

# De opdrachten pingen en traceroute begrijpen

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Achtergrondinformatie](#)

[De opdracht Ping](#)

[Kan niet pingen](#)

[Routerprobleem](#)

[Interface omlaag](#)

[Opdracht Toegangslijst](#)

[Probleem met Address Resolution Protocol \(ARP\)](#)

[Vertraging](#)

[Correct bronadres](#)

[Wachtrij voor hoge invoer](#)

[De opdracht Traceroute](#)

[Prestaties](#)

[Gebruik de opdracht Debug](#)

[Gerelateerde informatie](#)

## Inleiding

Dit document beschrijft het gebruik van de opdrachten ping en **traceroute** op Cisco-routers.

## Voorwaarden

### Vereisten

Er zijn geen specifieke vereisten van toepassing op dit document.

### Gebruikte componenten

Dit document is niet beperkt tot specifieke software- en hardware-versies.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

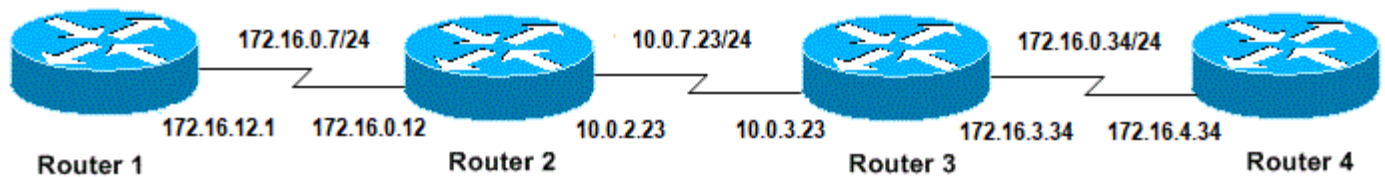
### Conventies

Raadpleeg Cisco Technical Tips Conventions (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

## Achtergrondinformatie

**Opmerking:** Om het even welk **debug** bevel dat op een productierouterie wordt gebruikt kan ernstige problemen veroorzaken. Lees het [gedeelte Gebruik de](#) sectie [Debug Command](#) voordat u **debug**-opdrachten uitvoert.

In dit document wordt deze basisconfiguratie gebruikt als voorbeeld in dit artikel:



Basisconfiguratie van IP's en routers

## De opdracht Ping

De **ping**-opdracht is een heel gebruikelijke methode om toegankelijkheid van apparaten problemen op te lossen. Het gebruikt een reeks van de Echo-berichten van het Internet Control Message Protocol (ICMP) om te bepalen:

- Of een externe host actief of inactief is.
- De ronde-trip vertraging gebruikt om met de gastheer te communiceren.
- Packet loss.

De **ping**-opdracht verstuurt eerst een echoverzoekpakket naar een adres en wacht vervolgens op een antwoord. Pingel is succesvol slechts als:

- het echo-verzoek de bestemming bereikt, en
- de bestemming is in staat om een echoantwoord terug te krijgen naar de bron binnen een vooraf bepaalde tijd genaamd een timeout. De standaardwaarde van deze timeout is twee seconden op Cisco routers.

De TTL-waarde van een **ping**-pakket kan niet worden gewijzigd.

Dit volgende codevoorbeeld toont **pingelt** bevel nadat het **debug ip** pakketdetailbevel wordt toegelaten.

**Waarschuwing:** Wanneer de opdracht **debug ip-pakketdetail** wordt gebruikt op een productierouterter, kan dit een hoog CPU-gebruik veroorzaken. Dit kan leiden tot een ernstige verslechtering van de prestaties of een netwerkstoring.

```
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.*

## Mogelijke ICMP-waarden

ICMP-type	letterlijk
0	weergalmen
3	onbereikbare code van bestemming 0 = netto onbereikbare 1 = host onbereikbare 2 = protocol onbereikbaar 3 = poort onbereikbaar 4 = fragmentatie nodig, en PDF-set 5 = bronroute mislukt
4	brondemping
5	redirect code 0 = redirect datagrammen voor het netwerk 1 = redirect datagrammen voor host 2 = redirect datagrammen voor het type service en netwerk 3 = redirect datagrammen voor het type service en host
6	afwisselend adres
8	weerkaatsen
9	router-reclame
10	routerverzoek
11	overschrijding van de tijdcode 0 = overschrijding van de levensduur tijdens het transport 1 = overschrijding van de herassemblagetijd van het fragment
12	parameterprobleem
13	tijdstempel-verzoek
14	tijdstempel-antwoord
15	verzoek om informatie
16	informatief antwoord
17	maskerverzoek
18	masker-antwoord
31	conversiefout
32	mobiele omleiding

## Mogelijke uitvoertekens uit de ping-voorziening

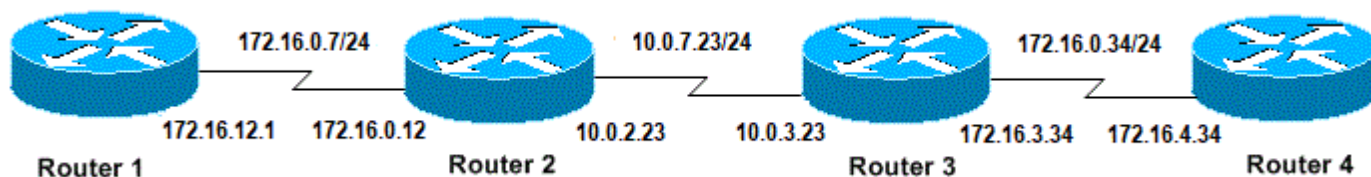
karakter	Beschrijving
!	Elk uitroepteken geeft de ontvangst van een antwoord aan.
.	Elke periode geeft de netwerkserver aan met een time-out wanneer deze wacht op een antwoord.
U	Er is een onbereikbare doelfout PDU ontvangen.
Q	Bron demping (bestemming te druk).
M	Kan niet fragmenteren.
?	Onbekend pakkettype.
en	Overschreden pakketlevensduur.

# Kan niet pingen

Als u niet met succes kunt pingen naar een IP-adres, overweeg de oorzaken die in deze sectie worden vermeld.

## Routerprobleem

Hier zijn voorbeelden van niet succesvolle pingpogingen, die het probleem kunnen bepalen, en wat te doen om het probleem op te lossen. Dit voorbeeld wordt getoond met dit diagram van de netwerktopologie:



Routerproblemen

### 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  
!
```

Probeer router 4 vanaf router 1 te pingen:

```
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)
```

Resultaten:

```
Router1#debug ip packet
```

```
IP packet debugging is on
```

**Waarschuwing:** Wanneer de **debug ip**-pakketopdracht op een productierouterter wordt gebruikt, kan dit een hoog CPU-gebruik veroorzaken. Dit kan leiden tot een ernstige verslechtering van de prestaties of een netwerkstoring.

```
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)
```

Aangezien geen routeringsprotocollen op Router1 lopen, weet het niet waar te om zijn pakket te verzenden en veroorzaakt een "unrouteable"bericht.

Voeg een statische route aan Router1 toe:

```
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
```

Resultaten:

```
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
```

Onderzoek wat op Router2 verkeerd is:

```
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
```

Router1 verzond correct zijn pakketten naar Router2, maar Router2 weet niet hoe te om tot adres 172.16.4.34 toegang te hebben. Router2 verzendt een "onbereikbaar ICMP"bericht naar Router1.

Routing Information Protocol (RIP) op router2 en router3 inschakelen:

```
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
```

Resultaten:

```
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)
```

Router1 verzendt pakketten naar Router4, maar Router4 verzendt geen antwoord terug.

Mogelijk probleem met 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
```

Router 4 ontvangt de ICMP-pakketten en probeert te antwoorden op 172.16.12.1, maar omdat het geen route naar dit netwerk heeft, mislukt het.

Voeg een statische route aan router 4 toe:

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

Nu hebben beide kanten toegang tot elkaar:

```
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 omlaag

Dit is een situatie waar de interface niet meer werkt. In dit volgende voorbeeld is een poging om router4 van Router1 te pingelen:

```
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)

Aangezien de routing correct is, voert u een stapsgewijze probleemoplossing uit voor het probleem. Probeer router2 te pingen:

```
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

Van het vorige voorbeeld, is het probleem tussen Router2 en Router3. Eén mogelijkheid is dat de seriële interface op Router3 is gesloten:

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

```
Serial0 172.16.3.34 YES manual up up
```

```
Serial1 10.0.3.23 YES manual administratively down down
```

Dit kan eenvoudig worden opgelost:

```
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
```

## Opdracht Toegangslijst

In dit scenario mag alleen Telnet-verkeer router 4 via interface Serial0 invoeren.

```
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

Probeer router4 te pingen:

```
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

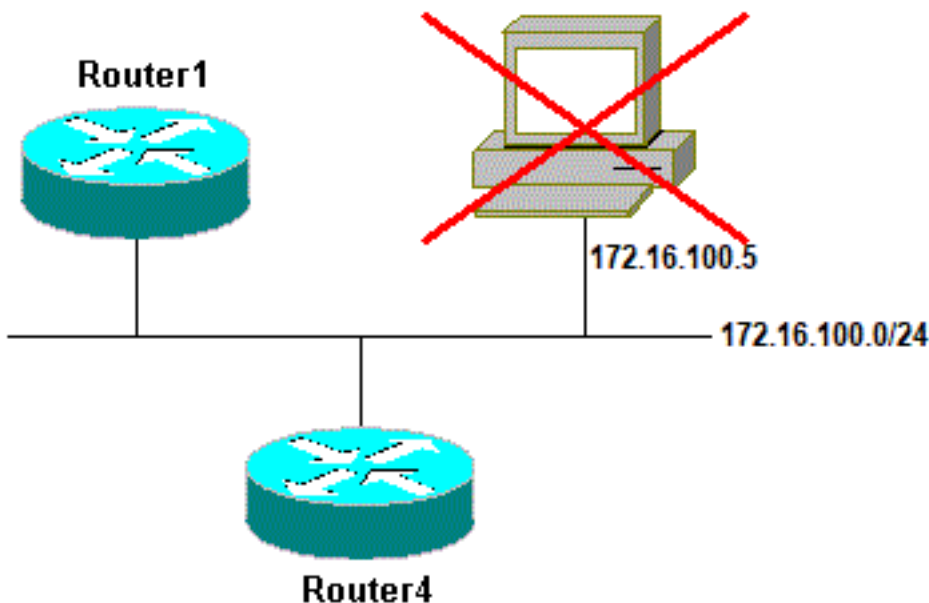
```

Aan het eind van een **toegang-lijst** bevel is er altijd impliciet **ontkennen allen**. Dit betekent dat de pakketten ICMP die Serial 0 interface op Router4 ingaan worden ontkend, en router 4 verzendt een "administratief verboden onbereikbaar"bericht ICMP naar de bron van het originele pakket zoals aangetoond in **debug** bericht. De oplossing is deze lijn toe te voegen in de opdracht **toeganglijst**:

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

## Probleem met Address Resolution Protocol (ARP)

In dit scenario is dit de Ethernet-verbinding:



*Probleem met adresoplossing*

```
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#

```

In dit voorbeeld, pingelt werkt niet wegens "ontbroken inkapselingsbericht". Dit betekent dat de router weet op welke interface het het pakket moet verzenden maar weet niet hoe het te doen. In dit geval moet u begrijpen hoe het Address Resolution Protocol (ARP) werkt.

ARP is een protocol dat wordt gebruikt om Layer 2-adres (MAC-adres) toe te wijzen aan een Layer 3-adres (IP-adres). U kunt dit controleren met de **show arp** opdracht:

```

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

```

Ga terug naar het probleem "inkapseling mislukt", maar schakel deze keer de **debug arp**-opdracht in:

```

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)

```

De vorige output toont aan dat Router4 pakketten uitzendt en hen naar het Ethernet uitzendingsadres FFFF.FFFF.FFFF verzendt. Hier betekent de 0000.000.0000 dat router4 zoekt naar het MAC-adres van de bestemming 172.16.100.5. Aangezien het het MAC-adres niet kent terwijl ARP in dit voorbeeld wordt gevraagd, gebruikt het 0000.000.000 als plaatsaanduiding in de

uitzendingframes verzonden uit interface Ethernet 0 en vraagt welk MAC-adres overeenkomt met 172.16.100.5. Als er geen antwoord is, komt het MAC-adres overeen met 170 het IP-adres in de arp-uitvoer van de **show** wordt als onvolledig gemarkeerd:

```
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
```

Na een vooraf bepaalde periode wordt deze onvolledige invoer uit de ARP-tabel gewist. Zolang het adres van MAC niet in de ARP lijst is, pingelt ontbreekt als resultaat van "ontbroken inkapseling".

## Vertraging

Als u standaard binnen twee seconden geen antwoord van het externe einde ontvangt, zal de ping mislukken:

```
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)
```

Op netwerken met een langzame verbinding of een lange vertraging, zijn twee seconden niet genoeg. U kunt dit gebrek met uitgebreid veranderen pingelt:

```
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
```

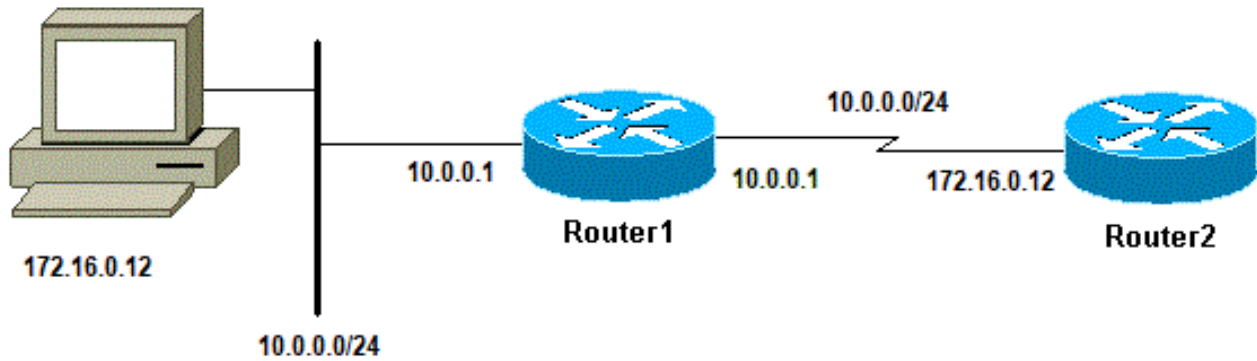
Zie [Uitgebreide ping](#) en [Uitgebreide traceroute-opdrachten begrijpen voor](#) meer informatie over [de uitgebreide ping-opdracht](#).

In het vorige voorbeeld, toen de onderbreking werd verhoogd ping succesvol was.

**Opmerking:** De gemiddelde retourtijd is meer dan twee seconden.

## Correct bronadres

Dit voorbeeld is een veelvoorkomend scenario:



Correct bronadres

Voeg een LAN-interface aan router 1 toe:

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

Van een station op LAN, kunt u router1 pingen. Van Router1 kunt u router2 pingen. Maar van een station op LAN, kunt u geen router2 pingen.

Van Router1, kunt u Router2 pingen omdat, door gebrek, u het IP adres van de uitgaande interface als bronadres in uw pakket ICMP gebruikt. Router2 heeft geen informatie over dit nieuwe LAN. Als het moet reageren op een pakket van dit netwerk, weet het niet hoe het te behandelen.

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

**Waarschuwing:** Wanneer de opdracht **debug ip-pakket** op een productierouterie wordt gebruikt, kan dit een hoog CPU-gebruik veroorzaken. Dit kan leiden tot een ernstige verslechtering van de prestaties of een netwerkstoring.

```
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
```

Het vorige uitvoervoorbeeld werkt omdat het bronadres van het verzonden pakket 172.16.12.1 is. Om een pakket vanuit het LAN te simuleren, moet u een uitgebreid ping gebruiken:

```
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)

```

Dit keer is het bronadres 10.0.0.1 en werkt het niet. De pakketten worden verzonden maar geen reactie wordt ontvangen. Om deze kwestie te bevestigen, voeg een route aan 10.0.0.0 in Router2 toe. De basisregel is dat het ingevoegde apparaat moet ook weten hoe te om het antwoord naar de bron van te verzenden pingelen.

## Wachtrij voor hoge invoer

Wanneer een pakket de router ingaat, probeert de router het op onderbrekingsniveau door:sturen. Als een gelijke niet in een aangewezen geheim voorgeheugenlijst kan worden gevonden, wordt het pakket een rij gevormd in de inputrij van de inkomende te verwerken interface. Sommige pakketten worden altijd verwerkt, maar met de juiste configuratie en in stabiele netwerken, moet de snelheid van verwerkte pakketten nooit de invoerwachtrij verstoppen. Als de invoerwachtrij vol is, wordt het pakket verbroken.

Hoewel de interface omhoog is, en u kunt niet het apparaat pingelen toe te schrijven aan de hoge dalingen van de inputrij. U kunt de inputdalingen met het bevel van de **showinterface** controleren.

```

Router1#show interface Serial0/0/0

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

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

```

!--- Output suppressed

Zoals gezien vanaf de uitvoer is de Input Queue Drop hoog. Raadpleeg [Problemen oplossen bij Invoerwachtrijdruppels en Uitvoerwachtrijdruppels](#) om problemen op te lossen bij het invoeren/uitzetten van wachtrijen.

## De opdracht Traceroute

Het **traceroute** bevel wordt gebruikt om de routes te ontdekken die de pakketten eigenlijk nemen wanneer zij naar hun bestemming reizen. Het apparaat (bijvoorbeeld een router of een PC) stuurt een reeks UDP-datagrammen (User Datagram Protocol) naar een ongeldig poortadres op de externe host.

Drie datagrammen worden verzonden, elk met een Time-To-Live (TTL) veldwaarde ingesteld op één. De waarde van TTL van 1 veroorzaakt het datagram aan "onderbreking" zodra het de eerste router in de weg raakt; deze router reageert vervolgens met een ICMP Time Exceeded Message (TEM) dat aangeeft dat het datagram is verlopen.

Er worden nu nog drie UDP-berichten verzonden, elk met de TTL-waarde ingesteld op 2, waardoor de tweede router ICMP TEM's teruggeeft. Dit proces gaat door tot de pakketten de andere bestemming daadwerkelijk bereiken. Aangezien deze datagrammen proberen om tot een ongeldige poort op de doelhost toegang te krijgen, worden ICMP-poort onbereikbare berichten geretourneerd en wordt een onbereikbare poort aangegeven; deze gebeurtenis geeft het Traceroute-programma aan dat het is voltooid.

Het doel achter dit is om de bron van elk Overschrijdend Bericht van de Tijd te registreren ICMP om een spoor van de weg te verstrekken het pakket nam om de bestemming te bereiken.

```
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
```

Dit is de eerste opeenvolging van pakketten wordt verzonden met een TTL=1. De eerste router, in dit geval Router2 (172.16.0.12), laat vallen het pakket, en verzendt terug naar de bron (172.16.12.1) een type=11 ICMP bericht. Dit komt overeen met het bericht "Tijd overschreden".

```
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

```

Het zelfde proces komt voor router3 (10.0.3.23) met een 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

```

Met een TTL=3, wordt Router4 eindelijk bereikt. Dit keer, aangezien de poort niet geldig is, stuurt router4 een ICMP bericht terug naar Router1 met type=3, een Bestemming Onbereikbaar Bericht, en code=3 dat poort onbereikbaar betekent.

De volgende lijst maakt een lijst van de tekens die in de opdrachtoutput van **traceroute** kunnen verschijnen.

### IP-traceroute teksttekens

karakter	Beschrijving
nn msec	Voor elke knoop, de round-trip tijd in milliseconden voor het gespecificeerde aantal sondes
*	Time-out bij sonde
A	Administratief verboden (bijvoorbeeld toegangslijst)
Q	Bron demping (bestemming te druk)
I	Door gebruiker onderbroken test
U	Poort onbereikbaar
H	Host onbereikbaar
N	Netwerk onbereikbaar
P	Protocol onbereikbaar
T	Timeout
?	Onbekend pakkettype

## Prestaties

U kunt de round-trip tijd (RTT) verkrijgen met de opdrachten **ping** en **traceroute**. Dit is de tijd die nodig is om een echopakket te verzenden en een antwoord terug te krijgen. Dit kan een ruw idee geven van de vertraging op de link. Deze cijfers zijn echter niet precies genoeg om te worden

gebruikt voor de evaluatie van de prestaties.

Wanneer een pakketbestemming de router zelf is, moet dit pakket procesgeschakeld zijn. De processor moet de informatie uit dit pakket verwerken en een antwoord terugsturen. Dit is niet het belangrijkste doel van een router. Per definitie wordt een router gebouwd om pakketten te routeren. Een beantwoord ping wordt aangeboden als een best-performance service.

Om dit te illustreren, is dit een voorbeeld van pingelt van Router1 aan 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
```

De RTT is ongeveer vier milliseconden. Nadat u sommige procesintensieve functies op Router2 hebt ingeschakeld, probeert u Router2 te pingen van Router1.

```
Router1#ping 172.16.0.12
```

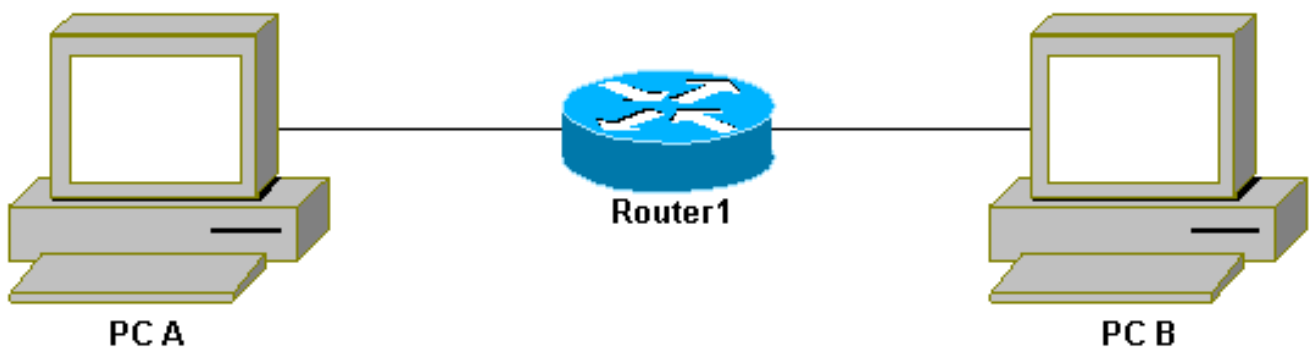
```
Type escape sequence to abort.
```

```
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
```

De RTT is hier dramatisch gestegen. Router2 is vrij bezig, en de prioriteit is niet om te antwoorden pingelt. Een betere manier om routerprestaties te testen is met verkeer dat door de router gaat.



*Verkeer door de router*

Het verkeer wordt dan snel-geschakeld en behandeld door de router met de hoogste prioriteit. Het basisnetwerk illustreert dit:



*netwerk 3 routers*

*Basis*

Ping router3 van router1:



```
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
```

Het verkeer gaat door Router2 en is nu snel-geschakeld. Schakel de procesintensieve optie in op Router2:

```
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
```

Er is bijna geen verschil. Dit is omdat, op Router2, de pakketten nu op onderbrekingsniveau worden behandeld.

## Gebruik de opdracht Debug

Voordat u **debug**-opdrachten gebruikt, raadpleegt u [Belangrijke informatie over debug-opdrachten](#)

Verschillende **debug** commando's die in dit artikel gebruikt worden tonen wat er gebeurt als een **ping** of **traceroute** commando gebruikt wordt. Met deze opdrachten kunt u problemen oplossen. In een productieomgeving moeten debugs echter voorzichtig worden gebruikt. Als je CPU niet krachtig is, of als je veel proces-switched pakketten hebt, kunnen ze je apparaat makkelijk tot stilstand brengen. Er zijn een paar manieren om het effect van te minimaliseren **zuiver** bevel op de router. Eén manier is om toegangslijsten te gebruiken om het specifieke verkeer dat u wilt controleren te beperken.

Hierna volgt een voorbeeld:

```
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)
```

Met deze configuratie, drukt Router4 slechts debug bericht dat toegang-lijst 150 aanpast. Pingel van Router1 veroorzaakt dit bericht om te tonen:

```
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
```

Het antwoord op het probleem komt niet van Router4 omdat deze pakketten niet de toegangslijst aanpassen. Om ze te zien, voeg toe:

```
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
```

## Resultaten:

```
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
```

Een andere manier om het effect van het **debug** bevel te verminderen is te bufferen debug berichten en hen te tonen met het bevel van het **showlogboek** zodra debug is uitgezet:

```
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

De opdrachten **ping** en **traceroute** zijn nuttige hulpprogramma's die u kunt gebruiken om problemen met netwerktoegang op te lossen. Ze zijn ook heel makkelijk te gebruiken. Deze twee opdrachten worden veel gebruikt door netwerkengineers.

## Gerelateerde informatie

- [Begrijp de uitgebreide pings- en uitgebreide tracerouteopdrachten](#)
- [Technische ondersteuning – Cisco Systems](#)

## Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document ([link](#)) te raadplegen.