

Troubleshooting Windows Network Connectivity

Brien M. Posey

Using ping for Troubleshooting Windows Network Connectivity

Using ping command for troubleshooting networks will help narrow down the causes of your Windows PC connectivity problems. The introduction to this TCP/IP diagnostic utility will give you an understanding and syntax of how ping works, plus what it means when your ping request times out or reaches a network host.

Diagnose issues from the command line (CL) prompt window and learn how to verify network connections in this tip, originally seen on WindowsNetworking.com.

Today's network hardware and software are more reliable than ever; but even so, things do occasionally go wrong. In this article series, I am going to discuss some troubleshooting techniques that you can use when a host on your Windows network has trouble communicating with other network hosts. For the sake of those with less experience in working with the TCP/IP protocol, I'm going to start with the basics, and then work toward the more advanced techniques.

Verify network connectivity

When one host has trouble communicating with another, the first thing that you must do is gather some information about the problem. More specifically, you need to document the host's configuration, find out if the host is having trouble communicating with any other machines on the network, and find out if the problem effects any other hosts.

For example, suppose that a workstation is having trouble communicating with a particular server. That in itself doesn't really give you a lot to go on. However, if you were to dig a little bit deeper into the problem and found out that the workstation couldn't communicate with any of the network servers, then you would know to check for a disconnected network cable, a bad switch port, or maybe a network configuration problem.

Likewise, if the workstation were able to communicate with some of the network servers, but not all of them, that too would give you a hint as to where to look for the problem. In that type of situation, you would probably want to check to see what the servers that could not be contacted had in common. Are they all on a common subnet? If so, then a routing problem is probably to blame.

If multiple workstations are having trouble communicating with a specific server, then the problem probably isn't related to the workstations unless those workstations were recently reconfigured. More than likely, it is the server itself that is malfunctioning.

The point is that by starting out with a few basic tests, you can gain a lot of insight into the problem at hand. The tests that I am about to show you will rarely show you the cause of the problem, but they will help to narrow things down so that you will know where to begin the troubleshooting process.

Ping

Ping is probably the simplest TCP/IP diagnostic utility ever created, but the information that it can provide you with is invaluable. Simply put, ping tells you whether or not your workstation can communicate with another machine.

The first thing that I recommend doing is opening a command prompt window, and then entering the ping command, followed by the IP address of the machine that you are having trouble communicating

Troubleshooting Windows Network Connectivity

Brien M. Posey

with. When you do, the machine that you have specified should produce four replies, as shown in Figure A.



```
e:\ Command Prompt
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator\FUBAR>ping 147.100.100.99

Pinging 147.100.100.99 with 32 bytes of data:

Reply from 147.100.100.99: bytes=32 time<1ms TTL=128
Reply from 147.100.100.99: bytes=32 time<1ms TTL=128
Reply from 147.100.100.99: bytes=32 time<1ms TTL=128
Reply from 147.100.100.99: bytes=32 time<1ms TTL=128

Ping statistics for 147.100.100.99:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\Administrator\FUBAR>
```

Figure A
Four replies generated after pinging specified machine

The responses essentially tell you how long it took the specified machine to respond with 32 bytes of data. For example, in Figure A, each of the four responses were received in less than four milliseconds.

Typically, when you issue the ping command, one of four things will happen, each of which has its own meaning:

The first thing that can happen is that the specified machine will produce four replies. This indicates that the workstation is able to communicate with the specified host at the TCP/IP level.

The second thing that can happen is that all four requests time out, as shown in Figure B. If you look at Figure A, you will notice that each response ends in TTL=128. TTL stands for "time to live." What this means is that each of the four queries and responses must be completed within 128 milliseconds. The TTL is also decremented once for each hop on the way back. A hop occurs when a packet moves from one network to another. I will be talking a lot more about hops later on in this series.

Troubleshooting Windows Network Connectivity

Brien M. Posey



```
C:\Documents and Settings\Administrator\FUBAR>ping 147.100.100.0
Pinging 147.100.100.0 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 147.100.100.0:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Documents and Settings\Administrator\FUBAR>
```

Figure B
Four requests time out after pinging host

At any rate, if all four requests have timed out, it means that the TTL expired before the reply was received. This can mean one of three things:

- Communication problems are preventing packets from flowing between the two machines: This could be caused by a disconnected cable, a bad routing table, or a number of other issues.
- Communications are occurring, but are too slow for ping to acknowledge: This can be caused by extreme network congestion, or by faulty network hardware or wiring.
- Communications are functional, but a firewall is blocking ICMP traffic: Ping will not work unless the destination machine's firewall (and any firewalls between the two machines) allow ICMP echoes.

A third thing that can happen when you enter the ping command is that some replies are received, while others time out. This can point to bad network cabling, faulty hardware, or extreme network congestion.

The fourth thing that can occur when pinging a host is that you receive an error similar to the one that is shown in Figure C.

Troubleshooting Windows Network Connectivity

Brien M. Posey



```
Administrator: Command Prompt
C:\Users\Administrator>ping 147.100.100.99
Pinging 147.100.100.99 with 32 bytes of data:
PING: transmit failed, error code 1231.
PING: transmit failed, error code 1231.
PING: transmit failed, error code 1231.
PING: transmit failed, error code 1231.
Ping statistics for 147.100.100.99:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\Users\Administrator>
```

Figure C
Error message indicating that TCP/IP is misconfigured

A Transmit Failed error indicates that TCP/IP is not configured correctly on the machine on which you are trying to enter the ping command. This particular error is specific to Vista though. Older versions of Windows produce an error when TCP/IP is configured incorrectly, but the error message is "Destination Host Unreachable."

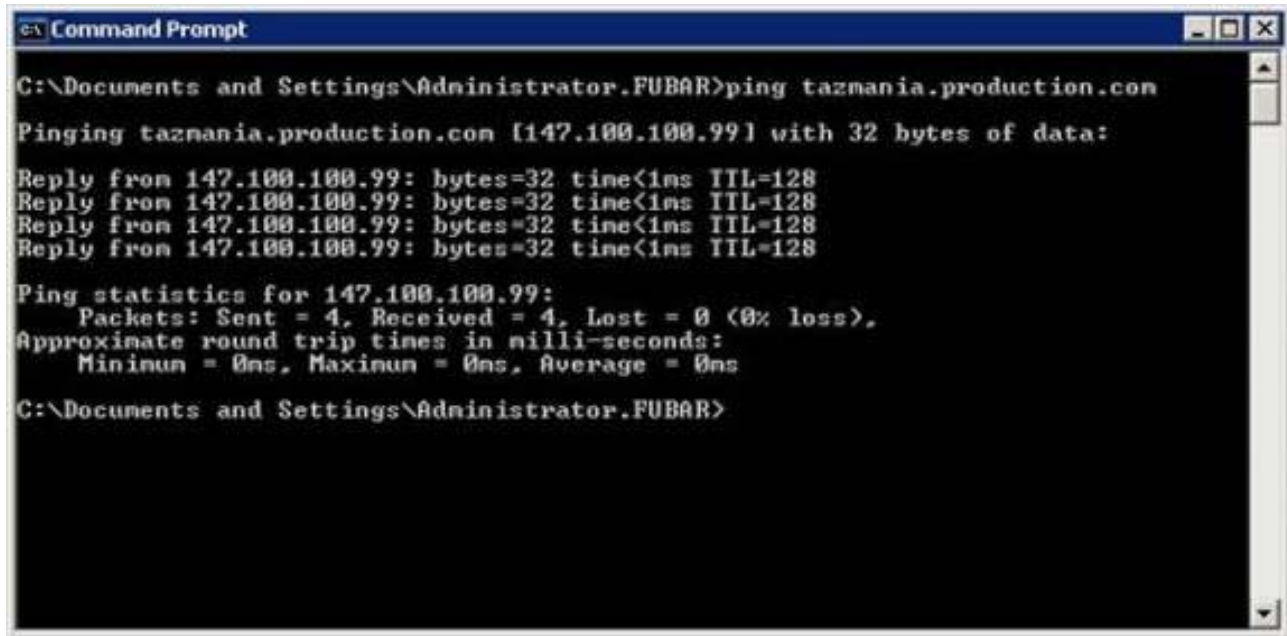
What if the ping is successful?

Believe it or not, it is not uncommon for a ping to succeed, even though two machines are having trouble communicating with each other. If this happens, it means that the underlying network infrastructure is good, and that the machines are able to communicate at the TCP/IP level. Typically, this is good news, because it means that the problem that is occurring is not very serious.

If normal communications between two machines are failing, but the two machines can ping each other successfully (be sure to run the ping command from both machines), then there is something else that you can try. Rather than pinging the network host by IP address, try replacing the IP address with the host's fully qualified domain name, as shown in Figure D.

Troubleshooting Windows Network Connectivity

Brien M. Posey



```
c:\ Command Prompt
C:\Documents and Settings\Administrator.FUBAR>ping tazmania.production.com
Pinging tazmania.production.com [147.100.100.99] with 32 bytes of data:
Reply from 147.100.100.99: bytes=32 time<1ms TTL=128
Reply from 147.100.100.99: bytes=32 time<1ms TTL=128
Reply from 147.100.100.99: bytes=32 time<1ms TTL=128
Reply from 147.100.100.99: bytes=32 time<1ms TTL=128

Ping statistics for 147.100.100.99:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\Administrator.FUBAR>
```

Figure D
Pinging network host by its fully qualified domain name

If you are able to ping the machine by its IP address, but not by its fully qualified domain name, then you most likely have a domain name system (DNS) issue. The workstation may be configured to use the wrong DNS server, or the DNS server may not contain a host record for the machine that you are trying to ping.

If you look at Figure D, you can see that the machine's IP address is listed just to the right of its fully qualified domain name. This proves that the machine was able to resolve the fully qualified domain name. Make sure that the IP address that the name was resolved to is correct. If you see a different IP address than the one you expected, then you may have an incorrect DNS host record.

Conclusion

In this article, I have shown you some steps for testing basic connectivity between two machines by using ping commands, and I showed you what those ping responses could mean to your hosts connectivity. In the next article in the series, I will show you some more techniques that you can use in the troubleshooting process.

Troubleshooting Windows Network Connectivity

Brien M. Posey

Checking IP Configuration To Troubleshoot Windows Network Connectivity

In the first article in this series, Using ping command to begin troubleshooting Windows network connectivity, I started out by showing you how to use the ping command to perform some basic connectivity tests, and then talked about how you can interpret the results. In this article, I want to continue the discussion by showing you some more simple tests that you can use to diagnose the current state of connectivity.

Introductory Note

As I explained in the first part of this article series, my goal is to create a troubleshooting guide that anyone with basic skills can follow. That being the case, I am starting with basic troubleshooting techniques, and as the series progresses, I will gradually move into more advanced techniques.

Confirming Connectivity

In the previous article, I showed you the basics of using the ping command to test network connectivity. However, if you are having trouble communicating with other hosts on the network, or hosts on remote networks, then there are a few more ping tests that you can perform in order to get a better idea of what's going on.

Before I show you those techniques though, it is important to understand how the host that is having communications problems is configured. The procedure for doing so varies from one version of Windows to the next, so I will show you how to check the network configuration on a machine that's running Windows Server 2003.

The first thing that you must do is to determine whether the machine in question is running a static or a dynamic IP address configuration. To do so, open the Control Panel, and choose the Network Connections option. Now, right click on the connection that you are trying to diagnose, and choose the Properties command from the resulting shortcut menu. Upon doing so, you will see the connection's properties sheet, as shown in Figure A.

Troubleshooting Windows Network Connectivity

Brien M. Posey

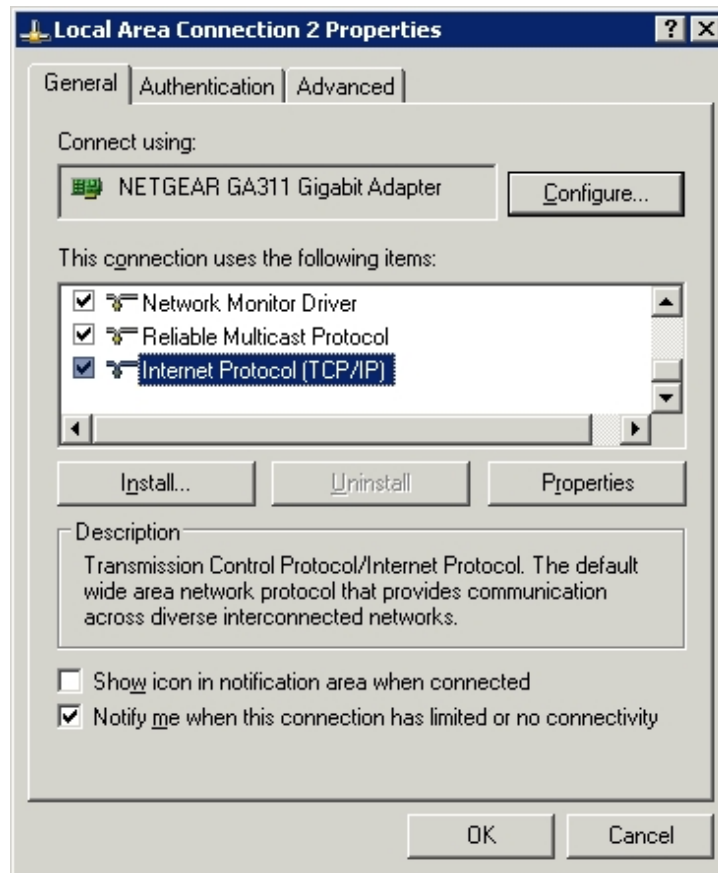


Figure A
This is the network connection's properties sheet

Now, scroll through the list of items that the connection uses until you locate the TCP/IP protocol (selected in Figure A). Select this protocol, and click the Properties button to reveal the Internet Protocol (TCP/IP) Properties sheet, shown in Figure B.

Troubleshooting Windows Network Connectivity

Brien M. Posey

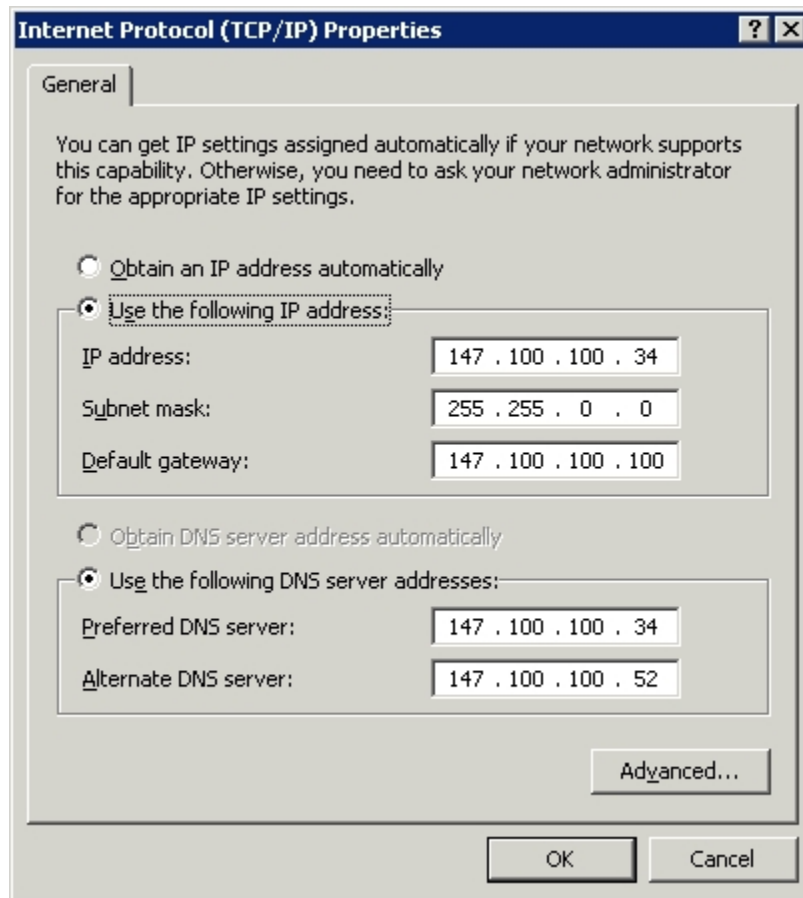


Figure B

The Internet Protocol (TCP/IP) Properties sheet is used to configure the TCP/IP protocol

Once you arrive at this screen, it is important to make note of the machine's IP configuration. Specifically, you will want to make note of the following items:

- Is the machine using a static or a dynamic configuration?
- If a static configuration is being used, what is the IP address, subnet mask and default gateway?
- Is the DNS server address being obtained automatically?
- If the DNS server address is being manually specified, what address is being used?

Before I move on, I also want to mention that if a computer has multiple network adapters installed, then there will be multiple connections that are listed in the Control Panel. It is very important that you know which connection corresponds to which network adapter, or else the techniques that I am about to show you will not work.

If you have any doubt as to which connection corresponds to which network adapter, then check the adapter type. If you look at Figure A, you will notice that the adapter type is listed at the top of the screen. If need be, you can open the case to see which network adapter the network cable is connected to, so that you can be absolutely sure that you are looking at the correct network connection.

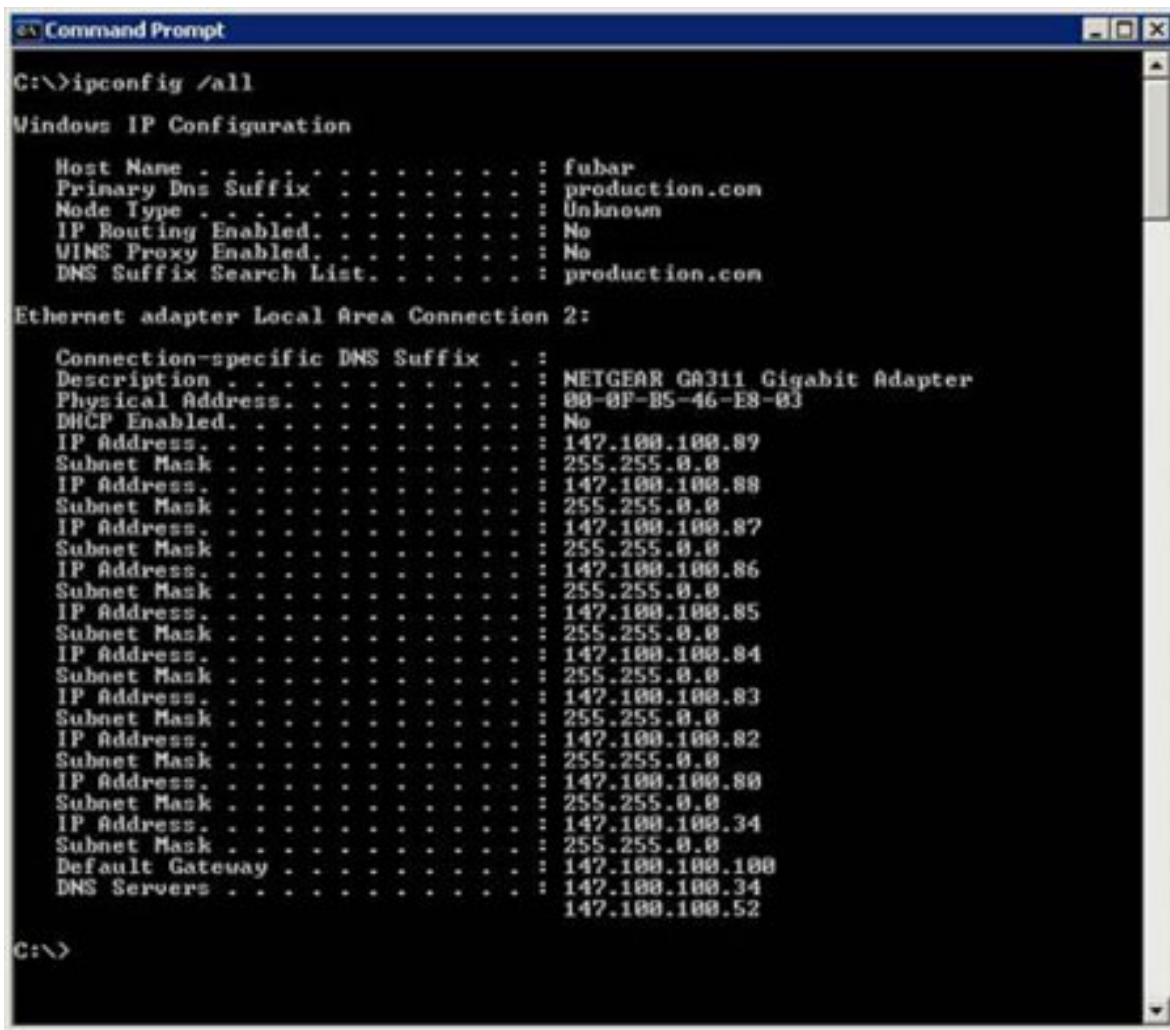
Troubleshooting Windows Network Connectivity

Brien M. Posey

Now that you know how TCP/IP is configured for the network adapter in question, we must determine whether or not Windows acknowledges the configuration. To do so, open a command prompt window, and enter the following command:

```
IPCONFIG /ALL
```

It might seem strange to have to make sure that Windows acknowledges your configuration, but ipconfig can really tell you a lot about what's going on. For example, take a look at the screen that's shown in Figure C. When you enter the ipconfig /all command, the first thing that you must do is to locate the correct network adapter. In this case, locating the correct adapter is easy, because only one adapter is listed. Notice though that ipconfig provides you with the connection number (in this case it's Ethernet adapter Local Area Connection 2). If you look back at Figure A, you will notice that the title of the properties sheet shown in the figure bears the same name. That, along with the description of the physical network adapter, tells you exactly which network connection you are looking at.



```
Command Prompt
C:\>ipconfig /all

Windows IP Configuration

Host Name . . . . . : fubar
Primary Dns Suffix . . . . . : production.com
Node Type . . . . . : Unknown
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : production.com

Ethernet adapter Local Area Connection 2:

Connection-specific DNS Suffix . : 
Description . . . . . : NETGEAR GA311 Gigabit Adapter
Physical Address. . . . . : 00-0F-B5-46-E8-03
Dhcp Enabled. . . . . : No
IP Address. . . . . : 147.100.100.89
Subnet Mask . . . . . : 255.255.0.0
IP Address. . . . . : 147.100.100.88
Subnet Mask . . . . . : 255.255.0.0
IP Address. . . . . : 147.100.100.87
Subnet Mask . . . . . : 255.255.0.0
IP Address. . . . . : 147.100.100.86
Subnet Mask . . . . . : 255.255.0.0
IP Address. . . . . : 147.100.100.85
Subnet Mask . . . . . : 255.255.0.0
IP Address. . . . . : 147.100.100.84
Subnet Mask . . . . . : 255.255.0.0
IP Address. . . . . : 147.100.100.83
Subnet Mask . . . . . : 255.255.0.0
IP Address. . . . . : 147.100.100.82
Subnet Mask . . . . . : 255.255.0.0
IP Address. . . . . : 147.100.100.80
Subnet Mask . . . . . : 255.255.0.0
IP Address. . . . . : 147.100.100.34
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : 147.100.100.100
DNS Servers . . . . . : 147.100.100.34
                        147.100.100.52

C:\>
```

Figure B

The ipconfig /all command shows you the machine's IP configuration as Windows sees it.

Troubleshooting Windows Network Connectivity

Brien M. Posey

Of course the first thing that you will probably notice about Figure C is that it lists many different IP addresses for the connection. The reason for this is that I created the screenshot on a Web server. The Web server hosts multiple websites, each with its own IP address. I wanted to use this server to illustrate the point that the IP address configuration that you see when you glance at the TCP/IP properties sheet isn't always what Windows is using. In this case, the IP configuration information shown in Figure B is still valid. It serves as the machine's primary IP address. However, there are many other IP addresses that are also in use.

The next step in the troubleshooting process varies depending on whether the machine is using a static or a dynamic IP address configuration. If the machine is using a static configuration, then for right now, just check to make sure that the IP address, subnet mask, default gateway, and DNS server address that is listed matches those entered on the TCP/IP Properties sheet.

If the machine is using a dynamic IP address, then you will want to look at the address and see if it falls within the expected address range. If you are troubleshooting a problem on an unfamiliar network, then you may not know what the address range should be. If that's the case, there are a few values that you can look for that have special meanings.

The most obvious clue that something has gone wrong is an IP address of 0.0.0.0. The presence of this address usually indicates one of three things:

- The network adapter is not connected to the network (possibly because of a cable problem or a bad switch port).
- The IP address was released.
- An IP address conflict has occurred.

If you receive this address, then try entering the following three commands:

```
IPCONFIG /RELEASE  
IPCONFIG /RENEW  
IPCONFIG /ALL
```

These commands will essentially tell the computer to give up its current address, try to obtain a new address, and then show you the new configuration information. Sometimes this process will fix the problem, and sometimes it won't. Often though, it will yield clues as to the cause of the problem.

Another telltale clue that something has gone wrong is that the IP address falls into the 169.254.x.x range with a subnet mask of 255.255.0.0. Some versions of Windows will automatically use this address if an IP address cannot be acquired from a DHCP server.

Conclusion

In this article, I began showing you how to examine a machine's IP address configuration for possible clues to the cause of the problem. In the next article in the series, I will show you how to use the configuration information that you've found to test for network connectivity.

Troubleshooting Windows Network Connectivity

Brien M. Posey

Test Your TCP/IP Protocol Stack To Troubleshoot Network Connectivity

In the previous article in this series, Checking IP configuration, I showed you how to determine which IP address your system is using as its primary address. The next step in the process is to verify that the IP address configuration is working correctly, and that there are no problems with the local TCP/IP protocol stack.

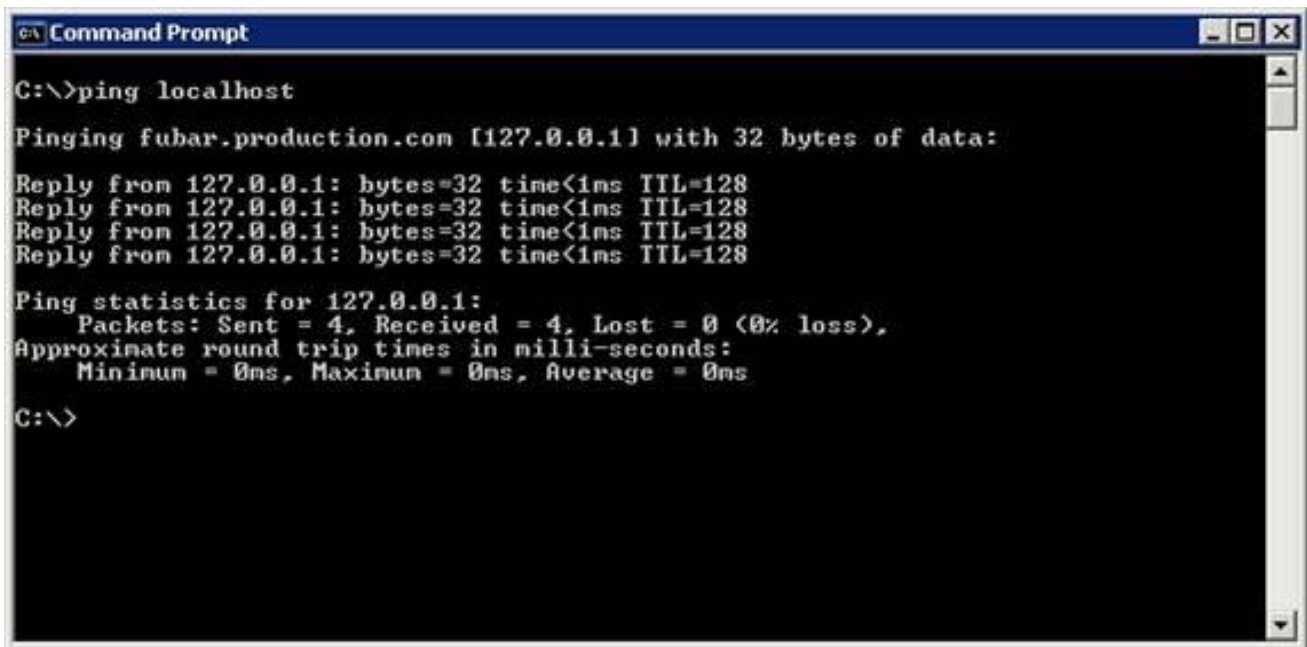
The first test that you need to perform is to ping the local host address. There are a couple of different ways of accomplishing this. One way is to enter the following command:

```
PING LOCALHOST
```

When you enter this command, Windows will ping the address 127.0.0.1. Regardless of your machine's IP address, Windows will always use 127.0.0.1 as the local host address. Therefore, an alternative to the command listed above is to simply enter the following command:

```
Ping 127.0.0.1
```

Upon entering this command, you should see a successful ping, just as you would with any other ping command. You can see an example of this, shown in Figure A.



```
Command Prompt
C:\>ping localhost

Pinging fubar.production.com [127.0.0.1] with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

Figure A
Receiving a successful ping when attempting to ping the local host address

Pinging the local host address does nothing to diagnose communication problems with a remote host. It does however allow you to confirm that your local TCP/IP stack is functioning correctly. If you ping the local host address and receive a destination host unreachable error message, it is almost always an indication that TCP/IP is configured incorrectly, or that some part of the local TCP/IP stack is damaged.

Troubleshooting Windows Network Connectivity

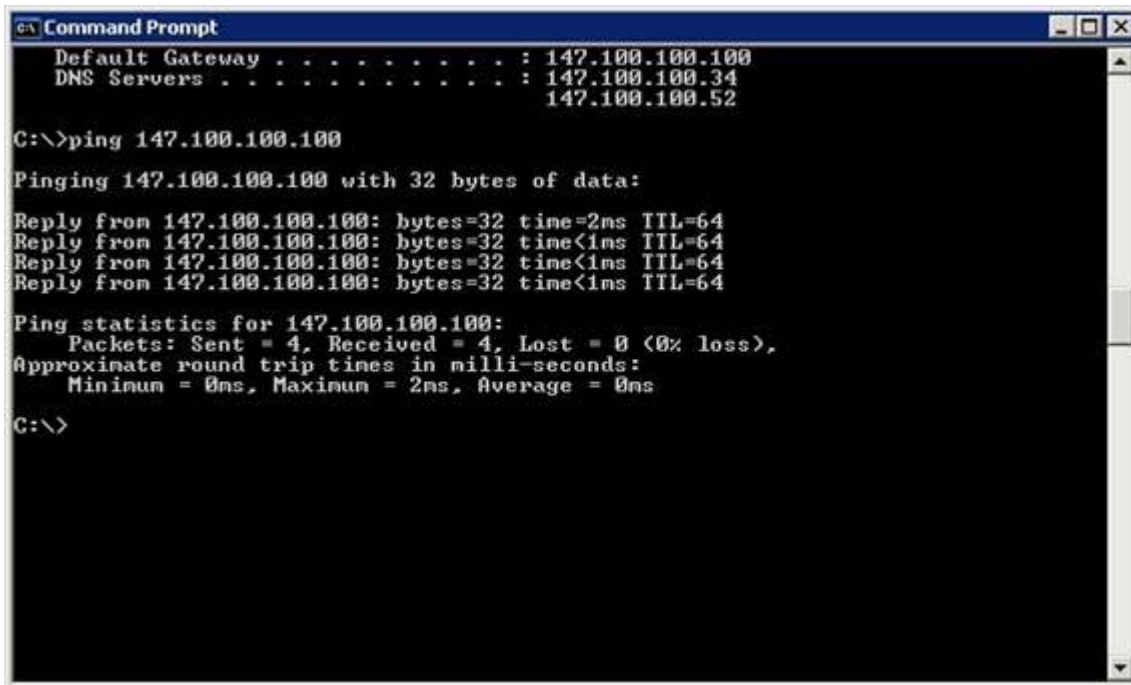
Brien M. Posey

It has been my experience that you can usually get around this problem by removing the TCP/IP protocol from the computer, and then reintroducing it from scratch.

Ping the Default Gateway

In the previous part of this article series, I mentioned that there were several different aspects of the TCP/IP configuration that you needed to document, and have on hand for the troubleshooting process. Among these pieces of information are the IP addresses of the default gateway and of the primary DNS server.

Assuming that the hosts that you're trying to communicate with is on a remote network, or on a different segment of your corporate network, then the next thing that you need to attempt is to ping the default gateway. You can accomplish this by simply appending the default gateway's IP address to the ping command. For example, if you look at Figure B, you will notice that my TCP/IP configuration lists my default gateway address as 147.100.100.100. I then simply pinged this address. This verifies that the local machine can communicate with the default gateway. It also tells you that communications on the local network are working as intended, at least at the IP address level.



```
Command Prompt
Default Gateway . . . . . : 147.100.100.100
DNS Servers . . . . . : 147.100.100.34
                       147.100.100.52

C:\>ping 147.100.100.100

Pinging 147.100.100.100 with 32 bytes of data:

Reply from 147.100.100.100: bytes=32 time=2ms TTL=64
Reply from 147.100.100.100: bytes=32 time<1ms TTL=64
Reply from 147.100.100.100: bytes=32 time<1ms TTL=64
Reply from 147.100.100.100: bytes=32 time<1ms TTL=64

Ping statistics for 147.100.100.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\>
```

Figure B
Default gateway ping verifying that IP packets can reach your network's default gateway

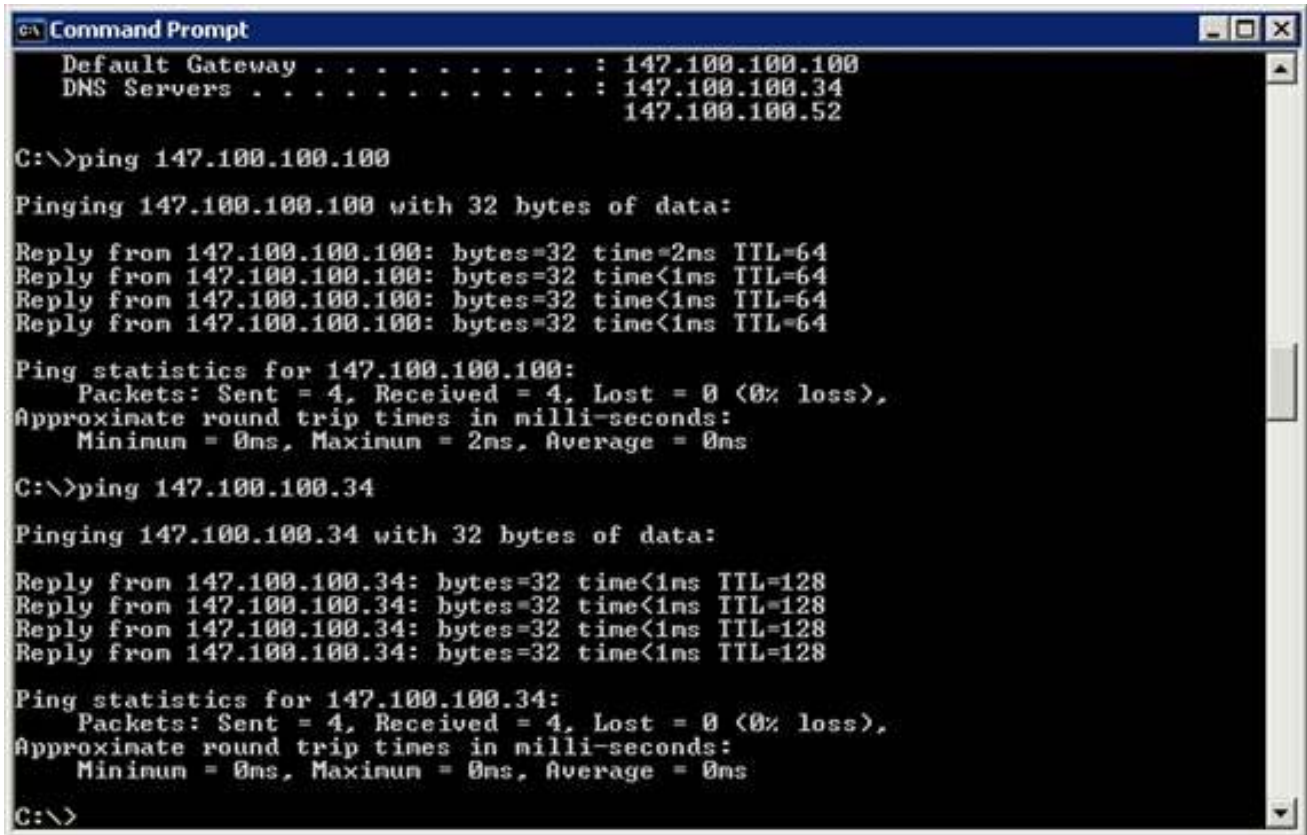
Ping the DNS Server

So far we have established that IP level communications are working between the local computer and the default gateway. This does not however guarantee that host names are being resolved to IP addresses. In the first part of the article series, Using ping command to troubleshoot network connectivity, I showed you how you could use the destination host's fully qualified domain name in conjunction with the ping command as a way of verifying that the DNS server is doing its job. There are a couple of other ways that you can easily test DNS name resolution though.

Troubleshooting Windows Network Connectivity

Brien M. Posey

One thing that you can do is ping the DNS server's IP address, as shown in Figure C. This does not guarantee the name resolution is working correctly, but it does verify that the local machine is able to communicate with the DNS server.



```
C:\ Command Prompt
Default Gateway . . . . . : 147.100.100.100
DNS Servers . . . . . : 147.100.100.34
                       147.100.100.52

C:\>ping 147.100.100.100

Pinging 147.100.100.100 with 32 bytes of data:

Reply from 147.100.100.100: bytes=32 time=2ms TTL=64
Reply from 147.100.100.100: bytes=32 time<1ms TTL=64
Reply from 147.100.100.100: bytes=32 time<1ms TTL=64
Reply from 147.100.100.100: bytes=32 time<1ms TTL=64

Ping statistics for 147.100.100.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\>ping 147.100.100.34

Pinging 147.100.100.34 with 32 bytes of data:

Reply from 147.100.100.34: bytes=32 time<1ms TTL=128
Reply from 147.100.100.34: bytes=32 time<1ms TTL=128
Reply from 147.100.100.34: bytes=32 time<1ms TTL=128
Reply from 147.100.100.34: bytes=32 time<1ms TTL=128

Ping statistics for 147.100.100.34:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

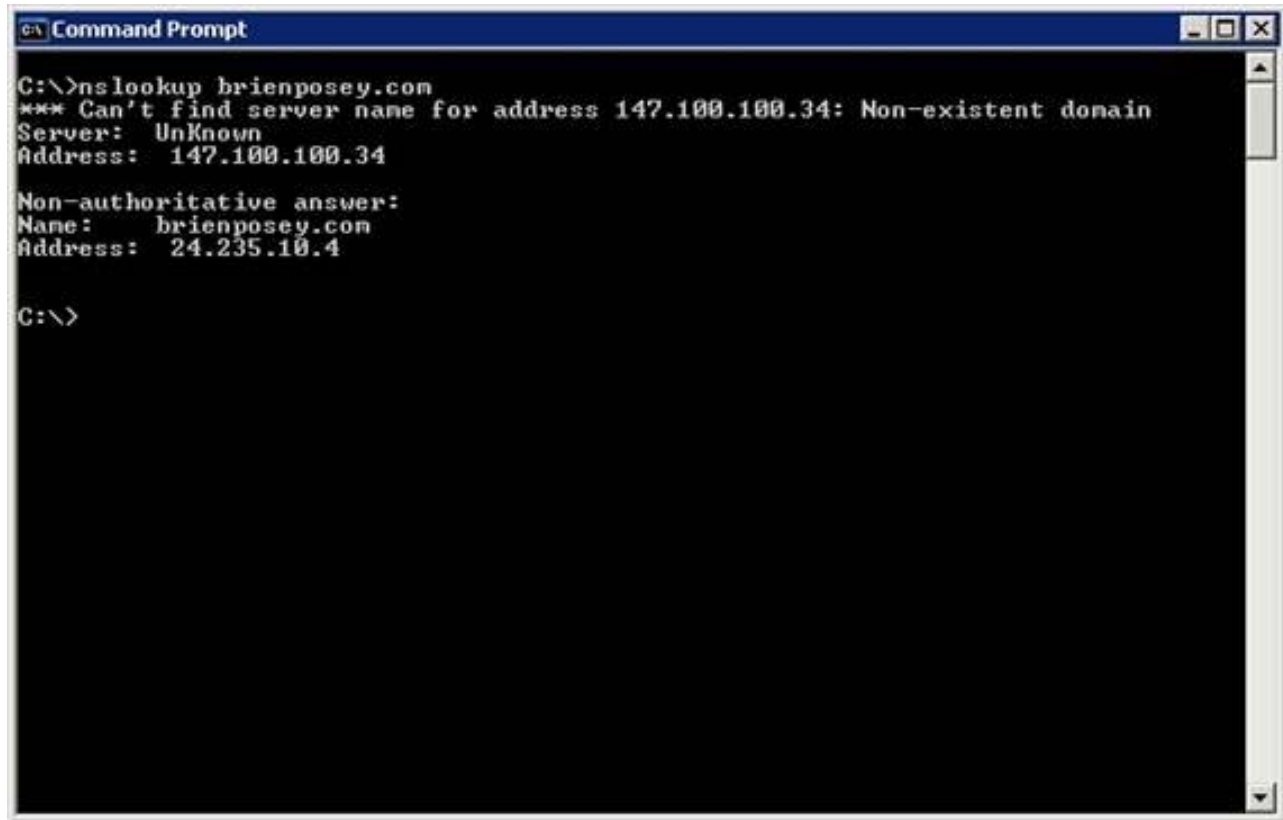
C:\>
```

Figure C
Verify that the host can communicate with your DNS server

Another option is to use the Nslookup command to verify that name resolution is working properly. To do so, simply enter the Nslookup command, followed by the remote host's fully qualified domain name. The Nslookup command should be able to resolve the fully qualified domain name to an IP address, as shown in Figure D.

Troubleshooting Windows Network Connectivity

Brien M. Posey



```
C:\>nslookup brienposey.com
*** Can't find server name for address 147.100.100.34: Non-existent domain
Server:    Unknown
Address:   147.100.100.34

Non-authoritative answer:
Name:     brienposey.com
Address:  24.235.10.4

C:\>
```

Figure D

The Nslookup command showing whether or not your DNS server can resolve the host name

The image above can be a bit misleading at first, if you are not used to working with Nslookup. Initially, this screen appears to be reporting an error. If you take a closer look though, you can see that the first part of the information that has been returned refers to the local DNS server. You can tell this because the IP address that is referenced matches the DNS server's IP address. However, the lower section of the returned information provides you with the IP address of the host that you have queried. As long as this IP address is listed, then the DNS query was successful.

If the name resolution process fails, then there is a DNS problem. The actual problem may be any one of a number of different problems with the DNS server. For example, the DNS servers forwarding address may not be correct, or the DNS server may not have access to the Internet, which it needs in order to contact higher level DNS servers. Likewise, the DNS server's DNS service may have stopped. Typically though, these types of problems will affect other clients as well since multiple clients usually rely on a single DNS server.

If DNS name resolution succeeds, it is important that you've verified the IP address was returned during the name resolution process. You can do this by comparing the IP address of the returned to the actual IP address that the remote host is using. These IP addresses should match, but there are conditions that could cause a mismatch, which would result in a communication failure.

If you do encounter an IP address mismatch, it could be the result of a malware infestation on the client, or it could be the result of DNS poisoning. DNS poisoning is a process in which the DNS cache is populated with invalid or incorrect IP addresses.

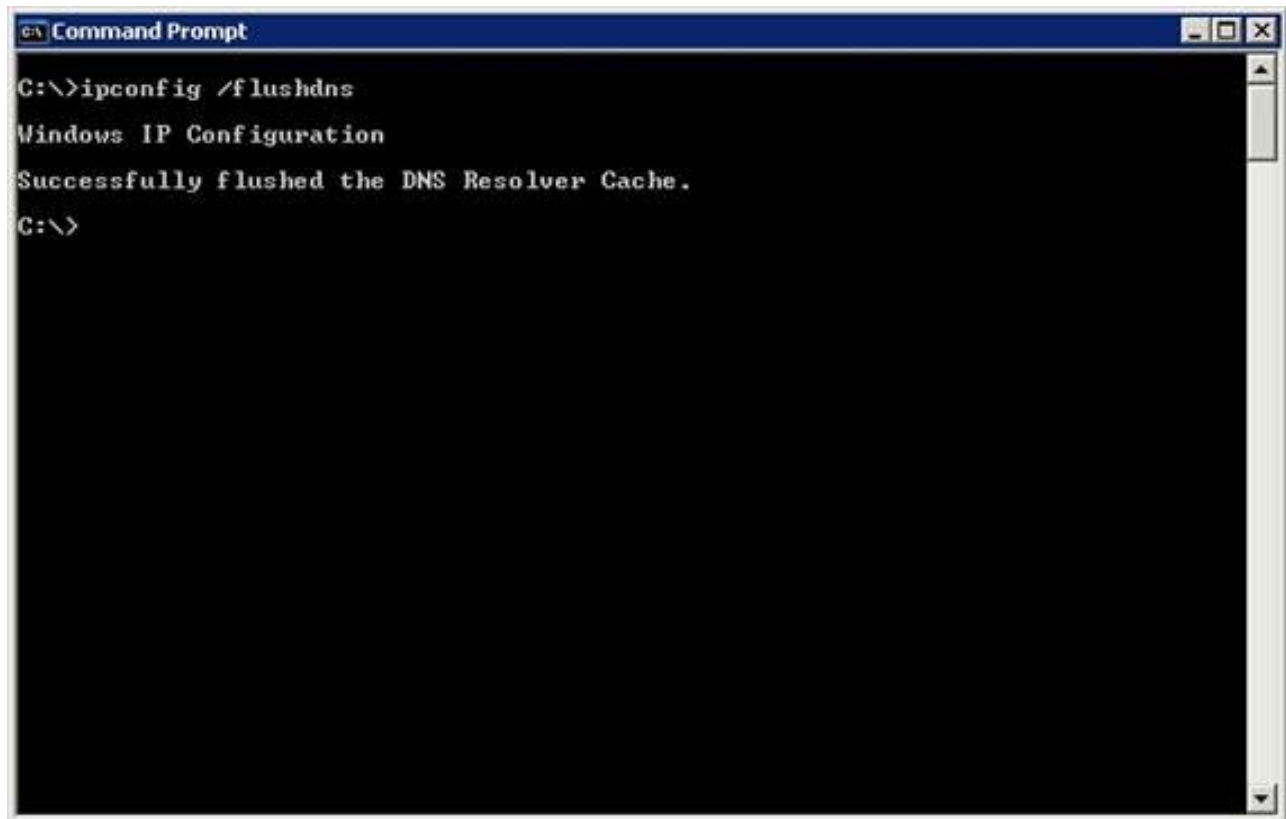
Troubleshooting Windows Network Connectivity

Brien M. Posey

If you should encounter such a problem, then I would recommend scanning the client machine for malware. It is important to scan for both spyware and viruses since both are known to cause this type of problem. Once the machine is free of malware, then try flushing the DNS cache. You can flush the DNS cache by entering the following command:

```
IPCONFIG /FLUSHDNS
```

You can see an example of this, shown in Figure E:



```
Command Prompt
C:\>ipconfig /flushdns
Windows IP Configuration
Successfully flushed the DNS Resolver Cache.
C:\>
```

Figure E
Flushing suspect DNS cache that may contain inaccurate information

It is important to keep in mind that just because the DNS cache contains inaccurate IP addresses, it does not always mean that DNS poisoning has taken place. Sometimes hosts are assigned new IP addresses, and it takes the DNS cache a while to become aware of the changes.

Conclusion

In this article, I have explained how you can verify that the local TCP/IP protocol stack is working correctly. I then went on to explain how to test the local host's ability to contact the DNS server and the default gateway server, and how to test host name resolution. In the next part of this series, I will discuss a few more common problems that you can detect using the ping command, and will begin discussing routing issues.

Troubleshooting Windows Network Connectivity

Brien M. Posey

Using Tracert And TTL To Troubleshoot Network Connectivity Problems

So far in this article series, I have shown you all kinds of tricks that involve using the ping command to diagnose network connectivity problems. In this article, I want to continue the discussion by showing you some variations of these techniques.

Packet Loss

So far when we have used the ping command, the command has either been successful, or it has failed. There really has not been any in-between. As you may recall, the ping command is designed to return four different responses. Occasionally, one or more of these responses may fail while others succeed. When this happens, it means that packet loss is occurring.

In such a situation, the local host and the remote host or both are functioning properly, but conditions exist that cause some packets to be lost along the way. The TCP/IP protocol is designed so that it can retry the transmission when packet loss occurs, but packet loss kills performance. A slow connection with no packet loss will often outperform a high-speed connection on which packet loss is occurring.

The tricky thing about packet loss is that it can sometimes be hard to spot. Sure, you know that packet loss is occurring if some of the ping responses fail, but ICMP packets used by pinging are so small that they will often be successfully returned even if a network condition exists that may cause packet loss in real world situations.

If you suspect that packet loss may be occurring but ping is not returning any errors then you can try increasing the size of the ICMP packets. Larger packets are more prone to failure if network problems exist. You can tell ping to use larger packet sizes by using the `-L` switch.

Using the `-L` switch is simple. All you have to do is enter the ping command followed by the address that you want to ping, and the `-L` switch and the number of bytes that you want to send. For example, suppose that your network was experiencing poor performance when connecting to a particular host. You suspect that packet loss is occurring, but ping is consistently successful. Therefore, you decide to tell ping to use a packet size of 1024 bytes. To do so, you would use the following command:

```
Ping 192.168.1.1 -L 1024
```

You can see a real world example of how this command works, in Figure A:

Troubleshooting Windows Network Connectivity

Brien M. Posey



```
C:\Documents and Settings\Administrator\FUBAR>ping 147.100.100.100 -l 1024

Pinging 147.100.100.100 with 1024 bytes of data:

Reply from 147.100.100.100: bytes=1024 time<1ms TTL=64
Reply from 147.100.100.100: bytes=1024 time<1ms TTL=64
Reply from 147.100.100.100: bytes=1024 time<1ms TTL=64
Reply from 147.100.100.100: bytes=1024 time<1ms TTL=64

Ping statistics for 147.100.100.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\Administrator\FUBAR>
```

Figure A

Appending the `-L` command to the ping command to increase the size of the ICMP packet Time-to-live

The next concept that I want to discuss in relation to the ping command is that of time-to-live (TTL). If you take a look at Figure A, you will notice that each of the ping replies ends in TTL=64.

As you probably know, the Internet consists of a huge number of routers that are connected to each other. Every router is connected to at least two other routers. The idea behind this architecture is that if a link fails, there should be at least one other path to the destination. The problem with this type of architecture is that under certain circumstances link failures could cause packets to travel in circles for infinity, never actually reaching their destination.

This is where the TTL value comes into play. Think of the TTL value as a self-destruct mechanism for the packet. The TTL value is initially set at a fairly high number, although this number varies depending on the operating system that is being used. Every time the packet travels across to a router, the packet is said to have performed a hop. Each time that a hop occurs, the TTL value is decremented by one. If the TTL value reaches zero, the packet is destroyed. This keeps a lost packet from traveling around the Internet for all eternity.

Traceroute

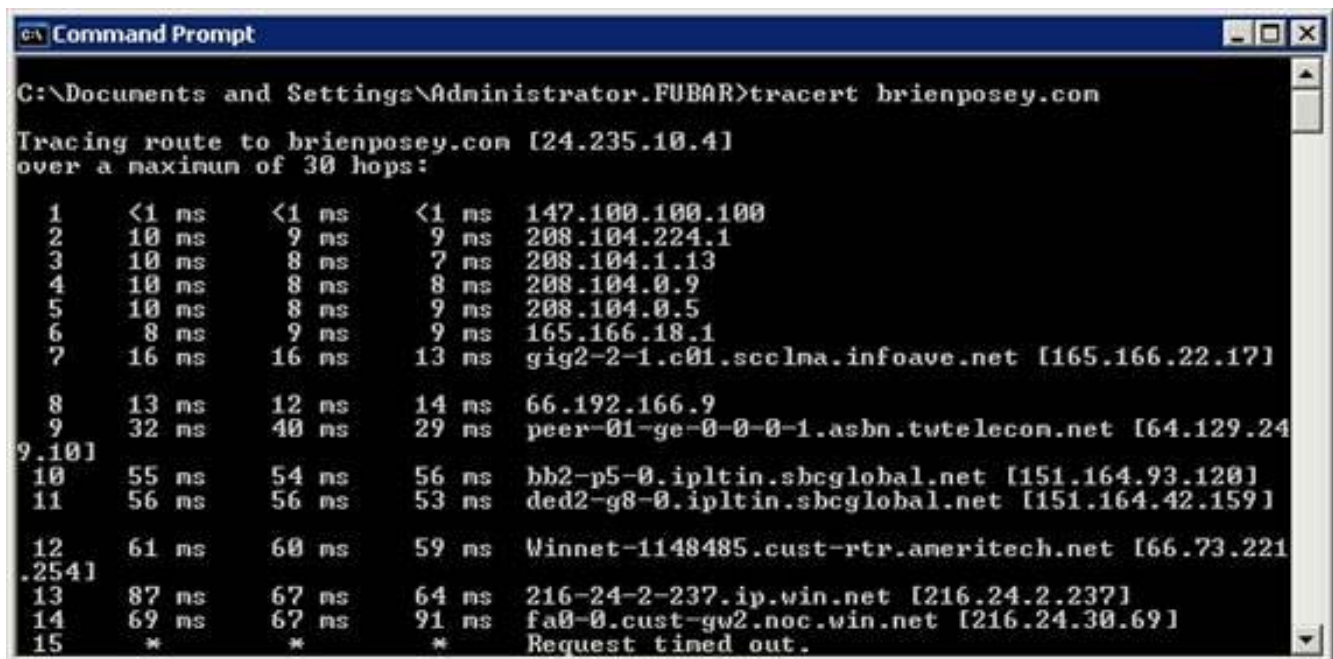
Another reason why the TTL value is so useful is because a troubleshooting tool called traceroute (tracert) is based on it. Using the ping command is fine for troubleshooting small networks in which the remote host is in close proximity to the sending host, but when it comes to the Internet or to a wide area network (WAN) the remote host may be thousands of miles away. As such, the ICMP packet generated by the ping command may have to travel through dozens of routers in order to reach the remote host. You may occasionally run into situations in which the local host and the remote host or both are working correctly, but one of the routers somewhere along the way is having problems. Fortunately, you can use the tracert command to diagnose these types of problems.

Troubleshooting Windows Network Connectivity

Brien M. Posey

The tracert command is actually based on the ping command. The basic idea behind tracert is that it sends out an ICMP packet to the remote host, but with the TTL value set to one. This causes the first router encountered to send back a TTL expired in transit message. This message contains information that identifies the router that produced the message. The router's identification is documented, and then the ICMP packet is sent out again, but this time with a TTL value of two. This time, the ICMP packet reaches the second router before the TTL value expires. This process is repeated, increasing the TTL value by one each time, until the host is eventually reached. This allows you to see a report of all of the routers between the local host and the remote host. You can sometimes use this information to spot problems along the route that may be affecting traffic flow.

Using the tracert command is very similar to using the ping command. To do so, simply enter the tracert command followed by the IP address or the fully qualified domain name of the remote host. Figure B shows the tracert command in action:



```
C:\Documents and Settings\Administrator\FUBAR>tracert brienposey.com

Tracing route to brienposey.com [24.235.10.4]
over a maximum of 30 hops:

  0  <1 ms    <1 ms    <1 ms    147.100.100.100
  1  10 ms     9 ms     9 ms     208.104.224.1
  2  10 ms     8 ms     7 ms     208.104.1.13
  3  10 ms     8 ms     8 ms     208.104.0.9
  4  10 ms     8 ms     9 ms     208.104.0.5
  5   8 ms     9 ms     9 ms     165.166.18.1
  6  16 ms    16 ms    13 ms    gig2-2-1.c01.scclma.infoave.net [165.166.22.17]
  7
  8  13 ms    12 ms    14 ms    66.192.166.9
  9  32 ms    40 ms    29 ms    peer-01-ge-0-0-1.asbn.twtelcom.net [64.129.24
9.10]
 10  55 ms    54 ms    56 ms    bb2-p5-0.ipltin.sbcglobal.net [151.164.93.120]
 11  56 ms    56 ms    53 ms    ded2-g8-0.ipltin.sbcglobal.net [151.164.42.159]
 12
 13  61 ms    60 ms    59 ms    Winnet-1148485.cust-rtr.ameritech.net [66.73.221
.254]
 14  87 ms    67 ms    64 ms    216-24-2-237.ip.win.net [216.24.2.237]
 15  69 ms    67 ms    91 ms    fa0-0.cust-gw2.noc.win.net [216.24.30.69]
 16  *        *        *        Request timed out.
```

Figure B
Tracert command being used to spot problems with traffic flow

There are a couple of different things to keep in mind when using the tracert command. First, some hosts use a firewall to block ICMP packets. As such, you will sometimes see a series of asterisks indicating that trace route was not able to get information from a particular host.

Another thing to keep in mind is that, like the hosts themselves, every router is assigned an IP address. Regardless of whether they are used for hosts or for routers, IP addresses are structured in a way that allows them to reflect their geographic location. In fact, sometimes this geographic information or even a description of the router is provided within the tracert. If you want more information though, there are third-party tools that can graphically track the tracert command based on this geographic information. You can see an example of such a tool in Figure C:

Troubleshooting Windows Network Connectivity

Brien M. Posey



Figure C
Performing a visual traceroute to determine a host's geographic location

Conclusion

In this article, I have shown you how to increase the number of bytes used by the ping command in an effort to make it easier to spot packet loss. I then went on to introduce you to the traceroute command. In the next part of this article series, I will continue the discussion by showing you how to interpret the results provided by traceroute.

Troubleshooting Windows Network Connectivity

Brien M. Posey

Understand Windows Tracert Output To Troubleshoot Network Connectivity

In the previous part of this article series, Using tracert and TTL to troubleshoot network connectivity problems, I explained that tracert could be used to help diagnose connectivity problems between local hosts, and hosts on remote networks. In that article, I showed you how to issue a basic tracert command. So in this article, I will continue the discussion by showing you how you can interpret the results.

For demonstration purposes, I have performed a tracert against www.espn.com. The only reason I chose this particular site is that it is one of the few sites that I know of off the top of my head that does not block Internet Control Message Protocol (ICMP) traffic.

You can see the output from the traceroute below. I will be referring to this output throughout the rest of the article:

```
C:\Users\Administrator>TRACERT www.espn.com
```

```
Tracing route to www.espn.com [199.181.132.250] over a maximum of 30 hops:
```

```
  1     2 ms     1 ms     <1 ms   147.100.100.100
  2    10 ms    10 ms     9 ms   208.104.224.1
  3     9 ms     9 ms     9 ms   208.104.1.13
  4     9 ms     8 ms     9 ms   208.104.0.13
  5    10 ms     9 ms    10 ms   208.104.0.1
  6    11 ms    14 ms    10 ms   165.166.125.193
  7    11 ms    10 ms    11 ms   gig-1-1-3.core01.ncchr1.infoave.net
[165.166.22.61]
  8    31 ms    31 ms    30 ms   64.200.130.17
  9    38 ms    39 ms    40 ms   hrndvalwcx2-pos15-3-oc48.wcg.net [64.200.240.213]
 10    31 ms    31 ms    31 ms   64.200.249.170
 11    31 ms    30 ms    31 ms   4.68.110.5
 12    48 ms    35 ms    35 ms   vlan99.csw4.Washington1.Level3.net [4.68.17.254]
 13    32 ms    31 ms    33 ms   ae-92-92.ebr2.Washington1.Level3.net
[4.69.134.157]
 14    60 ms    53 ms    54 ms   ae-2.ebr3.Chicago1.Level3.net [4.69.132.69]
 15    86 ms    71 ms    70 ms   ae-3.ebr2.Denver1.Level3.net [4.69.132.61]
 16   137 ms   103 ms   102 ms   ae-2.ebr2.Seattle1.Level3.net [4.69.132.53]
 17    95 ms    95 ms    95 ms   ae-23-52.car3.Seattle1.Level3.net [4.68.105.36]
 18    94 ms    95 ms    95 ms   WALT-DISNEY.car3.Seattle1.Level3.net [4.71.152.22]
 19     *      *      *      Request timed out.
 20    97 ms    95 ms    98 ms   199.181.132.250
```

```
Trace complete.
```

If you look at the tracert above, you will notice that each line of the output contains several different pieces of information. The first piece of information, found on the leftmost side of each line, is the hop number. As I explained in the previous article, tracert works by sending a ping request to the specified host. Initially, the request's time-to-live (TTL) value is set to 1. This ensures that the request will fail after the first hop. Information about the hop is presented, and then the ICMP request is transmitted again, but this time with the TTL value set to 2. The process is repeated over and over again, increasing the TTL value by 1 each time, until the specified host is finally reached. In doing so, tracert is able to report how many hops the request had to make in order to reach the remote host. If you look at the last line of the output above, you will see that it begins with the number 20. That is because it took 20 hops to reach the specified host.

Troubleshooting Windows Network Connectivity

Brien M. Posey

The next three pieces of information on each line display the length of time that it took to reach the router or host that the particular line refers to. If you look through the list, you will notice that the time links generally increase with each hop. There are two things that you really need to know about the time links that are displayed.

First, three separate time lengths are displayed for each hop. As I mentioned before, traceroute is based on the concept of sending multiple ICMP requests. When we worked with the ping command earlier in this article series, you saw that the ping command always returned four different values as a way of measuring packet loss. The same concept applies to traceroute, except that the length of time the request took is measured three times instead of four.

The second thing that you need to know about the response times is that an asterisk indicates that a request has timed out. This may or may not indicate a problem, depending on how the asterisk appears. If you look at hop number 19 in the output above, you will notice that all three response time values are presented as asterisks. When you see three asterisks in a row, it usually means that the device that is being pinged on at hop has its firewall configured to reject ICMP packets. This will cause each of the timers to time out, and the final column will simply display the words "Request Timed Out."

Keep in mind, however, that although this is usually the case, it is not the only possibility. Traceroute will also display three asterisks when the device in question is unreachable. Of course, that raises the question of how you can tell the difference between a site that blocks ICMP packets and a link failure. Well, it can be a little tricky.

At first glance, a link failure looks identical to what you see when a router or a host blocks ICMP requests. When a failure occurs, you are not going to see an error message. In fact, the process ends with the standard "Trace Complete" message.

There are two good signs that a link failure has occurred. One sign is that beyond a certain point in the trace, every result that is returned times out. Another sign of a link failure is that the tracert proceeds for a full 30 hops. Neither of these conditions guarantees that a link failure has occurred, even when they occur together. For example, my website, www.brienposey.com, is working fine at the moment, and yet when I run a tracert against it, both of these symptoms show up, as shown in the output below:

```
C:\Users\Administrator>TRACERT www.brienposey.com
```

```
Tracing route to www.brienposey.com [24.235.10.4] over a maximum of 30 hops:
  0  1 ms    1 ms    <1 ms   147.100.100.100
  1  8 ms    12 ms   8 ms    208.104.224.1
  2  9 ms    8 ms    9 ms    208.104.1.9
  3  10 ms   9 ms    8 ms    208.104.0.9
  4  10 ms   12 ms   11 ms   208.104.0.5
  5  12 ms   10 ms   9 ms    165.166.18.1
  6  15 ms   23 ms   13 ms   gig2-2-1.c01.scclma.infoave.net [165.166.22.17]
  7  13 ms   12 ms   13 ms   66.192.166.9
  8  31 ms   30 ms   *       peer-01-ge-0-0-0-1.asbn.twtelecom.net
[64.129.249.10]
  9  56 ms   57 ms   55 ms   bb2-p6-0.ipltin.sbcglobal.net [151.164.242.59]
 10  55 ms   53 ms   55 ms   ded2-g8-0.ipltin.sbcglobal.net [151.164.42.159]
```

Troubleshooting Windows Network Connectivity

Brien M. Posey

```
12      59 ms      56 ms      56 ms  Winnet-1148485.cust-rtr.ameritech.net
[66.73.221.254]
13      64 ms      63 ms      68 ms  216-24-2-237.ip.win.net [216.24.2.237]
14      68 ms      68 ms      64 ms  fa0-0.cust-gw2.noc.win.net [216.24.30.69]
15      *          *          *      Request timed out.
16      *          *          *      Request timed out.
17      *          *          *      Request timed out.
18      *          *          *      Request timed out.
19      *          *          *      Request timed out.
20      *          *          *      Request timed out.
21      *          *          *      Request timed out.
22      *          *          *      Request timed out.
23      *          *          *      Request timed out.
24      *          *          *      Request timed out.
25      *          *          *      Request timed out.
26      *          *          *      Request timed out.
27      *          *          *      Request timed out.
28      *          *          *      Request timed out.
29      *          *          *      Request timed out.
30      *          *          *      Request timed out.
```

Trace complete.

If you see an output like the one above, it may indicate that a link failure has occurred, but it does not guarantee it. The only way to know for sure is to try running a tracert against multiple sites to see whether you keep getting the same type of results. Keep in mind that higher-numbered hops are further away from you. The further away a failure is, the harder it will be to diagnose, because tests of other sites may take alternate routes. When you perform tracert tests against multiple sites, you will have to look at the routes that were actually taken to determine whether or not a link failure is occurring.

The final piece of information displayed on each row is the identity of the router or host that responded to the ICMP request. Tracert will identify each host or router by name whenever possible, but you will not always get a full name resolution. For example, if you look at the output above, you can see that about half of the routers are identified by name, while the others are not. In and of itself, that is not usually a big deal.

What you might find interesting is that the host that you are tracing the route to is not always going to be identified. For example, if you look at the very beginning of the first sample output above, you will notice that we entered the command TRACERT WWW.ESPN.COM. Immediately after doing so, tracert resolved www.espn.com to the IP address 199.181.132.250. If you skip ahead to the end of the sample output, you will notice that tracert eventually reaches its destination, but it does not identify the destination by name (at least, not in this case).

This behavior is not problematic, it is by design. The reason I showed you this is so that you would not try to perform a tracert to a site and think that the process failed because the destination host is not identified by name.

Conclusion

In this article, I have shown you how to decipher the output of a tracert. In the next article in this series, I will show you how to use the Route command to examine a machine's routing tables.