ping および traceroute コマンドについて

目次

はじめに

このドキュメントでは、**ping および traceroute コマンド**の使い方を説明します。 ここでは、いくつかの **debug コマンド**を使用して、これらのコマンドがどのように機能するかを詳細な図で示しています。

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」のセクションを入念に読むことをお勧めします。

前提条件

要件

このドキュメントに関しては個別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。

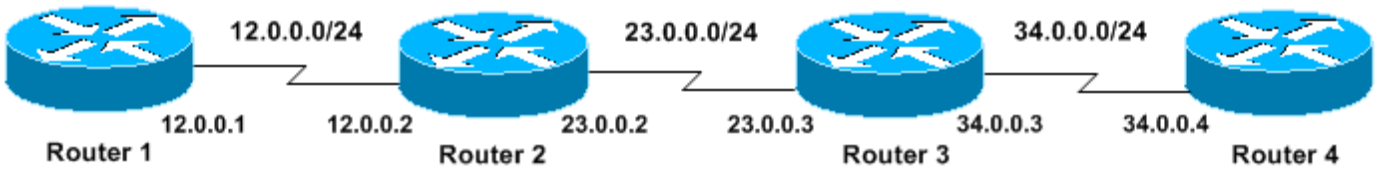
本書の情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。 このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。 稼働中のネットワークで作業を行う場合、コマンドの影響について十分に理解したうえで作業してください。

表記法

ドキュメント表記の詳細は、『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

このドキュメントでは、次に示す基本設定を、今回の例の基本として使用します。



ping コマンド

ping（Packet InterNet Groper）コマンドは、デバイスのアクセス可能性のトラブルシューティングに非常によく使用される方法です。 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
- ホストと通信する場合の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。

このコマンドのオプションについては、「トラブルシューティング コマンド」の「ping」を参照してください。

ping パケットの TTL 値は変更できません。

debug ip packet detail コマンドを有効にした後の **ping コマンド**の出力例を次に示します。

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。 **debug コマンド**を実行する前に、「debug コマンドの使用」のセクション

を入念に読むことをお勧めします。

```
Router1#debug ip packet detail
IP packet debugging is on (detailed)

Router1#ping 12.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms

Router1#
Jan 20 15:54:47.487: IP: s=12.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
  sending
Jan 20 15:54:47.491: ICMP type=8, code=0
!--- This is the ICMP packet 12.0.0.1 sent to 12.0.0.2. !--- ICMP type=8 corresponds to the echo
message. Jan 20 15:54:47.523: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 100, rcvd 3
Jan 20 15:54:47.527: ICMP type=0, code=0
!--- This is the answer we get from 12.0.0.2. !--- ICMP type=0 corresponds to the echo reply
message. !--- By default, the repeat count is five times, so there will be five !--- echo
requests, and five echo replies.
```

次の表に、有効な ICMP タイプの値を示します。

ICMP タイプ	内容
0	echo-reply
3	宛先到達不能コード 0 = ネット到達不能 1 = ホスト到達不能 2 = プロトコル到達不能 3 = ポート到達不能 4 = 必要なフラグメンテーションと DF セット 5 = 送信元ルート失敗
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	echo
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

次の表は、ping の機能から出力される可能性のある文字を一覧にしたものです。

文字	説明
!!	各感嘆符 (!) は、応答の受信を示します。
を探します。	ピリオド (.) は、ネットワーク サーバ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

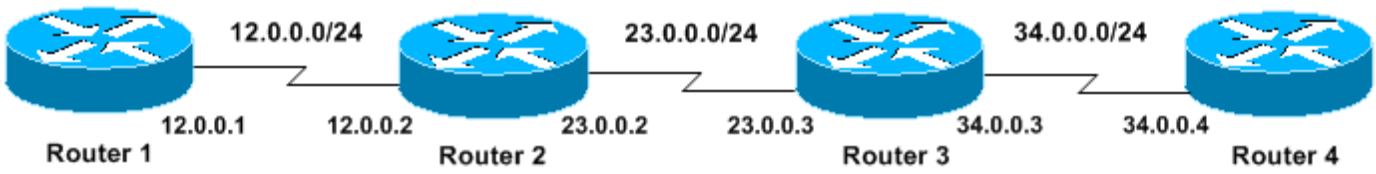
ping が失敗する理由

特定のアドレスに対して ping を正常に実行できない場合、次の原因を検討してください。

ルーティング問題

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。

次のネットワーク トポロジ ダイアグラムを使って、このシナリオを説明します。




```
Router1#
!
!
interface Serial0
ip address 12.0.0.1 255.255.255.0
no fair-queue
clockrate 64000
!
!

Router2#
!
!
```

```
interface Serial0
ip address 23.0.0.2 255.255.255.0
no fair-queue
clockrate 64000
!
interface Serial1
ip address 12.0.0.2 255.255.255.0
!
!
```

```
Router3#
!
!
interface Serial0
ip address 34.0.0.3 255.255.255.0
no fair-queue
!
interface Serial1
ip address 23.0.0.3 255.255.255.0
!
!
```

```
Router4#
!
!
interface Serial0
ip address 34.0.0.4 255.255.255.0
no fair-queue
clockrate 64000
!
!
```

Router1 から Router4 へ ping を実行します。

```
Router1#ping 34.0.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

何が起こったか、詳しく調べてみましょう。

```
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。 **debug コマンド**を実行する前に、「debug コマンドの使用」のセクションを入念に読むことをお勧めします。

```
Router1#ping 34.0.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:

Jan 20 16:00:25.603: IP: s=12.0.0.1 (local), d=34.0.0.4, len 100, unroutable.
Jan 20 16:00:27.599: IP: s=12.0.0.1 (local), d=34.0.0.4, len 100, unroutable.
Jan 20 16:00:29.599: IP: s=12.0.0.1 (local), d=34.0.0.4, len 100, unroutable.
Jan 20 16:00:31.599: IP: s=12.0.0.1 (local), d=34.0.0.4, len 100, unroutable.
```

```
Jan 20 16:00:33.599: IP: s=12.0.0.1 (local), d=34.0.0.4, len 100, unroutable.
Success rate is 0 percent (0/5)
```

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 ルートを追加してみましょう。

```
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
```

次のようになります。

```
Router1#debug ip packet detail
IP packet debugging is on (detailed)

Router1#ping 34.0.0.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

Jan 20 16:05:30.659: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
  sending
Jan 20 16:05:30.663:     ICMP type=8, code=0
Jan 20 16:05:30.691: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 56,
  rcvd 3
Jan 20 16:05:30.695:     ICMP type=3, code=1
Jan 20 16:05:30.699: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
  sending
Jan 20 16:05:30.703:     ICMP type=8, code=0
Jan 20 16:05:32.699: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
  sending
Jan 20 16:05:32.703:     ICMP type=8, code=0
Jan 20 16:05:32.731: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 56,
  rcvd 3
Jan 20 16:05:32.735:     ICMP type=3, code=1
Jan 20 16:05:32.739: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
  sending
Jan 20 16:05:32.743:     ICMP type=8, code=0
```

Router2 の不具合点を調べてみましょう。

```
Router2#debug ip packet detail
IP packet debugging is on (detailed)

Router2#
Jan 20 16:10:41.907: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:41.911:     ICMP type=8, code=0
Jan 20 16:10:41.915: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:41.919:     ICMP type=3, code=1
Jan 20 16:10:41.947: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:41.951:     ICMP type=8, code=0
Jan 20 16:10:43.943: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:43.947:     ICMP type=8, code=0
Jan 20 16:10:43.951: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:43.955:     ICMP type=3, code=1
Jan 20 16:10:43.983: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:43.987:     ICMP type=8, code=0
```

```
Jan 20 16:10:45.979: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:45.983:     ICMP type=8, code=0
Jan 20 16:10:45.987: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:45.991:     ICMP type=3, code=1
```

Router1 は Router2 に正しくパケットを送信していますが、Router2 はアドレス 34.0.0.4 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。

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

```
Router2#debug ip packet detail
IP packet debugging is on (detailed)

Router2#
Jan 20 16:10:41.907: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:41.911:     ICMP type=8, code=0
Jan 20 16:10:41.915: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:41.919:     ICMP type=3, code=1
Jan 20 16:10:41.947: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:41.951:     ICMP type=8, code=0
Jan 20 16:10:43.943: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:43.947:     ICMP type=8, code=0
Jan 20 16:10:43.951: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:43.955:     ICMP type=3, code=1
Jan 20 16:10:43.983: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:43.987:     ICMP type=8, code=0
Jan 20 16:10:45.979: IP: s=12.0.0.1 (Serial1), d=34.0.0.4, len 100, unroutable
Jan 20 16:10:45.983:     ICMP type=8, code=0
Jan 20 16:10:45.987: IP: s=12.0.0.2 (local), d=12.0.0.1 (Serial1), len 56, sending
Jan 20 16:10:45.991:     ICMP type=3, code=1
```

次のように出力されます。

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

Router1#ping 34.0.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:

Jan 20 16:16:13.367: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
 sending.
Jan 20 16:16:15.363: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
 sending.
Jan 20 16:16:17.363: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
 sending.
Jan 20 16:16:19.363: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
 sending.
Jan 20 16:16:21.363: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
 sending.
Success rate is 0 percent (0/5)
```

これでやや改善されました。 Router1 は Router4 にパケットを送っていますが、Router4 から応答を得ていません。

Router4 の問題点を調べて見ましょう。

```
Router4#debug ip packet
```

```
IP packet debugging is on

Router4#
Jan 20 16:18:45.903: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:45.911: IP: s=34.0.0.4 (local), d=12.0.0.1, len 100, unroutable
Jan 20 16:18:47.903: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:47.907: IP: s=34.0.0.4 (local), d=12.0.0.1, len 100, unroutable
Jan 20 16:18:49.903: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:49.907: IP: s=34.0.0.4 (local), d=12.0.0.1, len 100, unroutable
Jan 20 16:18:51.903: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:51.907: IP: s=34.0.0.4 (local), d=12.0.0.1, len 100, unroutable
Jan 20 16:18:53.903: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:53.907: IP: s=34.0.0.4 (local), d=12.0.0.1, len 100, unroutable
```

Router4 は ICMP パケットを受信し、12.0.0.1 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、成功しません。

Router4 にスタティック ルートを追加してみましょう。

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

これで正しく機能し、両サイドでお互いにアクセスできます。

```
Router1#ping 34.0.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/35/36 ms
```

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 ping を実行します。

```
Router1#ping 34.0.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

ルーティングには問題がないため、手順を追ってトラブルシューティングを行います。 最初に Router2 に ping を実行してみます。

```
Router1#ping 12.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

上記から、Router2 と Router3 の間に問題があることが分かります。 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。

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

これは、修復するのが非常に簡単です。

```
Router3#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router3(config)#interface s1
Router3(config-if)#no shutdown
Router3(config-if)#
Jan 20 16:20:53.900: %LINK-3-UPDOWN: Interface Serial1, changed state to up
Jan 20 16:20:53.910: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1,
  changed state to up
```

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 に着信できるようにしたいと考えています。

```
Router4(config)# access-list 100 permit tcp any any eq telnet
Router4(config)#interface s0
Router4(config-if)#ip access-group 100 in
```

```
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#access-list 100 permit ip host 12.0.0.1 host 34.0.0.4
Router1(config)#access-list 100 permit ip host 34.0.0.4 host 12.0.0.1
Router1(config)#end
Router1#debug ip packet 100
IP packet debugging is on
Router1#debug ip icmp
ICMP packet debugging is on
```

debug コマンドを使ったアクセス リストの使用については、「debug コマンドの使用」のセクションを参照してください。

Router4 に ping してみると、次のように出力されます。

```
Router1#ping 34.0.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 34.0.0.4, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

Jan 20 16:34:49.207: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
 sending
Jan 20 16:34:49.287: IP: s=34.0.0.4 (Serial0), d=12.0.0.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:34:49.291: ICMP: dst (12.0.0.1) administratively prohibited unreachable
 rcv from 34.0.0.4
Jan 20 16:34:49.295: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
 sending
Jan 20 16:34:51.295: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
```

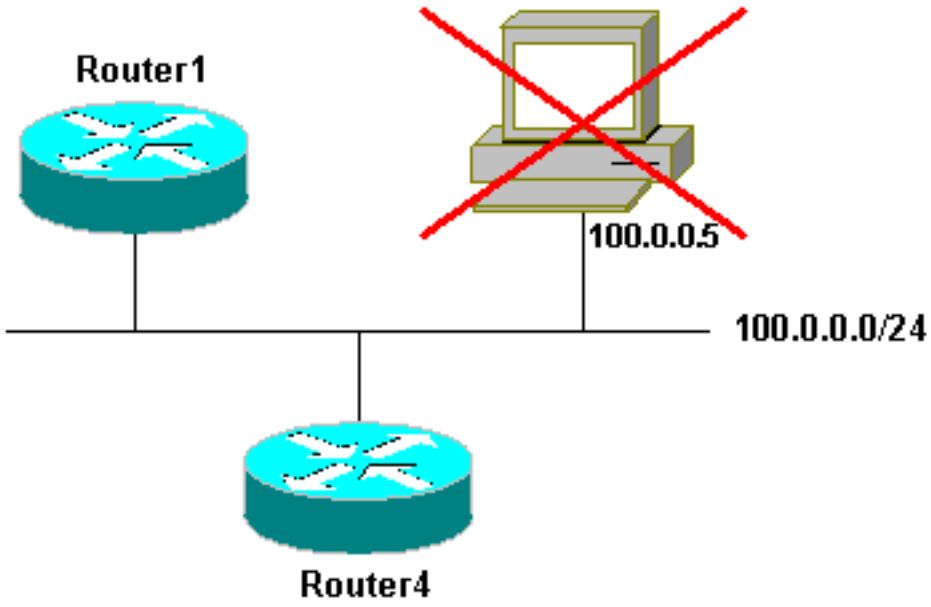
```
 sending
Jan 20 16:34:51.367: IP: s=34.0.0.4 (Serial0), d=12.0.0.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:34:51.371: ICMP: dst (12.0.0.1) administratively prohibited unreachable
 rcv from 34.0.0.4
Jan 20 16:34:51.379: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 100,
 sending
```

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。 これはつまり、Router4 の Serial 0 インターフェイスに到着した ICMP パケットが拒否されたことになります。 さらに 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ります。 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。

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

Address Resolution Protocol (ARP) 問題

ここでは、イーサネット接続を使ったシナリオについて説明します。




```
Router4#ping 100.0.0.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.0.0.5, timeout is 2 seconds:

Jan 20 17:04:05.167: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:05.171: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:07.167: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:07.171: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:09.175: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  sending
```

```
Jan 20 17:04:09.183: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:11.175: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:11.179: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  encapsulation failed.
Jan 20 17:04:13.175: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  sending
Jan 20 17:04:13.179: IP: s=100.0.0.4 (local), d=100.0.0.5 (Ethernet0), len 100,
  encapsulation failed.
Success rate is 0 percent (0/5)
Router4#
```

この例の中では、「encapsulation failed」のために ping は動作していません。 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) の動作方法を理解している必要があります。 詳細は、『アドレス解決方法の設定』を参照してください。

基本的に、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。 このマッピングは、**show arp コマンド**を使ってチェックできます。

```
Router4#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  100.0.0.4               -   0000.0c5d.7a0d  ARPA   Ethernet0
Internet  100.0.0.1              10   0060.5cf4.a955  ARPA   Ethernet0
```

「カプセル化失敗」の問題に戻ります。 この **debug コマンド**を使えば、この問題が把握しやすくなります。

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

Router4#ping 100.0.0.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.0.0.5, timeout is 2 seconds:

Jan 20 17:19:43.843: IP ARP: creating incomplete entry for IP address: 100.0.0.5
  interface Ethernet0
Jan 20 17:19:43.847: IP ARP: sent req src 100.0.0.4 0000.0c5d.7a0d,
                 dst 100.0.0.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:45.843: IP ARP: sent req src 100.0.0.4 0000.0c5d.7a0d,
                 dst 100.0.0.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:47.843: IP ARP: sent req src 100.0.0.4 0000.0c5d.7a0d,
                 dst 100.0.0.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:49.843: IP ARP: sent req src 100.0.0.4 0000.0c5d.7a0d,
                 dst 100.0.0.5 0000.0000.0000 Ethernet0.
Jan 20 17:19:51.843: IP ARP: sent req src 100.0.0.4 0000.0c5d.7a0d,
                 dst 100.0.0.5 0000.0000.0000 Ethernet0.
Success rate is 0 percent (0/5)
```

上記出力は、Router4 がイーサネット ブロードキャスト アドレス FFFF.FFFF.FFFF にパケットを送信することによって、そのパケットをブロードキャストしていることを示しています。 この場合、0000.0000.0000 は、Router4 が送信先 100.0.0.5 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 要求実行中、MAC アドレスを認識していないため、インターフェイス Ethernet 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、100.0.0.5 に対応する MAC アドレスを問い合せています。 応答が得られない場合、show arp 出力内の対応するアドレスが不完全として**マーキ**

ングされます。

```
Router4#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  100.0.0.4               -   0000.0c5d.7a0d  ARPA   Ethernet0
Internet  100.0.0.5               0   Incomplete      ARPA
Internet  100.0.0.1               2   0060.5cf4.a955  ARPA   Ethernet0
```

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。 対応する MAC アドレスが ARP テーブル中にない場合、ping が失敗して「encapsulation failed」の結果となります。

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。

```
Router1#ping 12.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

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。

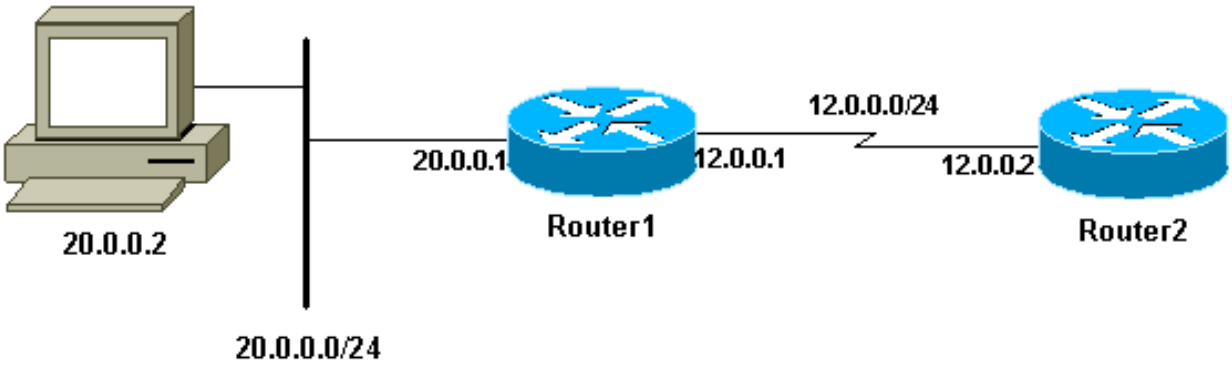
```
Router1#ping
Protocol [ip]:
Target IP address:  12.0.0.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]: 30
Extended commands [n]:
Sweep range of sizes [n]:

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 30 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1458/2390/6066 ms
```

上の例では、タイムアウト値を増やしたことによって ping が成功しています。

注: ラウンドトリップの平均時間は 2 秒を超えています。

正しい送信元アドレス

以下に、典型的な状況の例を示します。



Router1 に LAN インターフェイスを追加します。

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

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着信するパケットに応答する必要があっても、処理方法がわかりません。

```
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。 **debug コマンド**を実行する前に、「debug コマンドの使用」のセクションを入念に読むことをお勧めします。

```
Router1#ping 12.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/7/9 ms
Router1#

Jan 20 16:35:54.227: IP: s=12.0.0.1 (local), d=12.0.0.2 (Serial0), len 100, sending
Jan 20 16:35:54.259: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 100, rcvd 3
```

上記の出力例は、送信しているパケットの送信元アドレスが s=12.0.0.1 であるために機能します。 LAN から来るパケットをシミュレートする場合には、拡張 ping を使用する必要があります。

```
Router1#ping
Protocol [ip]:
Target IP address: 12.0.0.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
```

```
Extended commands [n]: y
Source address or interface: 20.0.0.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:

Jan 20 16:40:18.303: IP: s=20.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
 sending.
Jan 20 16:40:20.303: IP: s=20.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
 sending.
Jan 20 16:40:22.303: IP: s=20.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
 sending.
Jan 20 16:40:24.303: IP: s=20.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
 sending
Jan 20 16:40:26.303: IP: s=20.0.0.1 (local), d=12.0.0.2 (Serial0), len 100,
 sending.
Success rate is 0 percent (0/5)
```

この場合、ソース アドレスは 20.0.0.1 であり、これは動作していません。 パケットの送信はしていますが、受信するパケットがありません。 この問題を修正するには、単に Router2 に 20.0.0.0 へのルートを追加します。

基本的なルールは、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るということです。

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力インターフェイスの入力キューに入れられます。 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アップしていても、高入力キュー ドロップが原因でデバイスを ping できない可能性があります。 **show interface コマンド**で入力ドロップを確認できます。

```
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
```

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、パケットが宛先に移動するときに実際に経由するルートを検出するのに使います。 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 値が 1 の場合には、データグラムがパス上の最初のルータに到達した時点で、すぐにタイムアウトになります。 その後、このルータは、データグラムが期限切れになったことを示す ICMP Time Exceeded Message（TEM）で応答します。

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 Port Unreachable メッセージが返され、ポートが到達不能であることを示します。 このイベントは、終了の信号を traceroute プログラムに送ります。

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。 このコマンドのすべてのオプションについては、『Trace（特権コマンド）』を参照してください。

```
Router1#traceroute 34.0.0.4

Type escape sequence to abort.
Tracing the route to 34.0.0.4

  1 12.0.0.2 4 msec 4 msec 4 msec
  2 23.0.0.3 20 msec 16 msec 16 msec
  3 34.0.0.4 16 msec *  16 msec

Jan 20 16:42:48.611: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.615:     UDP src=39911, dst=33434
Jan 20 16:42:48.635: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:48.639:     ICMP type=11, code=0
!--- ICMP Time Exceeded Message from Router2. Jan 20 16:42:48.643: IP: s=12.0.0.1 (local),
d=34.0.0.4 (Serial0), len 28, sending Jan 20 16:42:48.647: UDP src=34237, dst=33435 Jan 20
16:42:48.667: IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 56, rcvd 3 Jan 20
16:42:48.671: ICMP type=11, code=0 Jan 20 16:42:48.675: IP: s=12.0.0.1 (local), d=34.0.0.4
(Serial0), len 28, sending Jan 20 16:42:48.679: UDP src=33420, dst=33436 Jan 20 16:42:48.699:
IP: s=12.0.0.2 (Serial0), d=12.0.0.1 (Serial0), len 56, rcvd 3 Jan 20 16:42:48.703: ICMP
type=11, code=0
```

これが、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（12.0.0.2））はパケットを廃棄し、送信元（12.0.0.1）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。

```
Jan 20 16:42:48.707: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.711:     UDP src=35734, dst=33437
Jan 20 16:42:48.743: IP: s=23.0.0.3 (Serial0), d=12.0.0.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:48.747:     ICMP type=11, code=0
!--- ICMP Time Exceeded Message from Router3. Jan 20 16:42:48.751: IP: s=12.0.0.1 (local),
d=34.0.0.4 (Serial0), len 28, sending Jan 20 16:42:48.755: UDP src=36753, dst=33438 Jan 20
16:42:48.787: IP: s=23.0.0.3 (Serial0), d=12.0.0.1 (Serial0), len 56, rcvd 3 Jan 20
16:42:48.791: ICMP type=11, code=0 Jan 20 16:42:48.795: IP: s=12.0.0.1 (local), d=34.0.0.4
(Serial0), len 28, sending Jan 20 16:42:48.799: UDP src=36561, dst=33439 Jan 20 16:42:48.827:
IP: s=23.0.0.3 (Serial0), d=12.0.0.1 (Serial0), len 56, rcvd 3 Jan 20 16:42:48.831: ICMP
type=11, code=0
```

TTL=2 の場合には Router3（23.0.0.3）で同じプロセスが発生します。

```
Jan 20 16:42:48.839: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.843:     UDP src=34327, dst=33440
Jan 20 16:42:48.887: IP: s=34.0.0.4 (Serial0), d=12.0.0.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:48.891:     ICMP type=3, code=3
!--- Port Unreachable message from Router4. Jan 20 16:42:48.895: IP: s=12.0.0.1 (local),
d=34.0.0.4 (Serial0), len 28, sending Jan 20 16:42:48.899: UDP src=37534, dst=33441 Jan 20
16:42:51.895: IP: s=12.0.0.1 (local), d=34.0.0.4 (Serial0), len 28, sending Jan 20 16:42:51.899:
UDP src=37181, dst=33442 Jan 20 16:42:51.943: IP: s=34.0.0.4 (Serial0), d=12.0.0.1 (Serial0),
len 56, rcvd 3 Jan 20 16:42:51.947: ICMP type=3, code=3
```

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**出力に現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
?	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）を取得します。 これは、エコー パケットの送信および、アンサーバックの取得に必要な時間です。 またこれは、リンクに対する遅延をおおまかに理解するのに便利です。 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て、アンサーバックを送信する必要があります。 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 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。

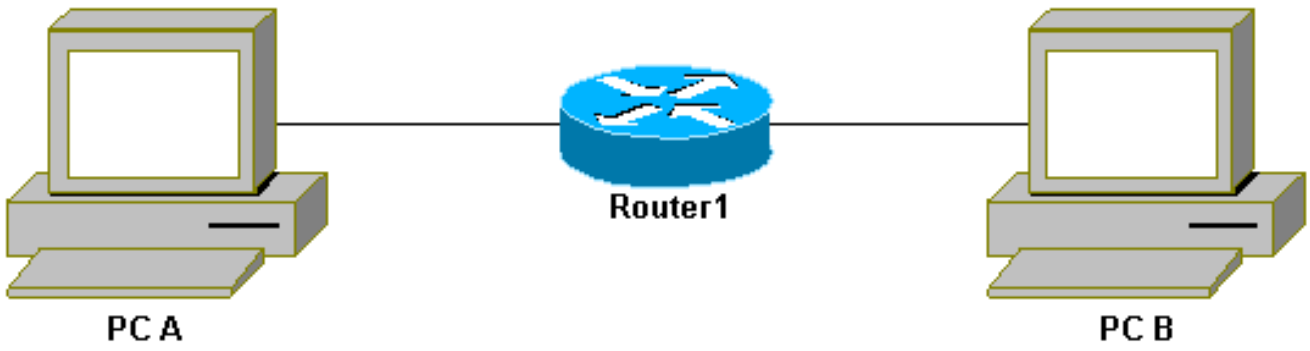
```
Router1#ping 12.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

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。

```
Router1#ping 12.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/25/28 ms
```

これで、RTT が大幅に増加します。 Router2 は非常にビジー状態であり、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。



次にトラフィックはファースト スイッチングされ、ルータにより最高の優先度で処理されます。 このことを説明するため、基本のネットワークに戻ってみましょう。

Router1 から Router3 へ ping します。

```
Router1#ping 23.0.0.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 23.0.0.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
```

トラフィックは Router2 を通過する際、ファストスイッチされます。

Router2 でプロセスを多用する機能を有効にしてみましょう。

```
Router1#ping 23.0.0.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 23.0.0.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
```

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 コマンドに関する重要な情報』を参照してください。

これまで使用してきたさまざまな **debug コマンド**によって、**ping または traceroute コマンド**を使用した場合に、どのようなことが起こるかがわかります。 これらのコマンドは、トラブルシューティングにも役立ちます。 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。 次に例を示します。

```
Router4#debug ip packet ?
  <1-199>      Access list
  <1300-2699>  Access list (expanded range)
  detail       Print more debugging detail

Router4#configure terminal
Router4(config)#access-list 150 permit ip host 12.0.0.1 host 34.0.0.4
Router4(config)#^Z

Router4#debug ip packet 150
IP packet debugging is on for access list 150
```

```
Router4#show debug
Generic IP:
  IP packet debugging is on for access list 150

Router4#show access-list
Extended IP access list 150
    permit ip host 12.0.0.1 host 34.0.0.4 (5 matches)
```

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 から到着する ping によって、次のメッセージが表示されます:

```
Router4#debug ip packet ?
  <1-199>      Access list
  <1300-2699>  Access list (expanded range)
  detail       Print more debugging detail

Router4#configure terminal
Router4(config)#access-list 150 permit ip host 12.0.0.1 host 34.0.0.4
Router4(config)#^Z

Router4#debug ip packet 150
IP packet debugging is on for access list 150

Router4#show debug
Generic IP:
  IP packet debugging is on for access list 150

Router4#show access-list
Extended IP access list 150
    permit ip host 12.0.0.1 host 34.0.0.4 (5 matches)
```

これらのパケットは access-list と一致しないため、Router4 からの応答は表示されなくなります。 応答を表示するには、次のコマンドを追加する必要があります。

```
Router4(config)#access-list 150 permit ip host 12.0.0.1 host 34.0.0.4
Router4(config)#access-list 150 permit ip host 34.0.0.4 host 12.0.0.1
```

次のような結果が得られます。

```
Router4(config)#access-list 150 permit ip host 12.0.0.1 host 34.0.0.4
Router4(config)#access-list 150 permit ip host 34.0.0.4 host 12.0.0.1
```

debug コマンドの影響を最小限に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。

```
Router4#configure terminal
Router4(config)#no logging console
Router4(config)#logging buffered 5000
Router4(config)#^Z

Router4#debug ip packet
IP packet debugging is on
Router4#ping 12.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.0.0.1, timeout is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/36/37 ms

Router4#undebug all
All possible debugging has been turned off

Router4#show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged
    Buffer logging: level debugging, 61 messages logged
    Trap logging: level informational, 59 message lines logged

Log Buffer (5000 bytes):

Jan 20 16:55:46.587: IP: s=34.0.0.4 (local), d=12.0.0.1 (Serial0), len 100,
 sending
Jan 20 16:55:46.679: IP: s=12.0.0.1 (Serial0), d=34.0.0.4 (Serial0), len 100,
 rcvd 3
```

このように、**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、非常に有用なユーティリティです。 使い方は非常に簡単です。 この 2 つのコマンドは、ネットワーク技術者によって最もよく使われているコマンドで、ネットワーク接続をトラブルシューティングするには、これらのコマンドを理解することが非常に重要です。

関連情報

- 拡張 ping および拡張 traceroute コマンドの使用
- テクニカルサポート - Cisco Systems