

ping および traceroute コマンドについて

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[背景説明](#)

[ping コマンド](#)

[ping を実行できません](#)

[ルータの問題](#)

[インターフェイスのダウン](#)

[access-list コマンド](#)

[Address Resolution Protocol \(ARP \) 問題](#)

[遅延](#)

[正しい送信元アドレス](#)

[高入カキュードロップ](#)

[traceroute コマンド](#)

[パフォーマンス](#)

[debug コマンドの使用](#)

[関連情報](#)

はじめに

このドキュメントでは、シスコのルータでの ping および traceroute コマンドの使い方を説明します。

前提条件

要件

このドキュメントに関する固有の要件はありません。

使用するコンポーネント

このドキュメントの内容は、特定のソフトウェアやハードウェアのバージョンに限定されるものではありません。


このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな (デフォルト) 設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認して

ください。

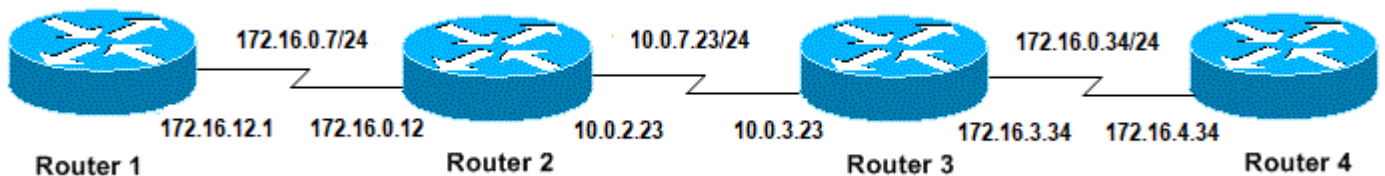
表記法

表記法の詳細については、『シスコ テクニカル ティップスの表記法』を参照してください。

背景説明

 注：実稼動ルータで debug コマンドを有効にすると、重大な問題が発生する場合があります。debug コマンドを発行する前に、「[debug コマンドの使用](#)」の項をお読みください。

このドキュメントでは、たとえばこの記事にあるような基本設定を使用します。



IP とルータの基本設定

ping コマンド

ping コマンドは、デバイスのアクセシビリティのトラブルシューティングに広く使用されている方法です。次に挙げた事項を決定する際に、一連の Internet Control Message Protocol (ICMP; インターネット制御メッセージ プロトコル) Echo メッセージを使用します。

- リモート ホストがアクティブか、非アクティブか
- ホストと通信するために使用するラウンドトリップ遅延。
- パケットロス


ping コマンドはまず、アドレスにエコー要求パケットを送信し、応答を待ちます。ping は、次の場合にだけ成功します。

- エコー要求が宛先に到達する場合。
- 事前定義された時間 (タイムアウト) 内に、送信先がエコー応答を送信元に返すことができた場合。Cisco ルータでは、このタイムアウトのデフォルト値は 2 秒です。

ping パケットの TTL 値は変更できません。

次のコード例は、debug ip packet detail コマンドを有効にした後の ping コマンドを示しています。

 警告：実稼働中のルータで debug ip packet detail コマンドを使用すると、CPU の利用率が

 高くなる可能性があります。これにより、パフォーマンスが大きく低下したり、ネットワークが停止することがあります。

```
<#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.
```

可能な ICMP-type の値

ICMP タイプ	内容
0	echo-reply
3	destination unreachable code 0 = net unreachable 1 = host unreachable 2 = protocol unreachable 3 = port unreachable 4 = fragmentation needed, and DF set 5 = source route failed
4	ソース クエンチ (始点抑制要求)
5	リダイレクト コード 0 = ネットワークのリダイレクト データグラム 1 = ホストのリダ

	イレクト データグラム 2 = タイプ オブ サービスとネットワークのリダイレクト データグラム 3 = タイプ オブ サービスとホストのリダイレクト データグラム
6	代替アドレス
8	エコー
9	ルータ アドバタイズメント
10	ルータ要求
11	時間超過コード 0 = トランジット中の存続時間超過 1 = フラグメント再構成時間超過
12	パラメータ問題
13	timestamp-request
14	タイムスタンプ応答
15	情報要求
16	情報応答
17	マスク要求
18	マスク応答
31	変換エラー
32	モバイル リダイレクト

ping の機能から出力される可能性のある文字

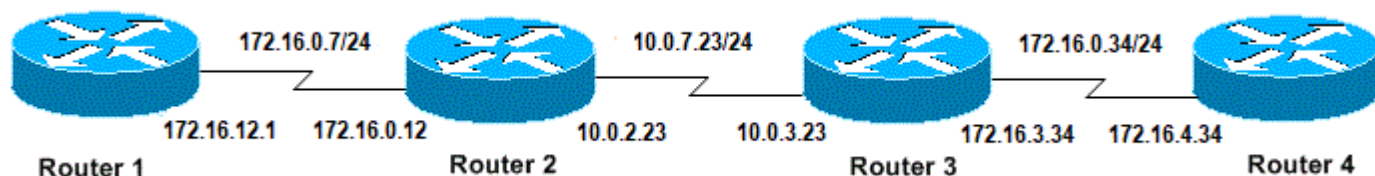
文字	説明
!	各感嘆符 (!) は、応答の受信を示します。
を参照。	ピリオド (.) は、ネットワークサーバーが応答を待機中にタイムアウトしたことを示します。
U	宛先到達不能エラーを表す PDU を受信したことを意味します。
Q	ソースクエンチ (始点抑制要求)。宛先がビジー状態です。
M	フラグメント化できません。
?	パケットタイプが不明です。
&	パケットの存続時間超過

ping を実行できません

特定のアドレスに対して ping を正常に実行できない場合、このセクションでリストしている原因を検討してください。

ルータの問題

次に、ping の不成功、問題の特定、および問題の解決手順の各例について説明します。この例は、次のネットワークトポロジ図を使用して示します。



ルータの問題

```
<#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  
!
```

Router1 から Router4 へ ping を実行してみましょう。

```
<#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)
```


[Results] :

```
<#root>
```

```
Router1#
```

```
debug ip packet
```

```
IP packet debugging is on
```

 警告：実稼働中のルータで debug ip packet コマンドを使用すると、CPU の利用率が高くなる可能性があります。これにより、パフォーマンスが大きく低下したり、ネットワークが停止することがあります。

```
<#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)
```

Router1 ではルーティングプロトコルが実行されていないため、パケットをどこに送信したらいいかわからず、「unroutable」メッセージが表示されます。

Router1 にスタティックルートを追加します。

```
<#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
```

[Results] :

```
<#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
```

Router2 の問題を調べましょう。

```
<#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
```

Router1 は Router2 に正しくパケットを送信しましたが、Router2 はアドレス 172.16.4.34 へのアクセス方法が分かりません。Router2 は Router1 に「unreachable ICMP」メッセージを戻します。

Router2 と Router3 で Routing Information Protocol (RIP) を有効にします。

```
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
```

[Results] :

```
<#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)
```

Router1 は Router4 にパケットを送信しますが、Router4 は応答を返しません。

Router4 で考えられる問題 :

```
<#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
```

Router4 は ICMP パケットを受信し、172.16.12.1 へ応答を返そうとしますが、Router4 にはこのネットワークへのルートがないため、失敗しました。

Router4 にスタティックルートを追加します。

```
<#root>
```

```
Router4(config)#
```

```
ip route 0.0.0.0 0.0.0.0 Serial0
```

これで両サイドがお互いにアクセスできるようになりました。

```
<#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
```

インターフェイスのダウン

これは、インターフェイスが機能を停止するという状況です。次の例では、Router1 から Router4 への ping の実行を試みます。

```
<#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)
```

ルーティングは正しいため、問題の段階的なトラブルシューティングを実行します。Router2 に ping を実行します。

```
<#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
```

前の例から、問題は Router2 と Router3 の間にあることがわかります。Router3 のシリアル インターフェイスがシャットダウンした可能性があります。

```
<#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
```

これは簡単に修復できます。

```
<#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
```

access-list コマンド

このシナリオでは、telnet トラフィックだけがインターフェイス Serial0 を経由して Router4 に入ることができます。

```
<#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
```

Router4 に ping を実行します。

```
<#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
```

access-list コマンドの終わりには、常に暗黙的な「deny all」があります。つまり、Router4の Serial 0 インターフェイスに到着した ICMP パケットは拒否され、Router 4 は debug メッセージに示されているように ICMP 「administratively prohibited unreachable」メッセージを元のパケットの送信元に送信します。これを解決する方法として、access-list コマンドに次の行を追加します。

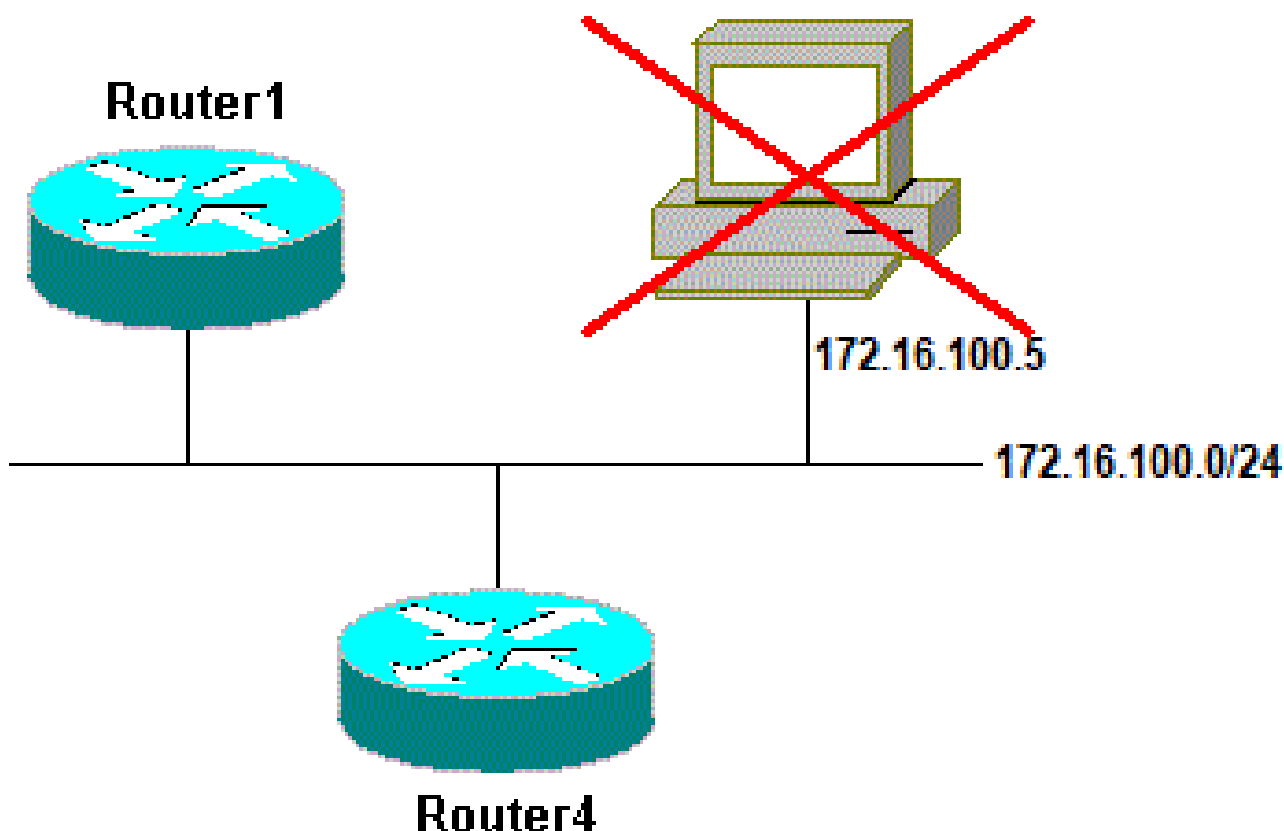
```
<#root>
```

```
Router4(config)#
```

```
access-list 100 permit icmp any any
```

Address Resolution Protocol (ARP) 問題

このシナリオでは、これはイーサネット接続です。



Address Resolution Protocol の問題

```
<#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#
```

この例では、「カプセル化失敗」メッセージが表示されたため、ping が動作しません。つまり、ルータはどのインターフェイスにパケットを送信したらいいかを認識しているにもかかわらず、パケットの送信方法がわからないということです。このケースでは、Address Resolution Protocol (ARP) の動作方法を理解する必要があります。

ARP はレイヤ 2 アドレス (MAC アドレス) をレイヤ 3 アドレス (IP アドレス) にマップするのに使われるプロトコルです。これは show arp コマンドを使ってチェックできます。

```
<#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

「カプセル化失敗」問題に戻りますが、今回は debug arp コマンドを有効にします。

```
<#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)

前の出力は、Router4 がパケットをブロードキャストし、イーサネットブロードキャストアドレス FFFF.FFFF.FFFF に送信していることを示しています。ここでは、0000.0000.0000 は、Router4 が送信先 172.16.100.5 の MAC アドレスを検索していることを意味しています。この例では、当該ルータは ARP 要求の実行中に MAC アドレスを認識していないため、インターフェイスイーサネット 0 から送信されるブロードキャストフレーム内で 0000.0000.0000 をプレースホルダとして使い、172.16.100.5 に対応する MAC アドレスを問い合わせます。応答がない場合、show arp 出力の IP アドレスに対応する MAC アドレスは不完全としてマークされます。

<#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

事前定義した期間が経過した後、この不完全なエントリは ARP テーブルから消去されます。その MAC アドレスが ARP テーブル中不在の場合、「カプセル化失敗」の結果 ping が失敗します。

遅延

デフォルトでは、2 秒以内にリモート エンドから応答を受信しないと、ping は失敗します。

```
<#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)
```

リンクが遅い、または遅延が長いネットワークの場合、2 秒では不十分です。このデフォルト値は、拡張 ping を使って変更できます。

```
<#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
```

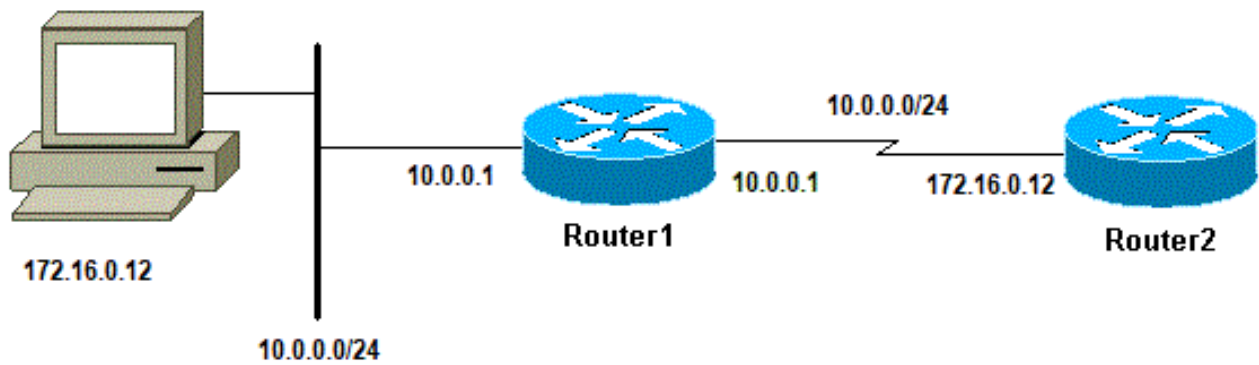
拡張 ping コマンドの詳細については、『[拡張 ping および拡張 Traceroute コマンドについて](#)』を参照してください。

前の例では、タイムアウト値を増やすと ping は成功しました。

 注：ラウンドトリップの平均時間は 2 秒以上です。

正しい送信元アドレス

この例は、一般的なシナリオです。



正しい送信元アドレス

Router1 に LAN インターフェイスを追加します。

```
<#root>
```

```
Router1(config)#
```

```
interface ethernet0
```

```
Router1(config-if)#
```

```
ip address 10.0.0.1 255.255.255.0
```

LAN 上のステーションから、Router1 へ ping することができます。Router1 から Router2 へ ping することができます。しかし、LAN 上のステーションから Router2 へは ping できません。

Router1 からは、Router2 へ ping できます。なぜならば、デフォルトで、ICMP パケット中のソースアドレスとして、発信インターフェイスの IP アドレスを使用するからです。Router2 には、この新しい LAN に関する情報がありません。このネットワークからのパケットに応答する必要がありますが、処理方法がわかりません。

```
<#root>
```

```
Router1#
```

```
debug ip packet
```

```
IP packet debugging is on
```

警告：実稼働中のルータで debug ip packet コマンドを使用すると、CPU の利用率が高くなる可能性があります。これにより、パフォーマンスが大きく低下したり、ネットワークが停止することがあります。

```
<#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
```

前の出力例は、送信されたパケットの送信元アドレスが 172.16.12.1 であるため、機能していません。LAN からのパケットをシミュレートするには、拡張 ping を使用する必要があります。

```
<#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)
```

この場合、発信元アドレスは 10.0.0.1 であり、これは動作していません。パケットは送信されま

すが、応答は得られません。この問題を修正するには、Router2 に 10.0.0.0 へのルートを追加します。基本のルールとして、ping の実行先デバイスは、ping の送信元への応答の送信方法も認識している必要があります。

高入力キュー ドロップ

パケットがルータに到着すると、ルータはそれを割り込みレベルで転送しようとしています。該当するキャッシュテーブル内に一致するものが見つからない場合、そのパケットはプロセス処理のために入インターフェイスの入力キューに入れます。一部のパケットは常にプロセス処理されていますが、設定が適切でネットワークが安定していれば、プロセス処理されるパケットで入力キューがいっぱいになることはありません。入力キューがいっぱいになると、パケットは廃棄されます。

インターフェイスは稼働していますが、入力キューのドロップが多いことが原因でデバイスを ping できません。show interface コマンドで入力ドロップを確認できます。

```
<#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 suppressed
```

出力からわかるように、Input Queue Drop が high になっています。入力/出力キュードロップのトラブルシューティングについては、『[入力キュードロップと出力キュードロップのトラブルシューティング](#)』を参照してください。

traceroute コマンド

traceroute コマンドは、パケットが宛先に向かうときに実際に通過するルート検出するのに使用されます。デバイス (たとえば、ルータまたは PC) は、User Datagram Protocol (UDP; ユーザデータグラムプロトコル) データグラムのシーケンスを、リモートホストの無効ポートアドレスに送信します。

3つのデータグラムが送られ、それぞれの Time-To-Live (TTL; 生存可能時間) フィールド値が 1 に設定されています。TTL の値を 1 にすることで、データグラムはパス内の最初のルータにヒットするとすぐに「タイムアウト」します。このルータは次に、データグラムが期限切れになったことを意味する ICMP の Time Exceeded Message (TEM) メッセージを返します。

次に、別の 3つの UDP メッセージが送信され、それぞれの TTL 値は 2 に設定されているため、2番目のルータは ICMP TEM を返します。このプロセスは、パケットが実際に他方の宛先に到達するまで続きます。これらのデータグラムは宛先ホストの無効なポートにアクセスしようとしているため、ICMP のポート到達不能メッセージが返され、到達不能なポートが示されます。このイベントによって、その終了を伝える信号が traceroute プログラムに送られます。

ここでの目的は、各 ICMP TEM の送信元を記録し、宛先に到達するのにパケットがとったパスのトレースを提供することです。

```
<#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
```

これは TTL=1 を使って送信したパケットの最初のシーケンスです。最初のルータ (このケースでは Router2 (172.16.0.12)) はパケットをドロップし、送信元 (172.16.12.1) に type=11 ICMP メッセージを送り返します。これは、Time Exceeded Message (TEM) に相当します。

<#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
```

TTL=2 の場合には Router3 (10.0.3.23) で同じプロセスが発生します。

<#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
```

TTL=3 では、最終的に Router4 に到達します。このとき、ポートは有効ではないため、Router4 は Router1 に type=3 (Destination Unreachable Message)、code=3 (Port Unreachable) の ICMP メッセージを返信します。

次の表は、traceroute コマンド出力に表示される可能性がある文字を一覧にしたものです。

IP traceroute テキスト文字

文字	説明
nn msec	各ノードについての、指定プローブ数に対するラウンドトリップ時間 (ミリ秒)
*	プローブがタイムアウト
A	管理上の理由による禁止 (例 : , access-list)
Q	ソースクエンチ (始点抑制要求)。宛先がビジー状態
I	ユーザ割り込みテスト
U	ポートが到達不能
H	ホストが到達不能
N	ネットワークが到達不能
P	プロトコル到達不能
T	[タイムアウト (Timeout)]
?	パケットタイプが不明

パフォーマンス

ping および traceroute コマンドを使用して、ラウンドトリップ時間 (RTT) を取得できます。これは、エコー パケットを送り、返答を受け取るのに必要な時間です。これにより、リンクでの遅延をおおまかに理解することができます。しかし、この数字はパフォーマンス評価に使えるほど正確ではありません。

パケットの宛先がルータ自体である場合、このパケットはプロセススイッチされる必要があります。プロセッサは、このパケットからの情報を処理し、返答を戻さなければなりません。これは、ルータの第一の目的ではありません。定義上、ルータはパケットをルート付けるように構築さ

れています。ping に対する応答は、ベストエフォート型サービスとして提供されます。

このことを説明するため、Router1 から Router2 への ping の実行例を次に示します。

```
<#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
```

RTT は約 4 ミリ秒です。Router2 でプロセスを多用する機能を有効にした後、Router1 から Router2 へ ping を実行してください。

```
<#root>
```

```
Router1#
```

```
ping 172.16.0.12
```

```
Type
```

```
escape sequence
```

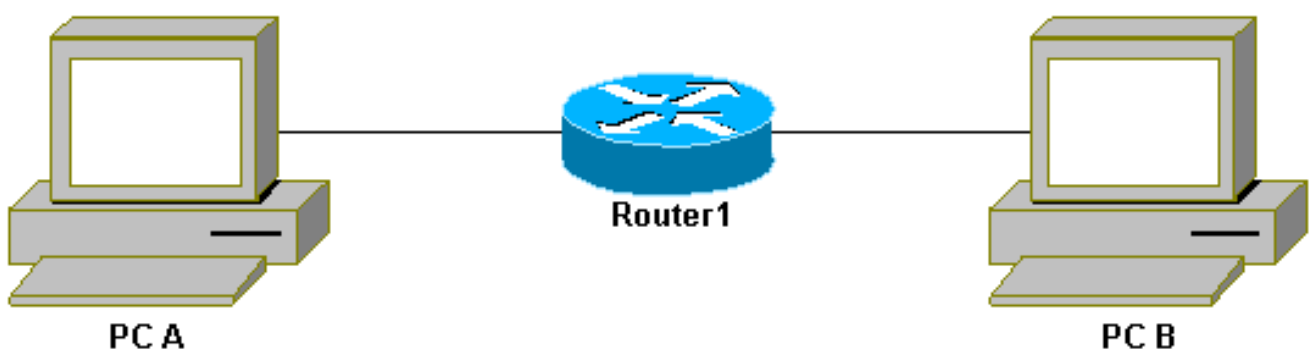
```
to abort.
```

```
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
```

これで、RTT が大幅に増加します。Router2 は非常にビジー状態ですので、ping に対する応答は優先順位が高くありません。ルータのパフォーマンスをテストするもっとよい方法として、ルータを通過するトラフィックを使うものがあります。



ルータを通過するトラフィック

トラフィックはルータにより最優先として、ファストスイッチされます。基本的なネットワークでこれを説明します。



ルータ3つの基本的なネットワーク

Router1 から Router3 へ ping を実行します。

```
<#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
```

トラフィックは Router2 を通過し、ファストスイッチされるようになります。Router2 でプロセスを多用する機能を有効にします。

```
<#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
```

ほとんど違いはありません。これは、Router2 で、パケットが割り込みレベルで扱われるためです。

debug コマンドの使用

debug コマンドを使用する前に、「debug コマンドの重要な情報」を参照してください。

この記事で使用されているさまざまな debug コマンドは、ping または traceroute コマンドが使用されたときに何が起こるかを示します。これらのコマンドは、問題のトラブルシューティングに役立ちます。しかし、実稼動環境では、debug は注意して使用する必要があります。使っている CPU が強力でない場合、またはプロセス交換されるパケットが大量にある場合、使用中のデバイスの処理速度が簡単に低下してしまう可能性があります。ルータに対する debug コマンドの影響を最小に抑える方法がいくつかあります。アクセス リストを使用して、監視が必要な特定のトラフィックに限定するのもその 1 つです。

ランダム データの例は次のとおりです。

```
<#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)
```

この設定を使うと、Router4 は、access-list 150 と一致する debug メッセージしか出力しません。Router1 から ping を実行すると、次のメッセージが表示されます。

```
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
```

これらのパケットは access-list と一致しないため、Router4 からは問題への応答は得られません。それらを表示するには、次を追加します。

<#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
```

[Results] :

```
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
```

debug コマンドの影響を抑えるもう一つの方法は、デバッグメッセージをバッファに入れ、デバッグをオフにした後に show log コマンドを使って表示するというものです。

```
<#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
```

ping および traceroute コマンドは、ネットワークアクセス障害をトラブルシューティングするために使用できる、便利なユーティリティです。使い方は非常に簡単です。これら 2 つのコマンドは、ネットワークエンジニアによって広く使用されています。

関連情報

- [拡張 ping および拡張 traceroute コマンドについて](#)
- [テクニカルサポート - Cisco Systems](#)

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。