Comprendre les commandes ping et traceroute

Table des matières

Introduction

Conditions préalables

- Exigences
- Composants utilisés
- Conventions

Informations générales

Commande Ping

Dépannage des échecs de ping

- Problème de routeur
- Interface désactivée
- Commande de liste d'accès
- Problème ARP (Address Resolution Protocol)
- Délai
- Adresse source correcte
- Pertes de file d'attente entrantes élevées

Commande Traceroute

rendement

Utiliser la commande Debug

Informations connexes

Introduction

Ce document décrit l’utilisation des commandes Ping et Traceroute sur les routeurs Cisco.

Conditions préalables

Exigences

Aucune exigence spécifique n'est associée à ce document.

Composants utilisés

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

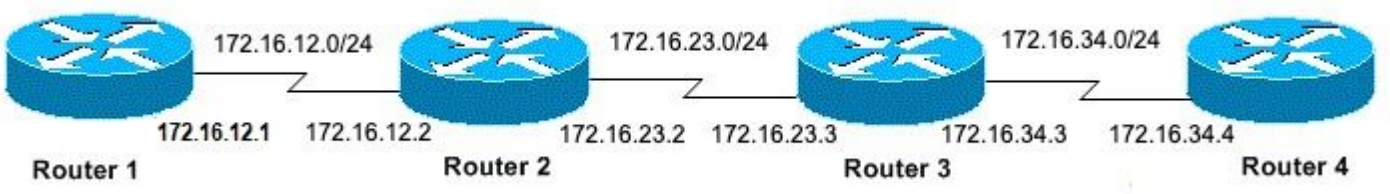
Conventions

Pour plus d'informations sur les conventions de document, référez-vous à Conventions relatives aux conseils techniques Cisco .

Informations générales

> Remarque : Toute commande debug utilisée sur un routeur de production peut causer de sérieux problèmes. Lisez la section Use the Debug Command (utiliser la commande de débogage) avant d’exécuter des commandes de débogage.

Les exemples de ce document utilisent cette topologie de réseau de base :



Topologie du réseau

Commande Ping

La commande ping est très courante pour le dépannage en cas de problèmes d’accessibilité des périphériques. Il utilise une série de messages d'écho ICMP (Internet Control Message Protocol) pour déterminer :

- Indique si un hôte distant est actif ou inactif.
- La durée aller-retour pour la communication avec l’hôte.
- Perte de paquets.

La commande ping envoie d'abord un paquet de requête d'écho à une adresse, puis attend une réponse. La requête ping aboutit uniquement si :

- la requête d'écho atteint la destination, et
- la destination est capable d'obtenir une réponse d'écho à la source dans un délai prédéterminé appelé délai d'attente. La valeur par défaut de ce délai d’attente est de deux secondes sur les routeurs Cisco.

La valeur TTL d'un paquet ping ne peut pas être modifiée.

Le prochain exemple de code montre la commande ping après l’activation de la commande debug ip packet detail.

Avertissement : Lorsque la commande debug ip packet detail est utilisée sur un routeur de production, elle peut entraîner une utilisation élevée du CPU. Une grave dégradation de la performance ou une panne du réseau pourrait s’ensuivre.

<#root>

```
Router1#

debug ip packet detail

IP packet debugging is on (detailed)

Router1#

ping 172.16.12.2


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, 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=172.16.12.1 (local), d=172.16.12.2 (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 172.16.12.2.
!--- ICMP type=8 corresponds to the echo message.

Jan 20 15:54:47.523: IP: s=172.16.12.2 (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 172.16.12.2. !--- 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.
```

Valeurs de type ICMP possibles

Type ICMP	Littéral
0	réponse d'écho
3	code de destination impossible à joindre 0 = réseau inaccessible 1 = hôte inaccessible 2 = protocole inaccessible 3 = port inaccessible 4 = fragmentation requise et bit DF 5 = échec du routage source
4	source-quench
5	code de redirection 0 = rediriger les datagrammes pour le réseau 1 = rediriger les datagrammes pour l'hôte 2 = rediriger les datagrammes pour le type de service et le réseau 3 = rediriger les datagrammes pour le type de service et l'hôte
6	adresse de remplacement
8	echo
9	annonce de routeur
10	routeur-sollicitation
11	code de temps dépassé 0 = durée de vie utile dépassée pendant le transit 1 = temps de réassemblage des fragments dépassé
12	problème-paramètre
13	timestamp-request
14	réponse d'horodatage
15	demande d'information
16	information-réponse
17	requête-masque
18	masque-réponse
31	erreur de conversion
32	redirection par téléphone mobile

Caractères de sortie possibles à partir de la fonction Ping

Caractère	Description
!	Chaque point d'exclamation indique la réception d'une réponse.
.	Un point (.) indique que le serveur du réseau a expiré lorsqu'il attendait une

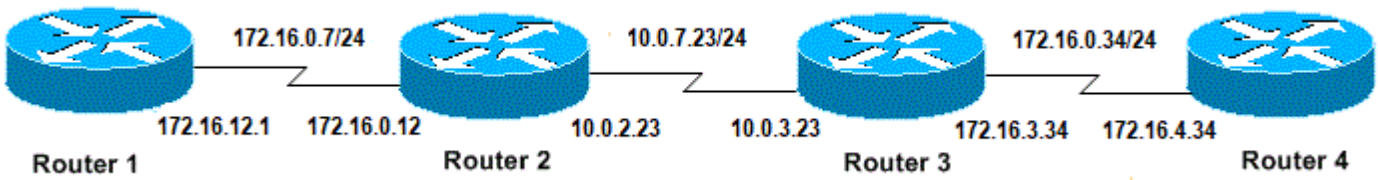
	réponse.
U	Une PDU d'erreur de destination inaccessible a été reçue.
Q	Épuisement de la source (destination trop occupée).
L	Impossible de fragmenter.
?	Type de paquet inconnu.
et	Durée de vie des paquets dépassée.

Dépannage des échecs de ping

Si vous ne parvenez pas à envoyer un message ping vers une adresse IP, examinez la liste de causes figurant dans cette section.

Problème de routeur

Voici des exemples de tentatives d'envoi de message ping infructueux qui peuvent déterminer le problème ainsi que la marche à suivre pour le résoudre. Cet exemple est accompagné du diagramme de topologie de réseau suivant :



Problèmes de routeur

<#root>

Router1#

```
!
interface Serial0
ip address 172.16.12.1 255.255.255.0
no fair-queue
clockrate 64000
!
```

Router2#

```
!
```

```
interface Serial0
ip address 172.16.23.2 255.255.255.0
no fair-queue
clockrate 64000
!
interface Serial1
ip address 172.16.12.2 255.255.255.0
!
```

Router3#

```
!
interface Serial0
ip address 172.16.34.3 255.255.255.0
no fair-queue
!
interface Serial1
ip address 172.16.23.3 255.255.255.0
!
```

Router4#

```
!
interface Serial0
ip address 172.16.34.4 255.255.255.0
no fair-queue
clockrate 64000
!
```

Essayez d’envoyer un message ping au routeur 4 à partir du routeur 1 :

<#root>

```
Router1#
```

ping 172.16.34.4

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
.....
```

Success rate is 0 percent (0/5)

Résultats :

<#root>

```
Router1#
```

debug ip packet

```
IP packet debugging is on
```

⚠ Avertissement : Lorsque la commande debug ip packet est utilisée sur un routeur de production, elle peut entraîner une utilisation élevée du CPU. Une grave dégradation de la performance ou une panne du réseau pourrait s'ensuivre.

<#root>

```
Router1#
```

ping 172.16.34.4

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:

Jan 20 16:00:25.603: IP: s=172.16.12.1 (local), d=172.16.34.4, len 100, unroutable.
Jan 20 16:00:27.599: IP: s=172.16.12.1 (local), d=172.16.34.4, len 100, unroutable.
Jan 20 16:00:29.599: IP: s=172.16.12.1 (local), d=172.16.34.4, len 100, unroutable.
Jan 20 16:00:31.599: IP: s=172.16.12.1 (local), d=172.16.34.4, len 100, unroutable.
Jan 20 16:00:33.599: IP: s=172.16.12.1 (local), d=172.16.34.4, len 100, unroutable.
Success rate is 0 percent (0/5)
```

Étant donné qu'aucun protocole de routage n'est exécuté sur le routeur 1, celui-ci ne sait pas où envoyer son paquet et génère un message « unroutable ».

Ajoutez une route statique au routeur 1 :

<#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

Résultats :

<#root>

```
Router1#

debug ip packet detail

IP packet debugging is on (detailed)

Router1#

ping 172.16.34.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, 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=172.16.34.4 (Serial0), len 100,
  sending
Jan 20 16:05:30.663:     ICMP type=8, code=0
Jan 20 16:05:30.691: IP: s=172.16.12.2 (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=172.16.34.4 (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=172.16.34.4 (Serial0), len 100,
  sending
Jan 20 16:05:32.703:     ICMP type=8, code=0
Jan 20 16:05:32.731: IP: s=172.16.12.2 (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=172.16.34.4 (Serial0), len 100,
  sending
Jan 20 16:05:32.743:     ICMP type=8, code=0
```

Examinez ce qui ne va pas sur le routeur 2 :

<#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=172.16.34.4, len 100, unroutable
Jan 20 16:10:41.911:     ICMP type=8, code=0
Jan 20 16:10:41.915: IP: s=172.16.12.2 (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=172.16.34.4, 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=172.16.34.4, len 100, unroutable
Jan 20 16:10:43.947:     ICMP type=8, code=0
Jan 20 16:10:43.951: IP: s=172.16.12.2 (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=172.16.34.4, 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=172.16.34.4, len 100, unroutable
Jan 20 16:10:45.983:     ICMP type=8, code=0
Jan 20 16:10:45.987: IP: s=172.16.12.2 (local), d=172.16.12.1 (Serial1), len 56, sending
Jan 20 16:10:45.991:     ICMP type=3, code=1
```

Le routeur 1 a envoyé correctement ses paquets au routeur 2, mais le routeur 2 ne sait pas comment accéder à l'adresse 172.16.34.4. Le routeur 2 renvoie un message « inaccessible ICMP » au routeur 1.

Configurez un protocole de routage dynamique sur les routeurs 2 et 3 (dans cet exemple, le protocole RIP (Routing Information Protocol) est utilisé) :

```
Router2#
router rip
 network 172.16.0.0
 no auto-summary
Router3#
router rip
 network 172.16.0.0
 no auto-summary
```

Résultats :

<#root>

```
Router1#

debug ip packet

IP packet debugging is on

Router1#

ping 172.16.34.4


Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:

Jan 20 16:16:13.367: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending.
Jan 20 16:16:15.363: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending.
Jan 20 16:16:17.363: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending.
Jan 20 16:16:19.363: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending.
Jan 20 16:16:21.363: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending.
Success rate is 0 percent (0/5)
```

Le routeur 1 envoie des paquets au routeur 4, mais le routeur 4 ne renvoie pas de réponse.

Problème possible sur le routeur 4 :

<#root>

```
Router4#
```

debug ip packet

```
IP packet debugging is on

Router4#
Jan 20 16:18:45.903: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:45.911: IP: s=172.16.34.4 (local), d=172.16.12.1, len 100,
```

unroutable

```
Jan 20 16:18:47.903: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:47.907: IP: s=172.16.34.4 (local), d=172.16.12.1, len 100, unroutable
Jan 20 16:18:49.903: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:49.907: IP: s=172.16.34.4 (local), d=172.16.12.1, len 100, unroutable
Jan 20 16:18:51.903: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:51.907: IP: s=172.16.34.4 (local), d=172.16.12.1, len 100, unroutable
Jan 20 16:18:53.903: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
  rcvd 3
Jan 20 16:18:53.907: IP: s=172.16.34.4 (local), d=172.16.12.1, len 100, unroutable
```

Le routeur 4 reçoit les paquets ICMP et tente de répondre à l'adresse 172.16.12.1, mais comme il ne dispose d'aucune route vers ce réseau, il échoue.

Ajouter une route statique au routeur 4 :

<#root>

```
Router4(config)#

ip route 0.0.0.0 0.0.0.0 Serial0
```

Les deux côtés peuvent désormais communiquer entre eux :

<#root>

```
Router1#

ping 172.16.34.4


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/35/36 ms
```

Interface désactivée

Il s'agit d'une situation où l'interface s'arrête et ne fonctionne plus. Dans l'exemple suivant, une tentative d'envoi d'un message ping vers le routeur 4 s'effectue à partir du routeur 1 :

<#root>

```
Router1#

ping 172.16.34.4


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

Comme le routage est correct, procédez au dépannage étape par étape du problème. Essayez d'envoyer un message ping au routeur 2 :

<#root>

```
Router1#
```

```
ping 172.16.12.2



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

D’après l’exemple précédent, le problème se situe entre Router2 et Router3. Il est possible que l’interface série sur Router3 ait été arrêtée :

<#root>

```
Router3#

show ip interface brief

Serial0   172.16.34.3    YES manual up                      up
Serial1   172.16.23.3    YES manual administratively down   down
```

Cette situation est simple à résoudre :

<#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
```

Commande de liste d'accès

Dans ce scénario, seul le trafic Telnet est autorisé à entrer dans le routeur 4 par l’interface Serial0.

<#root>

```
Router4(config)#

access-list 100 permit tcp any any eq telnet

Router4(config)#

interface serial0

Router4(config-if)#

ip access-group 100 in



Router1#

configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#

access-list 100 permit ip host 172.16.12.1 host 172.16.34.4

Router1(config)#

access-list 100 permit ip host 172.16.34.4 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
```

Essayez d’envoyer un message ping au routeur 4 :

<#root>

```
Router1#
```

ping 172.16.34.4

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, 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=172.16.34.4 (Serial0), len 100,
 sending
Jan 20 16:34:49.287: IP: s=172.16.34.4 (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 172.16.34.4
Jan 20 16:34:49.295: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending
Jan 20 16:34:51.295: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending
Jan 20 16:34:51.367: IP: s=172.16.34.4 (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 172.16.34.4
Jan 20 16:34:51.379: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 100,
 sending
```

À la fin d’une commande access-list, il y a toujours un deny all implicite. Cela signifie que les paquets ICMP qui entrent dans l’interface Serial 0 sur le routeur 4 sont refusés, et le routeur 4 envoie un message ICMP « administratively disabled unreachable » à la source du paquet d’origine, comme indiqué dans le message de débogage. La solution est d’ajouter cette ligne dans la commande access-list :

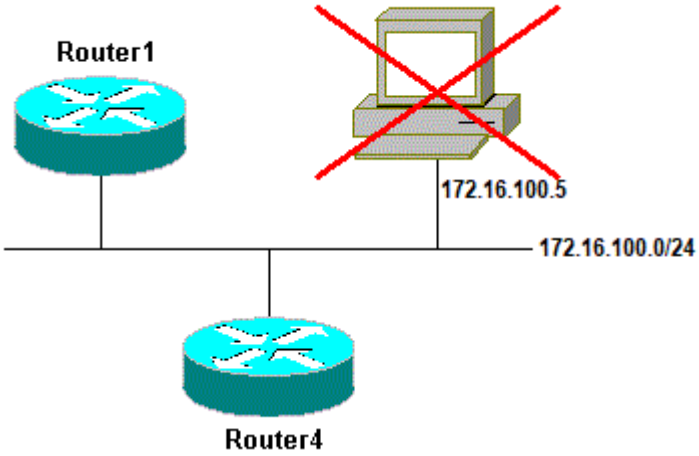
<#root>

```
Router4(config)#
```

access-list 100 permit icmp any any

Problème ARP (Address Resolution Protocol)

Dans ce scénario, la connexion Ethernet suivante est utilisée :



Problème de protocole ARP

<#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#
```

Dans cet exemple, la commande ping ne fonctionne pas à cause du message « encapsulation failed » (échec de l'encapsulation). Cela signifie que le routeur sait sur quelle interface il doit envoyer le paquet, mais ignore comment le faire. Dans ce cas, vous devez comprendre comment fonctionne le protocole ARP (Address Resolution Protocol).

ARP est un protocole utilisé pour mapper l'adresse de couche 2 (adresse MAC) avec une adresse de couche 3 (adresse IP). Vous pouvez vérifier à l'aide la commande show arp :

<#root>

```
Router4#

show arp


Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.16.100.4               -   0000.0c5d.7a0d  ARPA   Ethernet0
Internet  172.16.100.7              10   0060.5cf4.a955  ARPA   Ethernet0
```

Revenez au problème de l'« échec de l'encapsulation », mais activez cette fois la commande debug arp :

<#root>

```
Router4#

debug arp

ARP packet debugging is on

Router4#

ping 172.16.100.5



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

Jan 20 17:19:43.843: IP ARP: creating incomplete entry for IP address: 172.16.100.5
  interface Ethernet0
Jan 20 17:19:43.847: IP ARP: sent req src 172.16.100.4 0000.0c5d.7a0d,
```

```
dst 172.16.100.5 0000.0000.0000 Ethernet0.

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

La sortie précédente montre que le routeur 4 diffuse des paquets et les envoie à l'adresse de diffusion Ethernet FFFF.FFFF.FFFF. Ici, 0000.0000.0000 signifie que le routeur 4 recherche l'adresse MAC de destination 172.16.100.5. Comme il ne connaît pas l'adresse MAC alors que le protocole ARP est demandé dans cet exemple, il utilise 0000.000.000 comme espace réservé dans les trames de diffusion envoyées à partir de l'interface Ethernet 0 et demande quelle adresse MAC correspond à 172.16.100.5. S'il n'y a pas de réponse, l'adresse MAC correspond à l'adresse IP dans le résultat de la commande show arp est marqué comme incomplet :

<#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
```

Après une période prédéterminée, cette entrée incomplète est purgée de la table ARP. Tant que l'adresse MAC ne se trouve pas dans le tableau ARP, le message ping échoue en raison de l'échec de l'encapsulation.

Délai

Par défaut, si vous ne recevez pas de réponse de l'extrémité distante dans les deux secondes, la requête ping échoue :

<#root>

```
Router1#
```

```
ping 172.16.12.2



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2,

timeout is 2 seconds:


.....
Success rate is 0 percent (0/5)
```

Sur les réseaux à liaison lente ou à long délai, deux secondes ne suffisent pas. Vous pouvez modifier ce paramètre par défaut grâce à la commande extended ping :

<#root>

```
Router1#

ping


Protocol [ip]:
Target IP address:  172.16.12.2
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 172.16.12.2, timeout is 30 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1458/2390/6066 ms
```

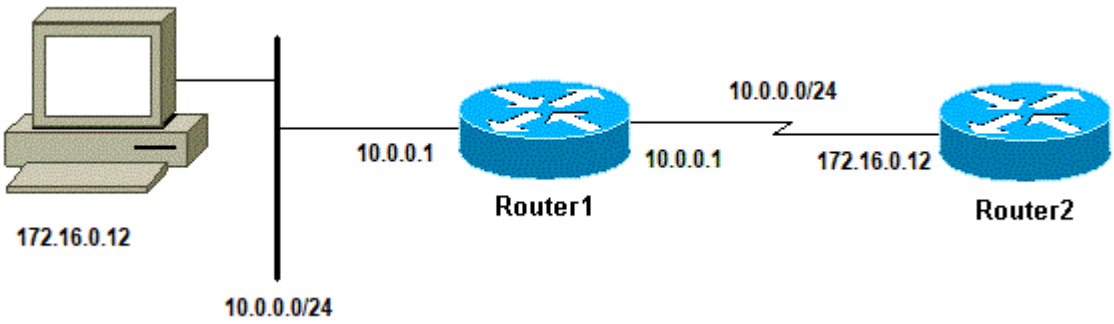
Pour plus d'informations sur la commande ping étendu, consultez Understand the Extended Ping and Extended Traceroute Commands (Comprendre les commandes Extended Ping et Extended Traceroute).

Dans l'exemple précédent, lorsque le délai d'expiration a été augmenté, l'envoi du message ping a réussi.

Remarque : La durée moyenne de l'aller-retour est supérieure à deux secondes.

Adresse source correcte

Voici un exemple de scénario courant :



Adresse source correcte

Ajoutez une interface LAN sur le routeur 1 :

<#root>

```
Router1(config)#
```

interface ethernet0

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

ip address 10.0.0.1 255.255.255.0

À partir d'une station du réseau local, vous pouvez envoyer une requête ping à Router1. À partir de Router1, vous pouvez envoyer une requête ping à Router2. Cependant, à partir d'une station du réseau local, vous ne pouvez pas envoyer de requête ping à Router2.

À partir de Router1, vous pouvez envoyer une requête ping à Router2 car, par défaut, vous utilisez l'adresse IP de l'interface sortante comme adresse source dans votre paquet ICMP. Le routeur 2 ne dispose pas d'informations sur ce nouveau réseau local. S'il doit répondre à un paquet provenant de ce réseau, il ne saura pas comment le gérer.

<#root>

```
Router1#
```

debug ip packet

```
IP packet debugging is on
```

⚠ Avertissement : Lorsque la commande debug ip packet est utilisée sur un routeur de production, elle peut entraîner une utilisation élevée du CPU. Une grave dégradation de la performance ou une panne du réseau pourrait s'ensuivre.

<#root>

```
Router1#

ping 172.16.12.2



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, 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=172.16.12.1 (local), d=172.16.12.2 (Serial0), len 100, sending
Jan 20 16:35:54.259: IP: s=172.16.12.2 (Serial0), d=172.16.12.1 (Serial0), len 100, rcvd 3
```

L'exemple de sortie précédent fonctionne parce que l'adresse source du paquet envoyé est 172.16.12.1. Pour simuler un paquet à partir du réseau local, vous devez utiliser une requête ping étendue :

<#root>

```
Router1#

ping



Protocol [ip]:
Target IP address: 172.16.12.2
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 172.16.12.2, timeout is 2 seconds:

Jan 20 16:40:18.303: IP: s=10.0.0.1 (local), d=172.16.12.2 (Serial0), len 100,
 sending.
Jan 20 16:40:20.303: IP: s=10.0.0.1 (local), d=172.16.12.2 (Serial0), len 100,
 sending.
Jan 20 16:40:22.303: IP: s=10.0.0.1 (local), d=172.16.12.2 (Serial0), len 100,
 sending.
Jan 20 16:40:24.303: IP: s=10.0.0.1 (local), d=172.16.12.2 (Serial0), len 100,
 sending
Jan 20 16:40:26.303: IP: s=10.0.0.1 (local), d=172.16.12.2 (Serial0), len 100,
 sending.
Success rate is 0 percent (0/5)
```

Cette fois, l’adresse source est 10.0.0.1, et l’envoi ne fonctionne pas. Des paquets sont envoyés, mais aucune réponse n’est reçue. Pour résoudre ce problème, ajoutez une route vers 10.0.0.0 dans Router2. La règle de base est que le périphérique ayant reçu une requête ping doit également savoir comment envoyer la réponse à la source de la requête ping.

Pertes de file d'attente entrantes élevées

Lorsqu’un paquet entre dans le routeur, le routeur tente de le transférer au niveau d’interruption. Si une correspondance ne peut pas être trouvée dans une table de cache appropriée, le paquet est mis en file d'attente dans la file d'attente d'entrée de l'interface entrante à traiter. Certains paquets sont toujours traités, mais avec la configuration appropriée et dans des réseaux stables, le taux de paquets traités ne doit jamais congestionner la file d'attente d'entrée. Si la file d'attente d'entrée est pleine, le paquet est abandonné.

Bien que l’interface soit opérationnelle, vous ne pouvez pas envoyer un message ping au périphérique en raison du nombre élevé d’abandons dans la file d’attente. Vous pouvez vérifier les suppressions d’entrées grâce à la commande show interface.

<#root>

```
Router1#

show interface Serial0/0/0


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

  MTU 1500 bytes, BW 1984 Kbit, DLY 20000 usec,
     reliability 255/255, txload 69/255, rxload 43/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:02, output 00:00:00, output hang never
  Last clearing of "show interface" counters 01:28:49
```

```
Input queue: 76/75/5553/0


 (size/max/drops/flushes);
     Total output drops: 1760
  Queueing strategy: Class-based queueing
  Output queue: 29/1000/64/1760 (size/max total/threshold/drops)
     Conversations  7/129/256 (active/max active/max total)
     Reserved Conversations 4/4 (allocated/max allocated)
     Available Bandwidth 1289 kilobits/sec


!--- Output supressed
```

Comme l'indique le résultat, la perte de la file d'attente d'entrée est élevée. Consultez la section Troubleshoot Input Queue Drops and Output Queue Drops (Dépannage des abandons dans la file d’attente des entrées et sorties).

Commande Traceroute

La commande traceroute est utilisée pour découvrir les routes que les paquets empruntent réellement pour se rendre à leur destination. Le périphérique (par exemple, un routeur ou un PC) envoie une séquence de datagrammes UDP (User Datagram Protocol) à une adresse de port non valide sur l’hôte distant.

Trois datagrammes sont envoyés, chacun avec une valeur de champ de durée de vie (TTL) égale à un. La valeur TTL de 1 provoque le « délai d'attente » du datagramme dès qu'il atteint le premier routeur sur le chemin ; ce routeur répond alors par un message ICMP de dépassement de délai (TEM) qui indique que le datagramme a expiré.

Trois autres messages UDP sont maintenant envoyés, chacun avec la valeur TTL définie sur 2, ce qui entraîne le retour des messages ICMP TEM par le deuxième routeur. Ce processus se poursuit jusqu’à ce que les paquets atteignent l’autre destination. Puisque ces datagrammes tentent d'accéder à un port non valide sur l'hôte de destination, les messages d'inaccessibilité de port ICMP sont retournés, et indiquent un port inaccessible ; cet événement signale au programme Traceroute qu'il a terminé.

L’objectif de cette opération est d’enregistrer la source de chaque message ICMP de dépassement de délai afin de fournir une trace du chemin emprunté par le paquet pour atteindre la destination.

```
Router1#
```

traceroute 172.16.34.4

```
Type escape sequence to abort.
Tracing the route to 172.16.34.4

  1 172.16.12.2 4 msec 4 msec 4 msec
  2 172.16.23.3 20 msec 16 msec 16 msec
  3 172.16.34.4 16 msec *  16 msec

Jan 20 16:42:48.611: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.615:     UDP src=39911, dst=
```

33434

```
Jan 20 16:42:48.635: IP: s=172.16.12.2 (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=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.647:     UDP src=34237, dst=33435
Jan 20 16:42:48.667: IP: s=172.16.12.2 (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=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.679:     UDP src=33420, dst=33436
Jan 20 16:42:48.699: IP: s=172.16.12.2 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:48.703:     ICMP type=11, code=0
```

Il s’agit de la première séquence de paquets envoyée avec un TTL=1. Le premier routeur, dans ce cas Router2 (172.16.12.2), abandonne le paquet et renvoie à la source (172.16.12.1) un message ICMP de type=11. Cela correspond au message de dépassement de délai.

<#root>

```
Jan 20 16:42:48.707: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.711:     UDP src=35734, dst=33437
Jan 20 16:42:48.743: IP: s=
```

172.16.23.3

```
 (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=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.755:     UDP src=36753, dst=33438
Jan 20 16:42:48.787: IP: s=172.16.23.3 (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=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.799:     UDP src=36561, dst=33439
Jan 20 16:42:48.827: IP: s=172.16.23.3 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:48.831:     ICMP type=11, code=0
```

Le même processus se produit pour le routeur 3 (172.16.23.3) avec une durée de vie de 2 :

<#root>

```
Jan 20 16:42:48.839: IP: s=172.16.12.1 (local), d=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:48.843:     UDP src=34327, dst=33440
Jan 20 16:42:48.887: IP: s=
```

172.16.34.4

```
 (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=172.16.34.4 (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=172.16.34.4 (Serial0), len 28,
 sending
Jan 20 16:42:51.899:     UDP src=37181, dst=33442
Jan 20 16:42:51.943: IP: s=172.16.34.4 (Serial0), d=172.16.12.1 (Serial0), len 56,
 rcvd 3
Jan 20 16:42:51.947:     ICMP type=3, code=3
```

Avec une TTL = 3, le routeur 4 est enfin atteint. Cette fois, puisque le port n'est pas valide, le routeur 4 renvoie au routeur 1 un message ICMP avec le type=3, un message Destination inaccessible et le code=3, ce qui signifie que le port est inaccessible.

Le tableau suivant répertorie les caractères qui peuvent figurer dans la sortie de la commande traceroute.

Caractères de texte IP Traceroute

Caractère	Description
nn msec	Pour chaque noeud, temps de propagation aller-retour en millisecondes pour le nombre de sondes spécifié
*	Le délai de la sonde a expiré
A	Interdiction administrative (par exemple, access-list)
Q	Épuisement de la source (destination trop occupée)
I	Test interrompu par l'utilisateur
U	Port inaccessible
H	Hôte inacessible
n	Réseau inaccessible
P	Protocole inaccessible
T	Timeout (Délai d’expiration)
?	Type de paquet inconnu

rendement

Vous pouvez obtenir le temps aller-retour (RTT) grâce aux commandes ping et traceroute. Il s’agit du temps nécessaire pour envoyer un paquet écho et obtenir une réponse. Cela peut donner une idée approximative du délai sur la liaison. Toutefois, ces chiffres ne sont pas assez précis pour être utilisés pour l'évaluation du rendement.

Lorsqu’un paquet est destiné au routeur lui-même, ce paquet doit être commuté par processus. Le processeur doit gérer les renseignements de ce paquet et renvoyer une réponse. Ce n'est pas l'objectif principal d'un routeur. Par définition, un routeur est conçu pour acheminer des paquets. Une réponse au message ping est offerte dans le cadre du service du meilleur effort.

Pour illustrer cette situation, voici un exemple de message ping envoyé du routeur 1 au routeur 2 :

<#root>

```
Router1#
```

```
ping 172.16.12.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

Le RTT est d'environ quatre millisecondes. Après avoir activé certaines fonctionnalités à forte intensité de processus sur le routeur 2, essayez d'envoyer une requête ping au routeur 2 à partir du routeur 1.

<#root>

```
Router1#
```

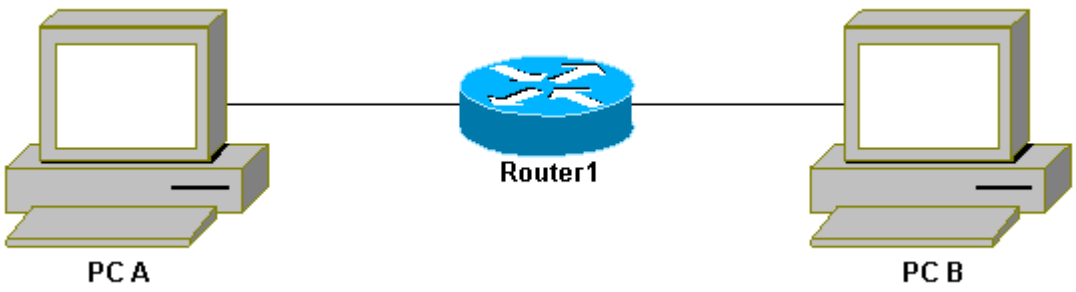
```
ping 172.16.12.2
```

```
Type
```

```
escape sequence
```

```
 to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/25/28 ms
```

La RTT a considérablement augmenté ici. Le routeur 2 est très occupé, et la priorité n'est pas de répondre au message ping. Une meilleure façon de tester la performance d'un routeur est d'utiliser le trafic qui circule par ce routeur.



Trafic passant par le routeur

Le trafic est ensuite commuté rapidement et géré par le routeur ayant la plus haute priorité. Le

réseau de base l’illustre :



Réseau de base avec 3 routeurs

Envoyez un message ping au routeur 3 à partir du routeur 1 :

<#root>

```
Router1#

ping 172.16.23.3



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
```

Le trafic passe par le routeur 2 et est maintenant à commutation rapide. Activez la fonctionnalité à processus intensif sur le routeur 2 :

<#root>

```
Router1#

ping 172.16.23.3



Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
```

Il n'y a presque aucune différence. En effet, sur le routeur 2, les paquets sont maintenant traités au niveau d’interruption.

Utiliser la commande Debug

Avant d'utiliser les commandes debug, référez-vous à Informations importantes sur les commandes de débogage .

Les différentes commandes debug utilisées dans cet article montrent ce qui se passe lorsqu'une commande Ping ou Traceroute est utilisée. Ces commandes peuvent vous aider à résoudre des problèmes. Toutefois, dans un environnement de production, les commandes debug doivent être utilisées avec prudence. Si votre processeur n'est pas puissant, ou si vous avez beaucoup de paquets à commutation de processus, ils peuvent facilement bloquer votre périphérique. Il existe plusieurs façons de minimiser l'impact de la commande debug sur le routeur. Une méthode consiste à utiliser des listes d'accès pour limiter le trafic spécifique que vous souhaitez surveiller.

Voici un exemple :

<#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 172.16.34.4

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 172.16.12.1 host 172.16.34.4 (5 matches)
```

Avec cette configuration, Router4 imprime uniquement le message de débogage qui correspond à la liste de contrôle d'accès 150. Une requête ping envoyée par Router1 entraîne l'affichage de ce message :

```
Router4#
Jan 20 16:51:16.911: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:51:17.003: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:51:17.095: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:51:17.187: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:51:17.279: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
```

La réponse au problème ne vient pas du routeur 4, car ces paquets ne correspondent pas à la liste d'accès. Pour les afficher, ajoutez :

<#root>

```
Router4(config)#

access-list 150 permit ip host 172.16.12.1 host 172.16.34.4

Router4(config)#

access-list 150 permit ip host 172.16.34.4 host 172.16.12.1
```

Résultats :

```
Jan 20 16:53:16.527: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
```

```
 rcvd 3
Jan 20 16:53:16.531: IP: s=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.627: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.635: IP: s=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.727: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.731: IP: s=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.823: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.827: IP: s=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:53:16.919: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
Jan 20 16:53:16.923: IP: s=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
```

Un autre moyen de réduire l'incidence de la commande debug est de mettre les messages de débogage en mémoire tampon et de les afficher avec la commande show log une fois le débogage désactivé :

<#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=172.16.34.4 (local), d=172.16.12.1 (Serial0), len 100,
 sending
Jan 20 16:55:46.679: IP: s=172.16.12.1 (Serial0), d=172.16.34.4 (Serial0), len 100,
 rcvd 3
```

Les commandes ping et traceroute fournissent des informations de base sur l'accessibilité et le chemin que les ingénieurs réseau utilisent généralement pour résoudre les problèmes d'accès au réseau.

Informations connexes

- Comprendre les commandes Ping et Traceroute étendues
- Support technique - Cisco Systems

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.