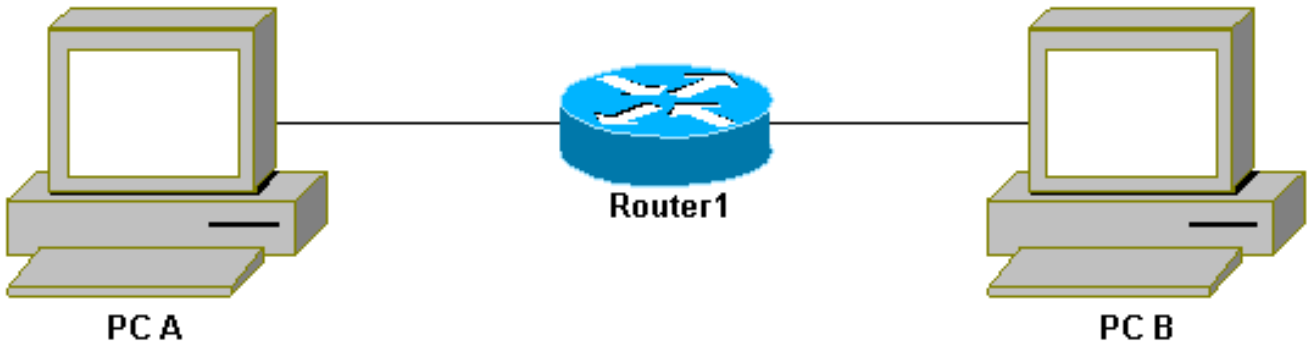
```
Router1#ping 12.0.0.2

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
```

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 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
```

O RTT aumentou dramaticamente. Roteador2 está muito ocupado e responder ao ping não é sua prioridade principal.

Uma maneira melhor ao desempenho do roteador de teste é com o tráfego que dirige o roteador:



O tráfego é então fast-switched, e é segurado pelo roteador com a prioridade mais alta. Para ilustrar isso, vamos voltar à rede básica.

Deixe-nos executar o Roteador3 do Roteador 1:

```
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
```

O tráfego está passando por Router2 e agora está Fast-Switched.

Deixe-nos agora permitir os recursos de processo intensivos em Roteador 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
```

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

Utilizar o comando debug

Antes de emitir **comandos debug**, veja por favor a informação importante em comandos Debug.

Os diferentes **comandos debug** que nós temos usado até agora nos da uma introspecção no que acontece quando nós usamos um **comando ping ou traceroute**. Elas também podem ser úteis para Troubleshooting. Contudo, em um ambiente de produção, o debug deve ser usado 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. Está aqui 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 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)
```

Com essa configuração, o Roteador4 imprime apenas a mensagem de depuração que coincide com a lista de acesso 150. Um ping proveniente do Roteador1 faz com que a seguinte mensagem seja exibida:

```
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)
```

A resposta do Roteador 4 não é mais visível porque tais pacotes não correspondem à lista de acesso. Para visualizá-los, deve-se incluir o seguinte:

```
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
```

Temos então:

```
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
```

Uma outra maneira de minimizar o impacto do comando debug é proteger as mensagens debugar e mostrá-las que usam o comando show log uma vez que debugar foi desligado:

```
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
```

Como você pode ver, os **comandos ping e traceroute** são utilitários muito eficientes que você pode usar para pesquisar defeitos problemas do acesso de rede. Eles também são muito fáceis de utilizar. Porque estes dois comandos são os comandos os mais amplamente utilizados por engenheiros de rede, compreendê-los é crucial para pesquisar defeitos a conectividade de rede.

Informações Relacionadas

- Usando os comandos ping e traceroute estendidos
- Suporte técnico - Cisco Systems