# Explicación de los comandos ping y traceroute

## Contenido



## Introducción

En este documento se describe el uso de los comandos ping y traceroute en routers Cisco.

## Prerequisites

### Requirements

No hay requisitos específicos para este documento.

### Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

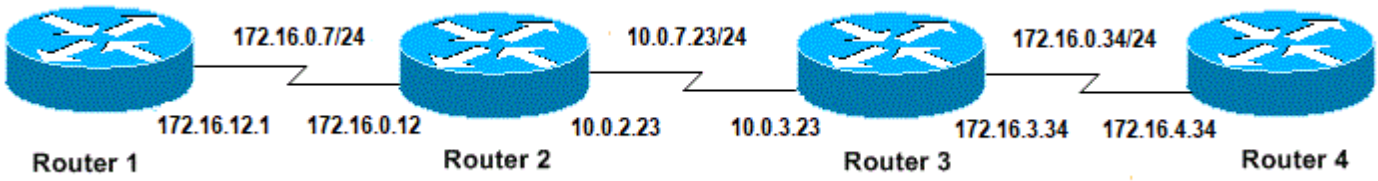
### Convenciones

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

## Antecedentes

Nota: Cualquier comando debug utilizado en un router de producción puede causar problemas serios. Lea la sección Uso del Comando Debug antes de ejecutar los comandos debug.

En este documento, esta configuración básica se utiliza para los ejemplos de este artículo:




Configuración básica de IP y routers

## El comando ping

El comando ping es un método muy común utilizado para resolver problemas de accesibilidad de dispositivos. Utiliza una serie de mensajes eco del protocolo Internet Control Message Protocol (ICMP) para determinar:

- Si un host remoto está activo o inactivo.
- Retraso de ida y vuelta utilizado para comunicarse con el host.
- Pérdida de paquetes.

En primer lugar, el comando ping envía un paquete de solicitud de eco a una dirección y después espera una respuesta. El ping es exitoso solo si:

- la petición de eco llega a destino y
- el destino puede obtener una respuesta de eco de regreso al origen dentro de un tiempo predeterminado llamado tiempo de espera agotado. El valor predeterminado de dicho tiempo de espera es de dos segundos en los routers de Cisco.

El valor TTL de un paquete ping no puede modificarse.

En el siguiente ejemplo de código se muestra el comando ping después de habilitar el comando debug ip packet detail .

Advertencia: Cuando se utiliza el comando debug ip packet detail en un router de

producción, puede causar una alta utilización de la CPU. Esto puede provocar una degradación grave del rendimiento o una interrupción de la red.

<#root>

```
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 de tipo ICMP posibles

| Tipo de ICMP | Literal |
|---|---|
| 0 | respuesta de eco |
| 3 | código de destino inalcanzable 0 = red inalcanzable 1 = host inalcanzable 2 = protocolo inalcanzable 3 = puerto inalcanzable 4 = fragmentación necesaria y DF configurado 5 = ruta de origen errónea |
| 4 | desconexión del origen |
| 5 | redireccionar código 0 = redireccionar datagramas para la red 1 = redireccionar |

| | |
|---|---|
| | datagramas para el host 2 = redireccionar datagramas para el tipo de servicio y la red 3 = redireccionar datagramas para el tipo de servicio y el host |
| 6 | dirección alternativa |
| 8 | eco |
| 9 | anuncio del router |
| 10 | solicitud de router |
| 11 | código de tiempo excedido 0 = tiempo a producción excedido en tránsito 1 = tiempo de re ensamblado de fragmento excedido |
| 12 | problema de parámetro |
| 13 | timestamp-request |
| 14 | timestamp-reply |
| 15 | petición de información |
| 16 | información-respuesta |
| 17 | petición de máscara |
| 18 | respuesta de máscara |
| 31 | conversion-error |
| 32 | mobile-redirect |

Posibles caracteres de salida de la función de ping

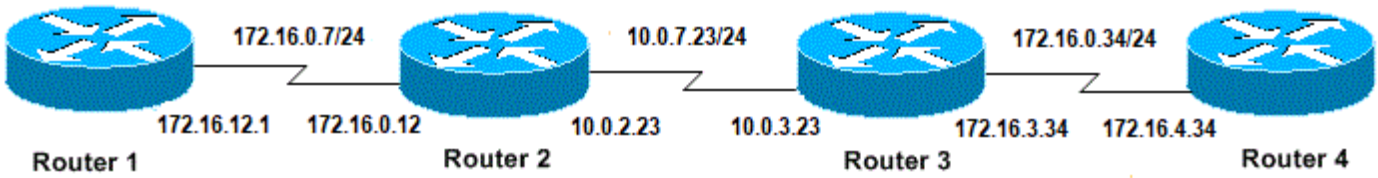
| Carácter | Descripción |
|---|---|
| ! | Cada signo de exclamación indica la recepción de una respuesta. |
| . | Cada periodo indica que el servidor de red ha agotado el tiempo de espera mientras esperaba una respuesta. |
| U | Se recibió la PUD del error "No se puede acceder al destino". |
| A | Desconexión del origen (destino demasiado ocupado). |
| M | No se pudo fragmentar. |
| ? | Tipo desconocido de paquete. |
| Y | Vida útil del paquete excedida. |

# No se puede hacer ping

Si no puede hacer un ping con éxito a una dirección IP, considere las causas que se enumeran en esta sección.

## Problema del router

Estos son ejemplos de intentos de ping fallidos, que pueden determinar el problema y qué hacer para resolverlo. Este ejemplo se muestra con este diagrama de topología de red:




Problemas del router

<#root>

**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
!
```

Intente hacer ping al Router 4 desde el Router 1:

<#root>

```
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:

<#root>

```
Router1#
```

**debug ip packet**

```
IP packet debugging is on
```

---

Advertencia: Cuando se utiliza el comando debug ip packet en un router de producción, puede causar una alta utilización de la CPU. Esto puede provocar una degradación grave del rendimiento o una interrupción de la red.

---

<#root>

```
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)
```

Dado que no se ejecuta ningún protocolo de ruteo en el Router1, no sabe dónde enviar su paquete y genera un mensaje de "no ruteo".

Agregue una ruta estática al Router1:

<#root>

```
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:

<#root>

```
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 qué está mal en el Router2:

<#root>

```
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
```

El Router1 envió correctamente sus paquetes al Router2, pero el Router2 no sabe cómo acceder a la dirección 172.16.4.34. El Router2 envía un mensaje hacia el Router1 donde informa sobre un "ICMP inalcanzable".

Habilitar el protocolo de información de enrutamiento (RIP) en el router 2 y el router 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:

<#root>

```
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)
```

El Router1 envía paquetes al Router4, pero el Router4 no envía una respuesta.

Posible problema en el Router 4:

<#root>

```
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
```

El router 4 recibe los paquetes ICMP e intenta responder a 172.16.12.1, pero como no tiene una ruta a esta red, falla.

Agregue una ruta estática al Router 4:

<#root>

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

Ahora ambos lados pueden acceder entre sí:

```
<#root>
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
```

## Interfaz inactiva

Esta es una situación en la que la interfaz deja de funcionar. En el siguiente ejemplo se intenta hacer ping al Router4 desde el Router1:

```
<#root>
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)
```

Dado que el ruteo es correcto, realice una resolución de problemas paso a paso del problema. Intente hacer ping al Router 2:

```
<#root>
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
```

Del ejemplo anterior, el problema está entre el Router2 y el Router3. Una posibilidad es que la interfaz serial en el Router 3 se haya apagado:

```
<#root>

Router3#

show ip interface brief

Serial0   172.16.3.34    YES manual up                      up
Serial1   10.0.3.23    YES manual administratively down   down
```

Esto es fácil de arreglar:

```
<#root>

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 de lista de acceso

En este escenario, sólo se permite que el tráfico telnet ingrese al Router4 a través de la interfaz Serial0.

```
<#root>

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
```

Intente hacer ping al Router 4:

<#root>

```
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
```

Al final de un comando access-list siempre hay un deny all implícito. Esto significa que se niegan los paquetes ICMP que ingresan a la interfaz Serial 0 en el Router 4, y el Router 4 envía un mensaje ICMP "administrativamente prohibido e inalcanzable" al origen del paquete original como se muestra en el mensaje de debug. La solución es agregar esta línea en el comando access-list:

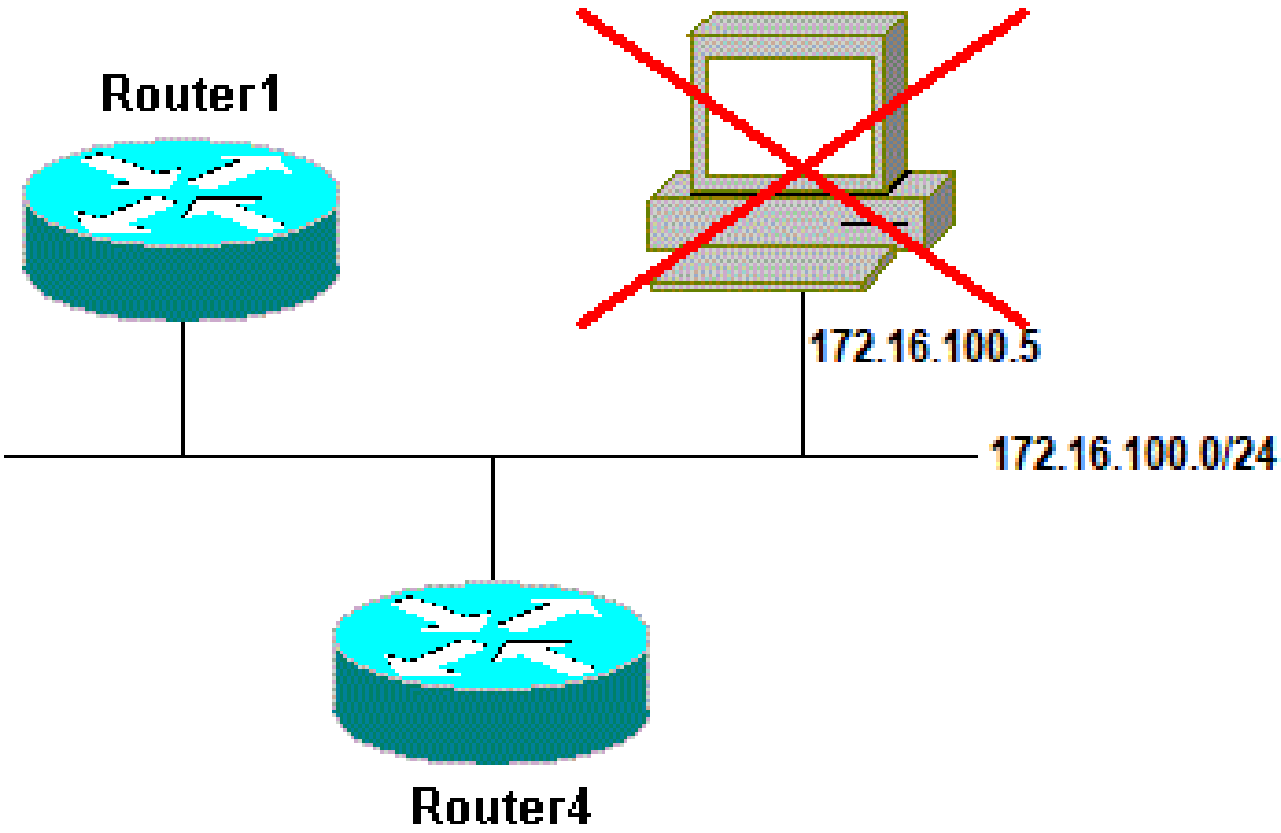
<#root>

```
Router4(config)#
```

**access-list 100 permit icmp any any**

## Problema de protocolo de resolución de direcciones (ARP)

En este escenario, esta es la conexión Ethernet:




Problema de protocolo de resolución de direcciones

<#root>

```
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#
```

En este ejemplo, el ping no funciona debido al mensaje "encapsulation failed". Esto significa que el router sabe en qué interfaz tiene que enviar el paquete pero no sabe cómo hacerlo. En este caso, debe entender cómo funciona el Protocolo de resolución de direcciones (ARP).

ARP es un protocolo utilizado para asignar la dirección de capa 2 (dirección MAC) a una dirección de capa 3 (dirección IP). Puede verificar esto con el comando show arp:

<#root>

```
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
```

Vuelva al problema de "encapsulación fallida", pero esta vez habilite el comando debug arp:

```
<#root>

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)
```

El resultado anterior muestra que el Router4 transmite los paquetes y los envía a la dirección de transmisión Ethernet FFFF.FFFF.FFFF. Aquí, el 0000.0000.0000 significa que el Router 4 busca la dirección MAC del destino 172.16.100.5. Dado que no conoce la dirección MAC mientras se solicita el ARP en este ejemplo, utiliza 0000.0000.000 como marcador de posición en las tramas de broadcast enviadas fuera de la interfaz Ethernet 0 y pregunta qué dirección MAC corresponde a 172.16.100.5. Si no hay respuesta, la dirección MAC que corresponde a la dirección IP en el resultado de show arp se marca como incompleta:

```
<#root>

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
```

Después de un período predeterminado, esta entrada incompleta se purga de la tabla ARP.

Mientras la dirección MAC no esté en la tabla ARP, el ping falla como resultado de "encapsulation failed" (error de encapsulación).

## Demora

De forma predeterminada, si no recibe una respuesta del extremo remoto en el plazo de dos segundos, el ping falla:

```
<#root>

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)
```

En las redes con un link lento o una demora prolongada, dos segundos no son suficientes. Puede cambiar este valor predeterminado con un ping extendido:

```
<#root>

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
```

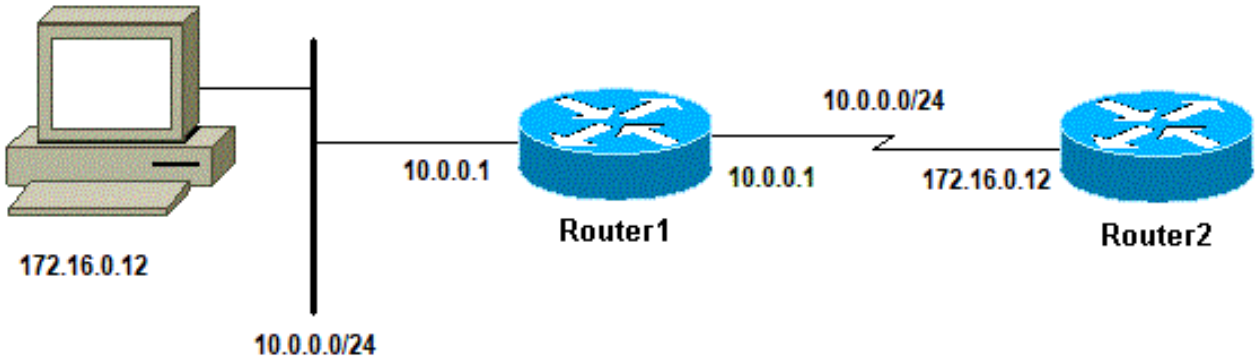
Para obtener más información sobre el comando ping extendido, vea Introducción a los comandos Extended Ping y Extended Traceroute .

En el ejemplo anterior, cuando se aumentó el tiempo de espera, el ping fue exitoso.

Nota: El tiempo medio de ida y vuelta es superior a dos segundos.

## Dirección de fuente correcta

Este ejemplo es un escenario común:




Dirección de fuente correcta

Agregue una interfaz LAN en el Router1:

```
<#root>

Router1(config)#

interface ethernet0

Router1(config-if)#

ip address 10.0.0.1 255.255.255.0
```

Desde una estación de la LAN, puede enviar un ping al Router 1. Desde el Router1 puede realizar un ping al Router2. Pero desde una estación de la LAN, no puede enviar un ping al Router2.

Desde el Router1, puede hacer ping al Router2 porque, de manera predeterminada, Usted utiliza la dirección IP de la interfaz saliente como la dirección de origen en su paquete ICMP. El Router 2 no tiene información sobre esta nueva LAN. Si tiene que responder a un paquete de esta red, no sabe cómo manejarlo.

```
<#root>

Router1#

debug ip packet

IP packet debugging is on
```

Advertencia: Cuando se utiliza el comando debug ip packet en un router de producción, puede causar una alta utilización de la CPU. Esto puede provocar una degradación grave del rendimiento o una interrupción de la red.

<#root>

```
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
```

El ejemplo de salida anterior funciona porque la dirección de origen del paquete enviado es 172.16.12.1. Para simular un paquete desde la LAN, debe utilizar un ping extendido:

<#root>

```
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)
```

Esta vez, la dirección de origen es 10.0.0.1 y no funciona. Los paquetes se envían pero no se recibe respuesta. Para solucionar este problema, agregue una ruta a 10.0.0.0 en el Router 2. La regla básica es que el dispositivo ping también debe saber cómo enviar la respuesta al origen del ping.

## Descarte de cola de entradas elevadas

Cuando un paquete ingresa en el router, el router intenta reenviarlo en el nivel de interrupción. Si no se puede encontrar una coincidencia en una tabla de caché adecuada, el paquete se coloca en la cola de entrada de la interfaz entrante que se procesará. Algunos paquetes se procesan siempre, pero con la configuración apropiada y en las redes estables, el índice de paquetes procesados nunca debe congestionar la cola de entrada. Si la cola de entrada está completa, se perderá el paquete.

Aunque la interfaz está activa, y no puede hacer ping al dispositivo debido a caídas altas de la cola de entrada. Puede verificar las caídas de entrada con el comando show interface.

<#root>

```
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
```

Según lo observado en el resultado, el descarte de cola de entrada es elevado. Consulte Resolución de Problemas de Caídas de Cola de Entrada y Caídas de Cola de Salida para resolver problemas de caídas de cola de Entrada/Salida.

## El comando traceroute

El comando traceroute se utiliza para descubrir las rutas que los paquetes toman realmente cuando viajan a su destino. El dispositivo (por ejemplo, un router o una PC) envía una secuencia de datagramas de Protocolo de datagrama de usuario (UDP) a una dirección de puerto no válida en el host remoto.

Se envían tres datagramas, cada uno con un valor de campo de Tiempo de vida (TTL) establecido en uno. El valor TTL de 1 hace que el datagrama "se agote el tiempo de espera" tan pronto como llega al primer router de la trayectoria; este router luego responde con un mensaje de tiempo excedido (TEM) ICMP que indica que el datagrama ha caducado.

Ahora se envían otros tres mensajes de UDP, cada uno de los cuales tiene el valor TTL configurado en 2. Esto hace que el segundo router devuelva TEM de ICMP. Este proceso continúa hasta que los paquetes alcancen realmente el otro destino. Dado que estos datagramas intentan acceder a un puerto no válido en el host de destino, se devuelven mensajes de puerto inalcanzable ICMP e indica un puerto inalcanzable; este evento indica al programa Traceroute que ha finalizado.

El propósito detrás de esto es registrar la fuente de cada mensaje de tiempo excedido ICMP para proporcionar la traza de la trayectoria que el paquete tomó para alcanzar el destino.

<#root>

```
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
```

Esta es la primera secuencia de paquetes que se envía con un TTL=1. El primer router, en este caso el Router2 (172.16.0.12) descarta el paquete y envía un mensaje type=11 ICMP al origens(172.16.12.1). Esto corresponde al Mensaje de tiempo excedido.

<#root>

```
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
```

El mismo proceso ocurre para el Router3 (10.0.3.23) con un TTL=2:

<#root>

```
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
```

Con un TTL=3, finalmente se llega al Router4. Esta vez, dado que el puerto no es válido, el Router4 devuelve al Router1 un mensaje ICMP tipo=3, un mensaje de destino inalcanzable, y un código=3 que significa que el puerto es inalcanzable.

La siguiente tabla enumera los caracteres que pueden aparecer en el resultado del comando traceroute.

Caracteres de Texto del Rastreador de Ruta del IP

| Carácter | Descripción |
|---|---|
| nn msec | Para cada nodo, el Round-Trip Time en milisegundos para el número especificado de probes |
| * | Se agotó el tiempo de espera del probe |
| R | Administrativamente prohibido (por ejemplo, lista de acceso) |
| A | Source quench (destino muy ocupado) |
| I | Prueba interrumpida del usuario |
| U | Puerto inalcanzable |
| H | Host fuera de alcance |
| N | Red inalcanzable |
| P | Protocolo inalcanzable |
| T | Tiempo de espera |
| ? | Tipo desconocido de paquete |

# Rendimiento

Puede obtener el tiempo de ida y vuelta (RTT) con los comandos ping y traceroute. Este es el tiempo necesario para enviar un paquete de eco y obtener una respuesta. Esto puede proporcionar una idea aproximada del retraso en el link. Sin embargo, estas cifras no son lo suficientemente exactas para ser utilizadas para la evaluación de rendimiento.

Cuando el destino de un paquete es el router en sí, se trata de un process-switched-packet. El procesador tiene que manejar la información de este paquete y enviar una respuesta. Este no es el objetivo principal de un router. Por definición, un router está diseñado para rutear paquetes. Se ofrece un ping contestado como servicio de mejor esfuerzo.

Para ilustrar esto, este es un ejemplo de un ping del Router1 al Router2:

<#root>

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

El RTT es aproximadamente cuatro milisegundos. Después de que habilite algunas características de proceso intensivo en el Router 2, intente hacer ping en el Router 2 desde el Router 1.

<#root>

Router1#

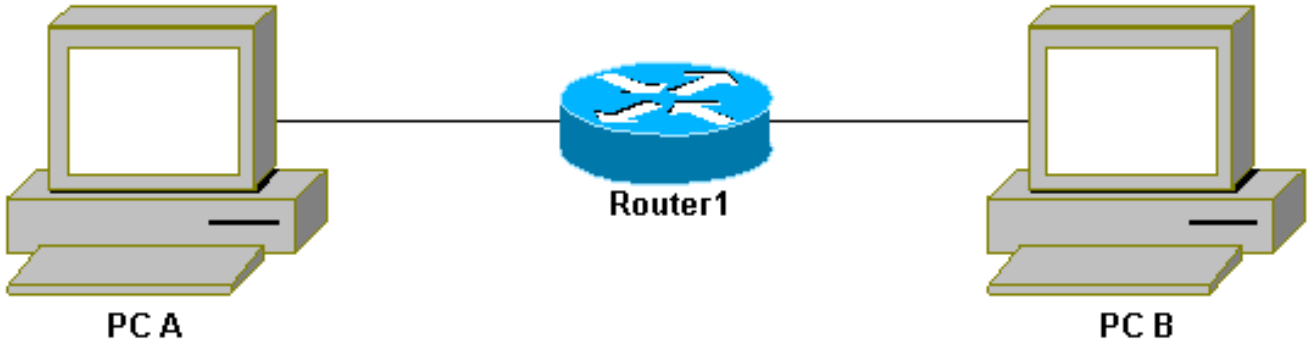
**ping 172.16.0.12**

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

El RTT ha aumentado dramáticamente aquí. El Router2 está bastante ocupado y la prioridad es no responder al ping. Una mejor manera de probar el rendimiento del router es con el tráfico que pasa a través del router.




Tráfico a través del router

El tráfico se conmuta rápidamente y es manejado por el router con la prioridad más alta. La red básica ilustra esto:


Routers de red 3 básicos

Haga ping al Router 3 desde el Router 1:

<#root>

```
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
```

El tráfico pasa a través del Router2 y ahora se conmuta rápidamente. Active la función de proceso intensivo en el Router 2:

<#root>

```
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
```

No hay casi diferencia. Esto se debe a que, en el Router 2, los paquetes se manejan ahora en el nivel de interrupción.

# Uso del Comando Debug

Antes de que utilice los comandos debug, consulte Información Importante sobre los Comandos Debug.

Los diferentes comandos debug utilizados en este artículo muestran lo que sucede cuando se utiliza un comando ping o un comando traceroute . Estos comandos pueden ayudarle a solucionar problemas. Sin embargo, en un entorno de producción, las depuraciones se deben utilizar con precaución. Si su CPU no es potente, o si tiene muchos process-switched packets, pueden fácilmente estancar su dispositivo. Hay un par de maneras de minimizar el impacto del comando debug en el router. Una manera es usar listas de acceso para acotar el tráfico específico que desea monitorear.

Aquí tiene un ejemplo:

<#root>

```
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)
```

Con esta configuración, el Router4 sólo imprime el mensaje de depuración que coincide con la lista de acceso 150. Un ping del Router1 hace que se muestre este mensaje:

```
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
```

La respuesta al problema no proviene del Router4 porque estos paquetes no coinciden con la lista de acceso. Para verlos, agregue:

<#root>

```
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
```

Otra manera de reducir el impacto del comando debug es almacenar en búfer los mensajes de depuración y mostrarlos con el comando show log una vez que se ha desactivado la depuración:

```
<#root>

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
```

Los comandos ping y traceroute son utilidades útiles que puede utilizar para resolver problemas de acceso a la red. También son muy fáciles de utilizar. Estos dos comandos son ampliamente utilizados por los ingenieros de redes.

## Información Relacionada

- Explicación de los comandos ping extendido y traceroute extendido
- Soporte Técnico - Cisco Systems

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).