Ping- und Traceroute-Befehle verstehen

Inhalt

Einleitung

In diesem Dokument wird die Verwendung der Befehle **ping** und **traceroute** auf Cisco Routern beschrieben.

Voraussetzungen

Anforderungen

Es gibt keine spezifischen Anforderungen für dieses Dokument.

Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardware-Versionen beschränkt.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle verstehen.

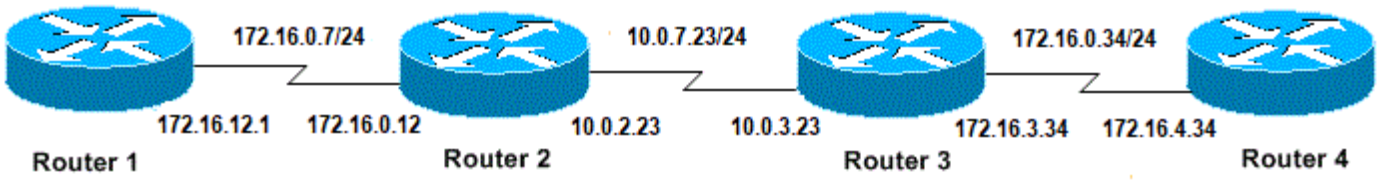
Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter Cisco Technical Tips Conventions (Technische Tipps von Cisco zu Konventionen).

Hintergrundinformationen

> **Anmerkung:** Jeder auf einem Produktions-Router verwendete **Debug**-Befehl kann schwerwiegende Probleme verursachen. Lesen Sie den Abschnitt Verwenden des Debug-Befehls, bevor Sie **Debug-**Befehle ausgeben.

In diesem Dokument wird diese Basiskonfiguration für Beispiele in diesem Artikel verwendet:



Basiskonfiguration von IPs und Routern

Der Befehl „ping“

Der Befehl **ping** ist eine gängige Methode zur Fehlerbehebung bei der Barrierefreiheit für Geräte. Es verwendet eine Reihe von ICMP-Echo-Nachrichten (Internet Control Message Protocol), um Folgendes zu ermitteln:

- Legt fest, ob ein Remote-Host aktiv oder inaktiv ist.
- Die Round-Trip-Verzögerung für die Kommunikation mit dem Host.
- Paketverlust:

Der **Ping**-Befehl sendet zunächst ein Echo-Anforderungspaket an eine Adresse und wartet dann auf eine Antwort. Der Ping-Test ist nur erfolgreich, wenn:

- die Echo-Anfrage beim Ziel eingeht und
- Das Ziel kann innerhalb einer vorgegebenen Zeit, die als Timeout bezeichnet wird, eine Echo-Antwort an die Quelle erhalten. Der Standardwert dieses Timeouts für Cisco Router beträgt zwei Sekunden.

Der TTL-Wert eines **Ping-**Pakets kann nicht geändert werden.

Im folgenden Codebeispiel wird der Befehl **ping** nach dem Aktivieren des Befehls **debug ip packet detail** veranschaulicht.

> **Warnung:** Wenn der Befehl **debug ip packet detail** auf einem Produktionsrouter verwendet wird, kann dies zu einer hohen CPU-Auslastung führen. Dies kann zu erheblichen Leistungseinbußen oder Netzwerkausfällen führen.

```
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.
```

Mögliche ICMP-Typwerte

ICMP-Typ	Literal
0	Echo-Antwort
3	destination unreachable code 0 = net unreachable 1 = host unreachable 2 = protocol unreachab port unreachable 4 = fragmentierung erforderlich und DF set 5 = source route failed
4	Quellenlöschung
5	Umleitungscode 0 = Umleitungsdatagramme für das Netzwerk 1 = Umleitungsdatagramme für c Host 2 = Umleitungsdatagramme für den Diensttyp und Netzwerk 3 = Umleitungsdatagramme fü Diensttyp und den Host
6	Alternativadresse
8	Echo
9	Router-Advertisement
10	Router-Anforderung
11	Zeitüberschreitung Code 0 = Lebensdauer bei Übertragung überschritten 1 = Zeitüberschreitung Reassemblierung des Fragments
12	Parameterproblem
13	Zeitstempelanforderung
14	Zeitstempel-Antwort
15	Informationsanfrage
16	Informationsantwort
17	Maskenanforderung
18	Maskenantwort
31	Konvertierungsfehler
32	mobile Umleitung

Mögliche Ausgabezeichen aus der Ping-Funktion

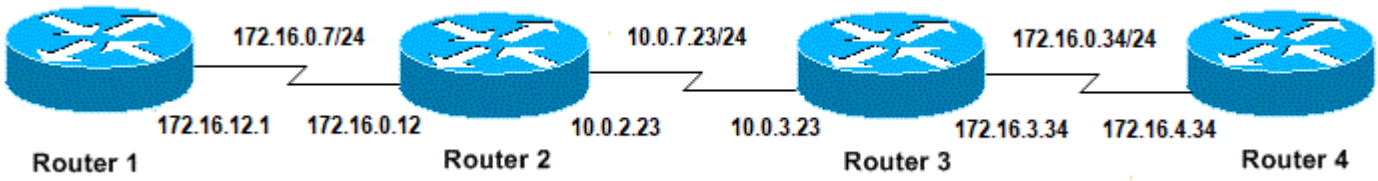
Zeichen	Beschreibung
!	Jeder Ausrufezeichen steht für den Erhalt einer Antwort.
.	Jeder Punkt gibt an, dass der Netzwerkserver beim Warten auf eine Antwort abgelauf ist.
U	Es wurde eine PDU für den nicht erreichbaren Zielfehler empfangen.
F	Quell-Quench (Ziel zu besetzt).
M	Konnte nicht fragmentiert werden.
?	Unbekannter Pakettyp.
u.	Paketlebensdauer überschritten.

Ping kann nicht gesendet werden

Wenn Sie nicht erfolgreich an eine IP-Adresse **pingen** können, berücksichtigen Sie die in diesem Abschnitt aufgeführten Ursachen.

Router-Problem

Im Folgenden finden Sie Beispiele für erfolglose Ping-Versuche, die das Problem ermitteln können, und Maßnahmen zur Behebung des Problems. Dieses Beispiel zeigt ein Netzwerktopologie-Diagramm:



Router-Probleme

```
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
!
```

Versuchen Sie, Router4 von Router1 aus zu 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)
```

Ergebnisse:

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

> **Warnung:** Wenn der Befehl **debug ip packet** auf einem Produktions-Router verwendet wird, kann dies zu einer hohen CPU-Auslastung führen. Dies kann zu erheblichen Leistungseinbußen oder Netzwerkausfällen führen.

```
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)
```

Da auf Router1 keine Routing-Protokolle ausgeführt werden, weiß er nicht, wohin sein Paket gesendet werden soll, und verursacht eine Nachricht, die nicht weitergeleitet werden kann.

Statische Route zu Router1 hinzufügen:

```
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
```

Ergebnisse:

```
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
```

Untersuchen Sie, was auf Router2 falsch ist:

```
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 hat seine Pakete korrekt an Router2 gesendet, aber Router2 weiß nicht, wie er auf die Adresse 172.16.4.34 zugreift. Router2 sendet eine "unreachable ICMP"-Nachricht an Router1 zurück.

Routing Information Protocol (RIP) auf Router 2 und Router 3 aktivieren:

```
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
```

Ergebnisse:

```
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 sendet Pakete an Router4, aber Router4 sendet keine Antwort zurück.

Mögliches Problem auf 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 empfängt die ICMP-Pakete und versucht, eine Antwort auf 172.16.12.1 zu geben. Da er jedoch keine Route zu diesem Netzwerk hat, schlägt er fehl.

Statische Route zu Router4 hinzufügen:

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

Beide Seiten können nun auf einander zugreifen:

```
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
```

Schnittstelle ausgefallen

In diesem Fall funktioniert die Schnittstelle nicht mehr. Im folgenden Beispiel wird versucht, einen Ping an Router4 von Router1 aus zu senden:

```
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)
```

Da das Routing korrekt ist, führen Sie eine schrittweise Fehlerbehebung durch. Versuchen Sie, einen Ping an Router2 zu senden:

```
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
```

Aus dem vorherigen Beispiel ergibt sich das Problem zwischen Router2 und Router3. Eine Möglichkeit besteht darin, dass die serielle Schnittstelle auf Router3 heruntergefahren wurde:

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

Dies ist einfach zu beheben:

```
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
```

Zugriffslisten-Befehl

In diesem Szenario darf nur Telnet-Datenverkehr über die Schnittstelle Serial0 auf Router4 gelangen.

```
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
```

Versuchen Sie, einen Ping an Router4 zu senden:

```
Router1#ping 172.16.4.34

Type escape sequence to abort.
```

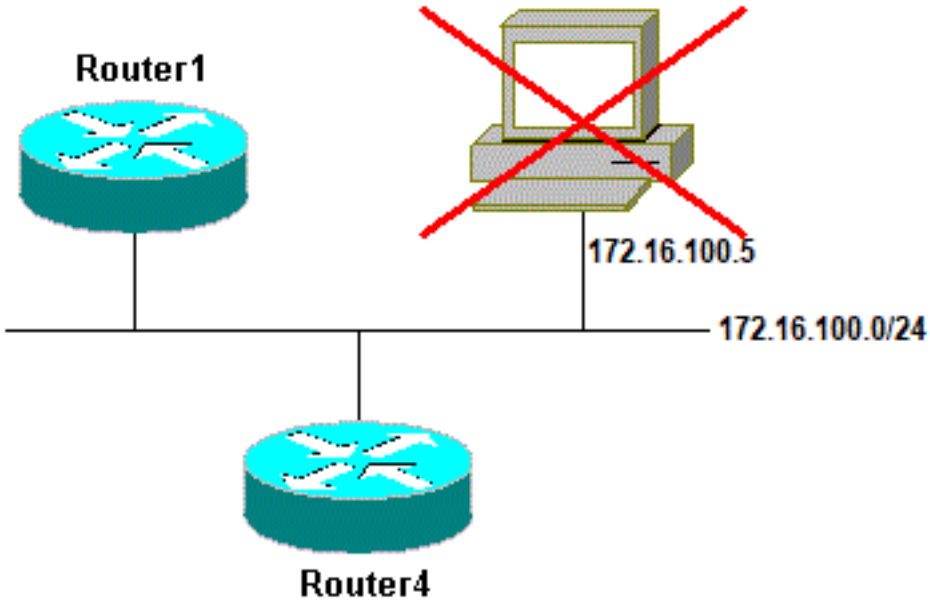
```
Sending 5, 100-byte ICMP Echos to 172.16.4.34, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

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
```

Am Ende eines **Zugriffslistenbefehls** steht immer ein impliziter Befehl **deny all.** Dies bedeutet, dass die ICMP-Pakete, die auf Router4 in die serielle 0-Schnittstelle gelangen, abgelehnt werden, und Router 4 sendet eine ICMP-Meldung "administratively verbot unreachable" an die Quelle des ursprünglichen Pakets, wie in der **Debug-**Meldung dargestellt. Die Lösung besteht darin, diese Zeile dem Befehl **access-list** hinzuzufügen:

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

ARP-Problem (Address Resolution Protocol)

In diesem Szenario ist dies die Ethernet-Verbindung:



Problem mit dem Address Resolution Protocol

```
Router4#ping 172.16.100.5

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

```
Jan 20 17:04:05.167: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:05.171: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:07.167: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:07.171: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:09.175: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:09.183: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:11.175: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:11.179: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:13.175: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:13.179: IP: s=172.16.100.4 (local), d=172.16.100.5 (Ethernet0), len 100,
  encapsulation failed.
Success rate is 0 percent (0/5)
Router4#
```

In diesem Beispiel funktioniert der Ping-Befehl aufgrund der Meldung "Encapsulation failed" (Kapselung fehlgeschlagen) nicht. Das bedeutet, dass der Router weiß, über welche Schnittstelle er das Paket senden muss, aber nicht weiß, wie er das tun soll. In diesem Fall müssen Sie wissen, wie das Address Resolution Protocol (ARP) funktioniert.

ARP ist ein Protokoll, das verwendet wird, um die Layer-2-Adresse (MAC-Adresse) einer Layer-3-Adresse (IP-Adresse) zuzuordnen. Sie können dies mit dem Befehl **show arp** überprüfen:

```
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
```

Kehren Sie zum Problem "Kapselung fehlgeschlagen" zurück, aktivieren Sie jedoch diesmal den Befehl **debug arp**:

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

Router4#ping 172.16.100.5

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

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

Die vorherige Ausgabe zeigt, dass Router4 Pakete sendet und an die Ethernet-Broadcast-Adresse FFFF.FFFF.FFFF sendet. Hier bedeutet "0000.0000.0000", dass Router4 nach der MAC-Adresse des Ziels 172.16.100.5 sucht. Da der Router die MAC-Adresse nicht kennt, während in diesem Beispiel der ARP angefordert wird, wird "0000.000.0000" verwendet. 0 als Platzhalter in den Broadcast-Frames, die von der Schnittstelle Ethernet 0 gesendet werden, und fragt, welche MAC-Adresse 172.16.100.5 entspricht. Wenn keine Antwort erfolgt, wird die MAC-Adresse, die der IP-Adresse in der Ausgabe von **show arp** entspricht, als unvollständig markiert:

```
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
```

Nach einem vorbestimmten Zeitraum wird dieser unvollständige Eintrag aus der ARP-Tabelle gelöscht. Solange sich die MAC-Adresse nicht in der ARP-Tabelle befindet, schlägt der Ping-Test aufgrund von "Encapsulation failed" fehl.

Verzögerung

Standardmäßig schlägt der Ping fehl, wenn Sie innerhalb von zwei Sekunden keine Antwort von der Gegenstelle erhalten:

```
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)
```

In Netzwerken mit langsamer Verbindung oder großer Verzögerung reichen zwei Sekunden nicht aus. Sie können diese Standardeinstellung mit einem erweiterten Ping ändern:

```
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
```

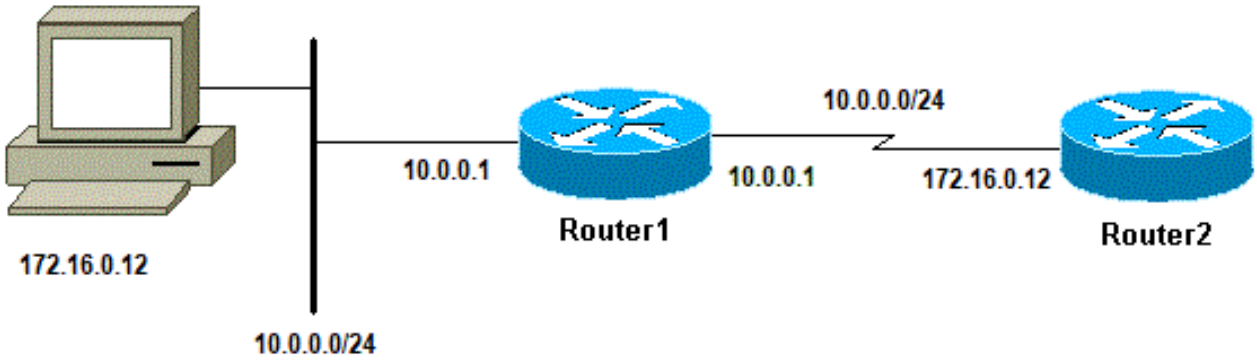
Weitere Informationen zum erweiterten Ping-Befehl finden Sie unter Verstehen der Befehle Erweiterter Ping und Erweiterte Traceroute .

Im vorherigen Beispiel war der Ping erfolgreich, wenn die Zeitüberschreitung erhöht wurde.

> **Anmerkung:** Die durchschnittliche Umlaufzeit beträgt mehr als zwei Sekunden.

Richtige Quelladresse

Dieses Beispiel ist ein gängiges Szenario:



Richtige Quelladresse

Hinzufügen einer LAN-Schnittstelle auf Router1:

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

Von einer Station im LAN können Sie Router1 pingen. Von Router1 können Sie Router2 pingen. Von einer Station im LAN aus können Sie Router2 jedoch nicht pingen.

Von Router1 können Sie Router2 anpingen, da Sie standardmäßig die IP-Adresse der ausgehenden Schnittstelle als Quelladresse in Ihrem ICMP-Paket verwenden. Router2 verfügt über keine Informationen zu diesem neuen LAN. Wenn er auf ein Paket aus diesem Netzwerk antworten muss, weiß er nicht, wie er damit umgehen soll.

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

> **Warnung:** Wenn der Befehl **debug ip packet** auf einem Produktions-Router verwendet wird, kann dies zu einer hohen CPU-Auslastung führen. Dies kann zu erheblichen Leistungseinbußen oder Netzwerkausfällen führen.

```
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
```

Das vorherige Ausgabebeispiel funktioniert, da die Quelladresse des gesendeten Pakets 172.16.12.1 lautet. Um ein Paket aus dem LAN zu simulieren, müssen Sie einen erweiterten Ping verwenden:

```
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)
```

Diesmal ist die Quelladresse 10.0.0.1 und funktioniert nicht. Pakete werden gesendet, aber es wird keine Antwort empfangen. Um dieses Problem zu beheben, fügen Sie in Router2 eine Route zu 10.0.0.0 hinzu. Die Grundregel lautet, dass das gepingte Gerät auch wissen muss, wie es die Antwort an die Ping-Quelle sendet.

Hohe Verluste in der Eingabewarteschlange

Wenn ein Paket den Router erreicht, versucht der Router, es auf der Unterbrechungsebene weiterzuleiten. Wenn keine Übereinstimmung in einer geeigneten Cache-Tabelle gefunden werden kann, wird das Paket in die Eingangswarteschlange der Eingangsschnittstelle gestellt, die verarbeitet werden soll. Einige Pakete werden immer verarbeitet, aber mit der entsprechenden Konfiguration und in stabilen Netzwerken darf die Rate der verarbeiteten Pakete die Eingangswarteschlange niemals überlasten. Wenn die Eingangswarteschlange voll ist, wird das Paket verworfen.

Obwohl die Schnittstelle aktiv ist, können Sie das Gerät aufgrund hoher Verluste in der Eingabewarteschlange nicht pingen. Mit dem Befehl **show interface** können Sie überprüfen, ob Eingaben verloren gehen.

```
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
```

Wie aus der Ausgabe ersichtlich, ist der Wert für "Input Queue Drop" hoch. Weitere Informationen zur Fehlerbehebung bei Verwerfungen von Eingangs- und Ausgangswarteschlangen finden Sie unter Verwerfen von Eingangs- und Ausgangswarteschlangen.

Der Traceroute-Befehl

Mit dem Befehl **traceroute** werden die Routen ermittelt, die Pakete tatsächlich nehmen, wenn sie an ihr Ziel gelangen. Das Gerät (z. B. ein Router oder ein PC) sendet eine Sequenz von User Datagram Protocol (UDP)-Datagrammen an eine ungültige Port-Adresse am Remote-Host.

Es werden drei Datagramme gesendet, von denen jedes einen TTL-Feldwert (Time-To-Live) aufweist, der auf eins festgelegt ist. Der TTL-Wert 1 bewirkt, dass das Datagramm eine Zeitüberschreitung verursacht, sobald es am ersten Router im Pfad ankommt; Dieser Router antwortet dann mit einer ICMP-Zeitüberschreitungsmeldung (Time Exceeded Message, TEM), die angibt, dass das Datagramm abgelaufen ist.

Es werden nun drei weitere UDP-Nachrichten mit einem auf 2 gesetzten TTL-Wert gesendet, woraufhin der zweite Router ICMP-TEMs zurückgibt. Dieser Prozess wird fortgesetzt, bis die Pakete tatsächlich das andere Ziel erreichen. Da diese Datagramme versuchen, auf einen ungültigen Port auf dem Zielhost zuzugreifen, werden Meldungen über nicht erreichbaren ICMP-Port zurückgegeben, die auf einen nicht erreichbaren Port hinweisen. Dieses Ereignis signalisiert dem Traceroute-Programm, dass es beendet ist.

Der Zweck dahinter ist die Aufzeichnung der Quelle jeder ICMP-Meldung über eine Zeitüberschreitung, um eine Nachverfolgung des Pfads bereitzustellen, über den das Paket zum Ziel gelangt ist.

```
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
```

Dies ist die erste Paketsequenz, die mit einem TTL=1 gesendet wird. Der erste Router, in diesem Fall Router2 (172.16.0.12), verwirft das Paket und sendet eine ICMP-Nachricht vom Typ 11 an die Quelle (172.16.12.1) zurück. Dies entspricht der Meldung Zeitüberschreitung.

```
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
```

Derselbe Prozess tritt bei Router3 (10.0.3.23) mit einem TTL=2 auf:

```
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
```

Mit einem TTL=3 ist schließlich Router4 erreicht. Da der Port ungültig ist, sendet Router4 diesmal eine ICMP-Nachricht vom Typ 3, eine Destination Unreachable Message und Code=3 zurück an Router1, was bedeutet, dass Port nicht erreichbar ist.

In der nächsten Tabelle sind die Zeichen aufgeführt, die in der Ausgabe des Befehls **traceroute** angezeigt werden können.

IP-Traceroute-Textzeichen

Zeichen	Beschreibung
nn ms	Die Round-Trip-Zeit in Millisekunden für die angegebene Anzahl von Tests fü jeden Knoten
*	Zeitüberschreitung bei der Überprüfung
A	Administrator verboten (z. B. Zugriffsliste)
F	Quellenlöschung (Ziel zu belegt)
I	Unterbrechungstest
U	Port nicht erreichbar
H	Host nicht erreichbar
N	Netzwerk nicht erreichbar
P	Protokoll nicht erreichbar
T	Zeitüberschreitung
?	Unbekannter Pakettyp

Leistung

Die Round-Trip-Zeit (Round Trip Time, RTT) kann mit den Befehlen **ping** und **traceroute ermittelt werden**. Dies ist die Zeit, die benötigt wird, um ein Echo-Paket zu senden und eine Antwort zu erhalten. Dies kann eine grobe Vorstellung der Verzögerung für die Verbindung liefern. Diese Zahlen sind jedoch nicht präzise genug, um für die Leistungsbewertung verwendet werden zu können.

Wenn ein Paketziel der Router selbst ist, muss dieses Paket prozessgesteuert werden. Der Prozessor muss die Informationen aus diesem Paket verarbeiten und eine Antwort zurücksenden. Dies ist nicht das Hauptziel eines Routers. Ein Router wurde per Definition für die Weiterleitung von Paketen entwickelt. Ein entgegengenommenes Ping wird als bestmöglicher Service angeboten.

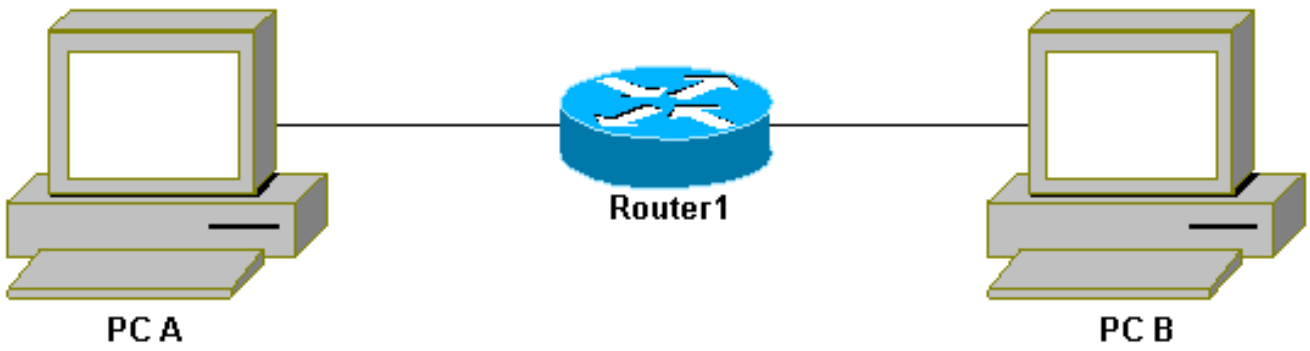
Um dies zu veranschaulichen, ist dies ein Beispiel für einen Ping von Router1 zu Router2:

```
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
```

Der RTT-Wert beträgt ungefähr vier Millisekunden. Nachdem Sie einige prozessintensive Funktionen auf Router2 aktiviert haben, versuchen Sie, Router2 von Router1 aus zu 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 = 24/25/28 ms
```

Die RTT hat hier drastisch zugenommen. Router2 ist sehr beschäftigt, und die Priorität besteht darin, den Ping nicht zu beantworten. Eine bessere Möglichkeit zum Testen der Router-Leistung ist der Datenverkehr, der durch den Router fließt.



Datenverkehr über den Router

Der Datenverkehr erfolgt dann über Fast-Switching und wird vom Router mit der höchsten Priorität verarbeitet. Das Basisnetzwerk veranschaulicht dies:

Grundlegend e Netzwerk-3-Router

Pingen Sie Router3 von 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
```

Der Datenverkehr läuft über Router2 und wird jetzt beschleunigt. Aktivieren Sie die prozessintensive Funktion auf 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
```

Es gibt fast keinen Unterschied. Dies liegt daran, dass die Pakete auf Router2 nun auf Interrupt-Ebene verarbeitet werden.

Verwenden des Befehls Debug

Bevor Sie **Debug**-Befehle verwenden, lesen Sie Wichtige Informationen zu Debug-Befehlen .

Die verschiedenen in diesem Artikel verwendeten **Debug**-Befehle zeigen, was geschieht, wenn ein **Ping-** oder **Traceroute-**Befehl verwendet wird. Diese Befehle können Ihnen bei der Fehlerbehebung helfen. In einer Produktionsumgebung muss das Debuggen jedoch mit Vorsicht eingesetzt werden. Wenn Ihre CPU nicht leistungsstark ist oder Sie viele prozessgesteuerte Pakete haben, können diese Ihr Gerät leicht blockieren. Es gibt mehrere Möglichkeiten, die Auswirkungen des Befehls **debug** auf den Router zu minimieren. Eine Möglichkeit besteht darin, Zugriffslisten zu verwenden, um den bestimmten Datenverkehr einzugrenzen, den Sie überwachen möchten.

Hier ein Beispiel:

```
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)
```

Bei dieser Konfiguration gibt Router4 nur die Debugmeldung aus, die mit der Zugriffsliste 150 übereinstimmt. Ein Ping von Router1 bewirkt, dass diese Meldung angezeigt wird:

```
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
```

Die Antwort auf das Problem kommt nicht von Router4, da diese Pakete nicht mit der Zugriffsliste übereinstimmen. Um sie anzuzeigen, fügen Sie hinzu:

```
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
```

Ergebnisse:

```
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
```

Eine andere Möglichkeit, die Auswirkungen des Befehls **debug** zu verringern, besteht darin, die Debugmeldungen zu puffern und mit dem Befehl **show log** anzuzeigen, sobald der Debugging-Befehl deaktiviert wurde:

```
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
```

Die Befehle **ping** und **traceroute** sind hilfreiche Dienstprogramme, mit denen Sie Probleme mit dem Netzwerkzugriff beheben können. Sie sind auch sehr einfach zu bedienen. Diese beiden Befehle werden von Netzwerktechnikern häufig verwendet.

Zugehörige Informationen

- Kenntnis der Befehle für erweiterten Ping und erweiterte Traceroute
- Technischer Support – Cisco Systems

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.