Using The traceroute Command

Using traceroute

Where do all those packets really go when we send them over the Internet? And, how do all the packets actually get to their destinations? Well, we can use the **TCP/IP traceroute** (*tracert with Windows*) command-line utility to help us answer both questions because its output will show us every router interface a TCP/IP packet passes through on the way to its destination.

**Traceroute** (trace for short) displays the path a packet takes to get to a remote device by using something we call **IP packet Time to Live (TTL) time-outs** and **Internet Control Message Protocol (ICMP)** error messages. And it’s also a handy tool for troubleshooting an internetwork because we can use it to figure out which router along a path through that internetwork happens to be causing a network failure when a certain destination machine or network is, or suddenly becomes, unreachable.

To use **tracert**, at a Windows command prompt, type tracert, a space, and the Domain Name Service (DNS) name or IP address of the host machine to which you want to find the route. The **tracert utility** will respond with a list of all the DNS names and IP addresses of the routers that the packet is passing through on its way. Plus, tracert uses TTL to indicate the time it takes for each attempt.

Following is the *tracert output* from a local pc to clarusway.com server:

Microsoft Windows [Version 10.0.18363.657]

(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\clarusway>tracert www.clarusway.com

Tracing route to www.clarusway.com [54.164.151.235]

over a maximum of 30 hops:

1 1 ms 1 ms 1 ms 192.168.1.1

2 4 ms 4 ms \* 195.87.128.37

3 10 ms 10 ms \* 10.135.53.154

4 11 ms 10 ms \* 10.135.53.153

5 11 ms 11 ms \* 46.234.28.57

6 11 ms 10 ms \* ae4-17-ucr1.tuz.cw.net [195.2.23.129]

7 133 ms 134 ms 132 ms 195.2.25.86

8 130 ms 130 ms 130 ms 195.2.28.57

9 132 ms 133 ms 132 ms ae17.pcr1.fnt.cw.net [195.2.20.226]

10 \* \* 131 ms ae15-pcr1.ptl.cw.net [195.2.9.126]

11 131 ms 131 ms 133 ms et-7-1-0-xcr1.nyh.cw.net [195.2.24.241]

12 131 ms 131 ms 155 ms ae13-xcr2.nyk.cw.net [195.2.25.69]

13 132 ms 132 ms 132 ms 52.95.216.78

14 141 ms 135 ms 131 ms 52.93.4.85

15 131 ms 131 ms 132 ms 52.93.4.46

16 \* \* \* Request timed out.

17 140 ms 136 ms 137 ms 150.222.242.116

18 \* \* \* Request timed out.

19 \* \* \* Request timed out.

20 \* \* \* Request timed out.

21 \* \* \* Request timed out.

22 137 ms 158 ms 140 ms 150.222.241.173

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.

You see that the packet bounces through several routers before arriving at its destination. This utility is useful if you are having problems reaching a web server on the Internet and you want to know if a wide area network (WAN) link is down or if the server just isn’t responding. What this means to you is that, basically, wherever the trace stops is a great place to start troubleshooting. Notice in the output the *ms*. This is the latency of each hop, meaning the delay. Tracert (or traceroute) is a great troubleshooting tool you can use to find out where your network bottlenecks are.

If you use traceroute or tracert and receive an asterisk, this indicates that the attempt to reach that router took longer than the default time-out value. This is very good to know because it can mean that either the router is extremely busy or a particular link is slow. Another reason for getting an asterisk could be that the administrator has disabled ICMP on the router that the packet is trying to hop through because of security reasons. It happens to be a typical strategic move done on the routers that interface to the ISP to conceal their actual location so bad guys can’t hack into them and therefore into your internetwork.

In addition to traceroute and tracert, you can use pathping (for Windows), which is a lot like traceroute:

C:\Users\clarusway>pathping www.clarusway.com

Tracing route to www.clarusway.com [54.164.151.235]

over a maximum of 30 hops:

0 freestyler.home [192.168.1.22]

1 192.168.1.1

2 195.87.128.37

3 10.135.53.154

4 10.135.53.153

5 46.234.28.57

6 ae4-17-ucr1.tuz.cw.net [195.2.23.129]

7 ae2-ucr1.pra.cw.net [195.2.25.86]

8 ae16-xcr1.fix.cw.net [195.2.28.57]

9 ae17.pcr1.fnt.cw.net [195.2.20.226]

10 ae15-pcr1.ptl.cw.net [195.2.9.126]

11 et-7-1-0-xcr1.nyh.cw.net [195.2.24.241]

12 ae13-xcr2.nyk.cw.net [195.2.25.69]

13 52.95.216.78

14 52.93.4.85

15 52.93.4.46

16 \* \* \*

Computing statistics for 375 seconds...

Source to Here This Node/Link

Hop RTT Lost/Sent = Pct Lost/Sent = Pct Address

0 freestyler.home [192.168.1.22]

0/ 100 = 0% |

1 1ms 0/ 100 = 0% 0/ 100 = 0% 192.168.1.1

0/ 100 = 0% |

2 --- 100/ 100 =100% 100/ 100 =100% 195.87.128.37

0/ 100 = 0% |

3 13ms 0/ 100 = 0% 0/ 100 = 0% 10.135.53.154

0/ 100 = 0% |

4 12ms 0/ 100 = 0% 0/ 100 = 0% 10.135.53.153

0/ 100 = 0% |

5 12ms 0/ 100 = 0% 0/ 100 = 0% 46.234.28.57

0/ 100 = 0% |

6 15ms 0/ 100 = 0% 0/ 100 = 0% ae4-17-ucr1.tuz.cw.net [195.2.23.129]

0/ 100 = 0% |

7 47ms 0/ 100 = 0% 0/ 100 = 0% ae2-ucr1.pra.cw.net [195.2.25.86]

0/ 100 = 0% |

8 56ms 0/ 100 = 0% 0/ 100 = 0% ae16-xcr1.fix.cw.net [195.2.28.57]

0/ 100 = 0% |

9 59ms 0/ 100 = 0% 0/ 100 = 0% ae17.pcr1.fnt.cw.net [195.2.20.226]

0/ 100 = 0% |

10 64ms 0/ 100 = 0% 0/ 100 = 0% ae15-pcr1.ptl.cw.net [195.2.9.126]

0/ 100 = 0% |

11 134ms 0/ 100 = 0% 0/ 100 = 0% et-7-1-0-xcr1.nyh.cw.net [195.2.24.241]

0/ 100 = 0% |

12 133ms 0/ 100 = 0% 0/ 100 = 0% ae13-xcr2.nyk.cw.net [195.2.25.69]

100/ 100 =100% |

13 --- 100/ 100 =100% 0/ 100 = 0% 52.95.216.78

0/ 100 = 0% |

14 --- 100/ 100 =100% 0/ 100 = 0% 52.93.4.85

0/ 100 = 0% |

15 --- 100/ 100 =100% 0/ 100 = 0% 52.93.4.46

Trace complete.

Using The traceroute Command

Using the ipconfig Utility

With the new Mac, Windows 10, and Windows Server 2016 operating systems, you can now see the IPv6 configuration because IPv6 is enabled by default. The output of the ipconfig command provides the basic routed protocol information on your machine. From a DOS prompt, type ipconfig, and you’ll see something like this:

C:\Users\clarusway>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

Media State . . . . . . . . . . . : Media disconnected

Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection\* 3:

Media State . . . . . . . . . . . : Media disconnected

Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : home

Link-local IPv6 Address . . . . . : fe80::19ac:8efb:2c6e:f512%10

IPv4 Address. . . . . . . . . . . : 192.168.1.22

Subnet Mask . . . . . . . . . . . : 255.255.255.0

Default Gateway . . . . . . . . . : 192.168.1.1

Tunnel adapter Teredo Tunneling Pseudo-Interface:

Connection-specific DNS Suffix . :

IPv6 Address. . . . . . . . . . . : 2001:0:2851:782c:148e:f3fd:6aff:55b8

Link-local IPv6 Address . . . . . : fe80::148e:f3fd:6aff:55b8%17

Default Gateway . . . . . . . . . : ::

You can see that Ethernet adapter shows up first, and it has an IP address, a mask, and a default gateway plus an IPv6 address and a DNS suffix. The next configured interface is the wireless local area network (LAN) adapter, which has an IP address, a mask, a default gateway, an IPv6 address, and the IPv6 default gateway as well.

The next adapters are disconnected because they are logical interfaces and are not being used. But just in case the ipconfig command doesn’t provide enough information for you, try the ipconfig /all command. Here’s the beginning of that output:

C:\Users\clarusway>ipconfig /all

Windows IP Configuration

Host Name . . . . . . . . . . . . : clarusway

Primary Dns Suffix . . . . . . . :

Node Type . . . . . . . . . . . . : Hybrid

IP Routing Enabled. . . . . . . . : No

WINS Proxy Enabled. . . . . . . . : No

DNS Suffix Search List. . . . . . : home

Ethernet adapter Ethernet:

Media State . . . . . . . . . . . : Media disconnected

Connection-specific DNS Suffix . :

Description . . . . . . . . . . . : Intel(R) I211 Gigabit Network Connection

Physical Address. . . . . . . . . : 9C-5C-8E-CE-D9-C9

DHCP Enabled. . . . . . . . . . . : Yes

Autoconfiguration Enabled . . . . : Yes

Ethernet adapter Ethernet 3:

Media State . . . . . . . . . . . : Media disconnected

Connection-specific DNS Suffix . :

Description . . . . . . . . . . . : Intel(R) Ethernet Connection (2) I219-V

Physical Address. . . . . . . . . : 9C-5C-8E-CE-D9-CA

DHCP Enabled. . . . . . . . . . . : Yes

Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Local Area Connection\* 3:

Media State . . . . . . . . . . . : Media disconnected

Connection-specific DNS Suffix . :

Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter

Physical Address. . . . . . . . . : 76-C6-3B-00-62-86

DHCP Enabled. . . . . . . . . . . : Yes

Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Local Area Connection\* 13:

Media State . . . . . . . . . . . : Media disconnected

Connection-specific DNS Suffix . :

Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2

Physical Address. . . . . . . . . : 76-C6-3B-00-6A-86

DHCP Enabled. . . . . . . . . . . : Yes

Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : home

Description . . . . . . . . . . . : Broadcom 802.11ac Network Adapter

Physical Address. . . . . . . . . : 74-C6-3B-00-62-86

DHCP Enabled. . . . . . . . . . . : Yes

Autoconfiguration Enabled . . . . : Yes

Link-local IPv6 Address . . . . . : fe80::19ac:8efb:2c6e:f512%10(Preferred)

IPv4 Address. . . . . . . . . . . : 192.168.1.22(Preferred)

Subnet Mask . . . . . . . . . . . : 255.255.255.0

Lease Obtained. . . . . . . . . . : March 3, 2020 01:01:19 PM

Lease Expires . . . . . . . . . . : March 4, 2020 02:54:14 PM

Default Gateway . . . . . . . . . : 192.168.1.1

DHCP Server . . . . . . . . . . . : 192.168.1.1

DHCPv6 IAID . . . . . . . . . . . : 242533947

DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-21-FA-0A-0E-9C-5C-8E-CE-D9-C9

DNS Servers . . . . . . . . . . . : 192.168.1.1

192.168.1.1

NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter Teredo Tunneling Pseudo-Interface:

Connection-specific DNS Suffix . :

Description . . . . . . . . . . . : Microsoft Teredo Tunneling Adapter

Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0

DHCP Enabled. . . . . . . . . . . : No

Autoconfiguration Enabled . . . . : Yes

IPv6 Address. . . . . . . . . . . : 2001:0:2851:782c:148e:f3fd:6aff:55b8(Preferred)

Link-local IPv6 Address . . . . . : fe80::148e:f3fd:6aff:55b8%17(Preferred)

Default Gateway . . . . . . . . . : ::

DHCPv6 IAID . . . . . . . . . . . : 167772160

DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-21-FA-0A-0E-9C-5C-8E-CE-D9-C9

NetBIOS over Tcpip. . . . . . . . : Disabled

As you can see, it’s more of the same—a whole lot more. The most important thing that you can see the hardware information about each interface, including the Media Access Control (MAC) address. Also significant is that you can see the Dynamic Host Configuration Protocol (DHCP) lease times and DNS addresses now.

There are two more valuable options you need to use with the ipconfig command. They are /release and /renew.

When you change networks, you need to get the IP address of that subnet and/or virtual LAN (VLAN). Windows 10 works most of the time without doing anything, but sometimes you have to renew the IP configuration when changing networks. For that, just type ipconfig /renew from a command prompt, and if you’re connected to a DHCP server that’s available. Now, if it still doesn’t work, you’ll need to release and renew your TCP/IP settings. To release your current DHCP TCP/IP information, you must elevate your command prompt or you’ll get this warning:

C:\Users\clarusway>ipconfig /release

The requested operation requires elevation.

In order to avoid this, choose Start > All Programs > Accessories > Command Prompt, right-click, and choose Run As Administrator.

Once your command prompt has been duly elevated, you can use the ipconfig /release command and then the ipconfig /renew command to get new TCP/IP information for your host.

Using The traceroute Command

Using the ifconfig Utility

There is a utility in Linux/Unix/Mac that will give you information similar to what ipconfig shows. It’s called ifconfig (short for interface configuration). Although ipconfig and ifconfig show similar information, there are major differences between these two utilities.

The ipconfig utility is mainly used to view the TCP/IP configuration for a computer. You can use ifconfig to do the same thing, but ifconfig can also be used to configure a protocol or a particular network interface.

The general syntax of the ifconfig command is as follows:

ifconfig interface [address [parameters]]

The interface parameter equals the Unix name of the interface, such as eth0. If the optional address parameter is specified, the ifconfig command sets the IP address for the interface to the address you’ve specified. When the ifconfig command is used by itself with no parameters, all configured interfaces will be reported on. But if only the interface name is specified, you’ll get output that looks like this:

# ifconfig eth0

eth0 Link encap 10Mbps Ethernet HWaddr 00:00:C0:90:B3:42

inetaddr 172.16.0.2 Bcast 172.16.0.255 Mask 255.255.255.0 UP

BROADCAST RUNNING MTU 1500 Metric 0

RX packets 3136 errors 217 dropped 7 overrun 26

TX packets 1752 errors 25 dropped 0 overrun 0

Looking at this, we can see that the eth0 interface is a 10 Mbps Ethernet interface. The interface’s MAC and IP address information is displayed in this output as well. And, although not shown in the output, the ifconfig tool can show you the DNS information configured on the host.

Using The traceroute Command

Using the iptables Utility

The **iptables firewall utility** is built for the Linux operating system. It is a command-line utility that uses what are called chains to allow or disallow traffic. When traffic arrives, iptables looks for a rule that addresses that traffic type, and if none exists, it will enforce the default rule. There are three different chain types:

1. **Input:** Controls behavior for incoming connections
2. **Forward:** Used for incoming connections that aren’t being delivered locally
3. **Output:** Used for outgoing connections

You can set the default action to accept, drop, or reject, with the difference between reject and drop being that reject sends an error message back to the source.

Examples of iptables

* To block a connection from the device at 192.168.10.1, use this command:

iptables -A INPUT -s 192.168.10.1 -j DROP

* To block all connections from all devices in the 172.16.0.0/16 network, use this command:

iptables -A INPUT -s 172.16.0.0/16 -j DROP

* Here is the command to block SSH connections from 10.110.61.5:

iptables -A INPUT -p tcp --dport ssh -s 10.110.61.5 -j DROP

* Use this command to block SSH connections from any IP address:

iptables -A INPUT -p tcp --dport ssh -j DROP

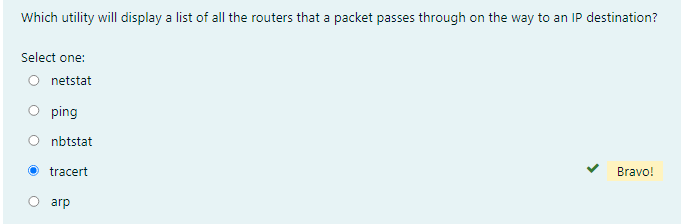
* The following command is used to save the changes in Ubuntu:

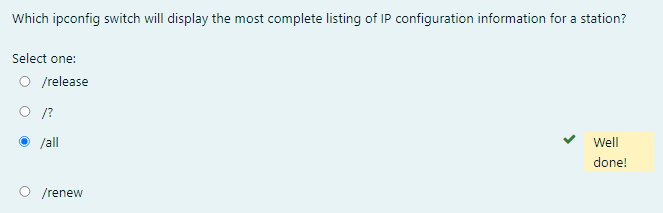
sudo /sbin/iptables-save

* In Red Hat/CentOS, use either of the following commands:

/sbin/service iptables save

/etc/init.d/iptables save





Using the ping Utility

The **ping utility** is the most basic TCP/IP utility, and it’s included with most TCP/IP stacks for most platforms. In most cases, ping is a command-line utility, although there are many GUI implementations available. You use the ping utility for two primary purposes:

* To find out if a host is responding
* To find out if you can reach a host

Here’s the syntax (you can use either command):

ping hostname

ping IP address

If you ping any station that has an IP address, the ICMP that’s part of that particular host’s TCP/IP stack will respond to the request. The ICMP test and response looks something like this:

C:\Users\clarusway>ping 3.225.75.90

Pinging 3.225.75.90 with 32 bytes of data:

Reply from 3.225.75.90: bytes=32 time=137ms TTL=233

Reply from 3.225.75.90: bytes=32 time=136ms TTL=233

Reply from 3.225.75.90: bytes=32 time=134ms TTL=233

Reply from 3.225.75.90: bytes=32 time=134ms TTL=233

Ping statistics for 3.225.75.90:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 134ms, Maximum = 137ms, Average = 135ms

Because we’ve received a reply from the destination station we know that we can reach the host and that it’s responding to basic IP requests. Don’t forget that you can use name resolution and ping to a name, such as ping www.clarusway.com. Most versions of ping work the same way, but there are some switches you can use to specify certain information, like the number of packets to send, how big a packet to send, and so on. And if you’re running the Windows command-line version of ping, just use the /? or –? switch to display a list of the available options like this:

C:\Users\clarusway>ping /?

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]

[-r count] [-s count] [[-j host-list] | [-k host-list]]

[-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]

[-4] [-6] target\_name

Options:

-t Ping the specified host until stopped.

To see statistics and continue - type Control-Break;

To stop - type Control-C.

-a Resolve addresses to hostnames.

-n count Number of echo requests to send.

-l size Send buffer size.

-f Set Don't Fragment flag in packet (IPv4-only).

-i TTL Time To Live.

-v TOS Type Of Service (IPv4-only. This setting has been deprecated

and has no effect on the type of service field in the IP

Header).

-r count Record route for count hops (IPv4-only).

-s count Timestamp for count hops (IPv4-only).

-j host-list Loose source route along host-list (IPv4-only).

-k host-list Strict source route along host-list (IPv4-only).

-w timeout Timeout in milliseconds to wait for each reply.

-R Use routing header to test reverse route also (IPv6-only).

Per RFC 5095 the use of this routing header has been

deprecated. Some systems may drop echo requests if

this header is used.

-S srcaddr Source address to use.

-c compartment Routing compartment identifier.

-p Ping a Hyper-V Network Virtualization provider address.

-4 Force using IPv4.

-6 Force using IPv6.

As you can see, there are many options you can use with the ping command from a Windows DOS prompt.

The -a switch is handy if you have name resolution (such as a DNS server), you can see the name of the destination host even if you only know its IP address. The -n switch sets the number of echo requests to send, where four is the default, and the -w switch allows you to adjust the time-out in milliseconds. The default ping time-out is 1 second (1,000 ms).

The -6 is also nice if you want to ping an IPv6 host. And then there’s -t, which keeps the ping running.

From a MAC, you can use the ping6 command. Here are the options:

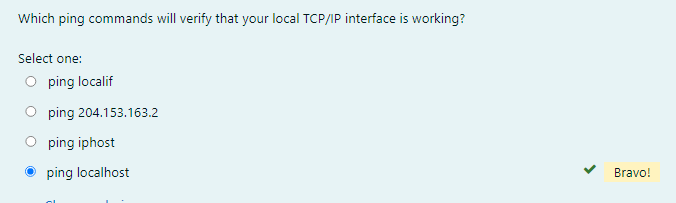
$ ping6

usage: ping6 [-DdfHmnNoqrRtvwW] [-a addrtype] [-b bufsiz] [-B boundif]

[-c count][-g gateway] [-h hoplimit] [-I interface] [-i wait] [-l preload][-p

pattern] [-S sourceaddr] [-s packetsize] [-z tclass]

[hops ...] host



Using the Address Resolution Protocol

The **Address Resolution Protocol (ARP)** is part of the TCP/IP protocol stack. It’s used to translate TCP/IP addresses to MAC addresses using broadcasts. When a machine running TCP/IP wants to know which machine on an Ethernet network is using a certain IP address, it will send an ARP broadcast that says, in effect, “Who is IP address xxx.xxx.xxx.xxx?” The machine that owns the specific address will respond with its own MAC address, supplying the answer. The machine that made the inquiry will respond by adding the newly gained information to its own ARP table.

The ARP table in Windows includes a list of TCP/IP addresses and their associated physical (MAC) addresses. This table is cached in memory so that Windows doesn’t have to perform ARP lookups for frequently accessed TCP/IP addresses like those of servers and default gateways. Each entry contains an IP address and a MAC address plus a value for TTL that determines how long each entry will remain in the ARP table.

Remember that the ARP table contains two kinds of entries:

* Dynamic
* Static

*Dynamic ARP table entries* are created whenever the Windows TCP/IP stack performs an ARP lookup but the MAC address isn’t found in the ARP table. When the MAC address of the requested IP address is finally found or resolved, that information is then added into the ARP table as a dynamic entry. Whenever a request to send a packet to the host is sent to the Data Link layer, the ARP cache is checked first before an ARP broadcast is sent out.

### Using the arp Utility

ARP is used by IP to determine the MAC address of a device that exists on the same subnet as the requesting device. When a TCP/IP device needs to forward a packet to a device on the local subnet, it first looks in its own table, called an ARP cache or MAC address lookup table, for an association between the known IP address of the destination device on the local subnet and that same device’s MAC address. The cache is called that because the contents are periodically weeded out.

If no association that includes the destination IP address can be found, the device will then send out an ARP broadcast that includes its own MAC and IP information as well as the IP address of the target device and a blank MAC address field. Filling in that blank is the object of the whole operation—it’s the unknown value that the source device is requesting to be returned to it in the form of an ARP reply. Windows includes a utility called arp that allows us to check out the operating system’s ARP cache. To view this, from a Windows DOS prompt, use the arp command like this:

C:\Users\clarusway>arp

Displays and modifies the IP-to-Physical address translation tables used by

address resolution protocol (ARP).

ARP -s inet\_addr eth\_addr [if\_addr]

ARP -d inet\_addr [if\_addr]

ARP -a [inet\_addr] [-N if\_addr] [-v]

-a Displays current ARP entries by interrogating the current

protocol data. If inet\_addr is specified, the IP and Physical

addresses for only the specified computer are displayed. If

more than one network interface uses ARP, entries for each ARP

table are displayed.

-g Same as -a.

-v Displays current ARP entries in verbose mode. All invalid

entries and entries on the loop-back interface will be shown.

inet\_addr Specifies an internet address.

-N if\_addr Displays the ARP entries for the network interface specified

by if\_addr.

-d Deletes the host specified by inet\_addr. inet\_addr may be

wildcarded with \* to delete all hosts.

-s Adds the host and associates the Internet address inet\_addr

with the Physical address eth\_addr. The Physical address is

given as 6 hexadecimal bytes separated by hyphens. The entry

is permanent.

eth\_addr Specifies a physical address.

if\_addr If present, this specifies the Internet address of the

interface whose address translation table should be modified.

If not present, the first applicable interface will be used.

Example:

> arp -s 157.55.85.212 00-aa-00-62-c6-09 .... Adds a static entry.

> arp -a

The Windows arp utility is primarily useful for resolving duplicate IP addresses. For example, let’s say your workstation receives its IP address from a DHCP server but it accidentally receives the same address that some other workstation gets. And so, when you try to ping it, you get no response. Your workstation is basically confused—it’s trying to determine the MAC address, and it can’t because two machines are reporting that they have the same IP address. To solve this little snag, you can use the arp utility to view your local ARP table and see which TCP/IP address is resolved to which MAC address.

To display the entire current ARP table, use the arp command with the –a switch like so to show you the mac address lookup table:

C:\Users\clarusway>arp -a

Interface: 192.168.1.22 --- 0xa

Internet Address Physical Address Type

192.168.1.1 24-00-ba-b8-c7-ec dynamic

192.168.1.255 ff-ff-ff-ff-ff-ff static

224.0.0.22 01-00-5e-00-00-16 static

224.0.0.251 01-00-5e-00-00-fb static

224.0.0.252 01-00-5e-00-00-fc static

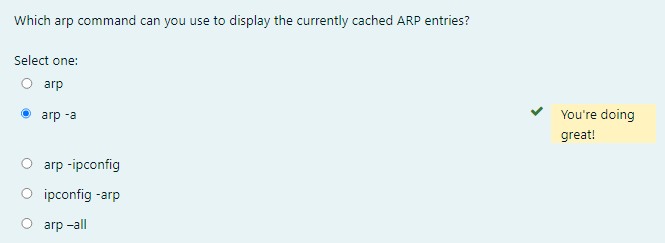
224.0.0.253 01-00-5e-00-00-fd static

239.255.255.250 01-00-5e-7f-ff-fa static

255.255.255.255 ff-ff-ff-ff-ff-ff static

Now, from this output, you can tell which MAC address is assigned to which IP address. Then, for static assignments, you can tell which workstation has a specific IP address and if it’s indeed supposed to have that address by examining your network documentation.

For DHCP-assigned addresses, you can begin to uncover problems stemming from multiple DHCP scopes or servers doling out identical addresses and other common configuration issues. And remember that under normal circumstances, you shouldn’t see IP addresses in the ARP table that isn’t a member of the same IP subnet as the interface.



Using the nslookup Utility

Whenever you’re configuring a server or a workstation to connect to the Internet, you’ve got to start by configuring DNS if you want name resolution to happen. When configuring DNS, it’s a very good thing to be able to test what IP address DNS is returning to ensure that it’s working properly. The nslookup utility allows you to query a name server and quickly find out which name resolves to which IP address.

**💡Tip:**

* The Unix **dig** (short for domain information groper) utility does the exact same thing as **nslookup**. It’s primarily a command-line utility that allows you to perform a single DNS lookup for a specific entity, but it can also be employed in batch mode for a series of lookups.

You can run nslookup from a Windows command prompt. When you’re inside this utility, the command prompt will change from something similar to a C:\> sign to a shorter > sign. It will also display the name and IP address of the default DNS server you will be querying. Then you can start using nslookup. The following output gives you a sample of the display after the nslookup command has been entered at the C:\> prompt.

C:\Users\clarusway> nslookup

Default Server: gnt-corpdc1.globalnet.local

Address: 10.100.36.12

>

The primary job of nslookup is to tell you the many different features of a particular domain name, the names of the servers that serve it, and how they’re configured. To get that, just type in a domain name at the > prompt, and the nslookup utility will then return this information:

> clarusway.com

Server: UnKnown

Address: 192.168.1.1

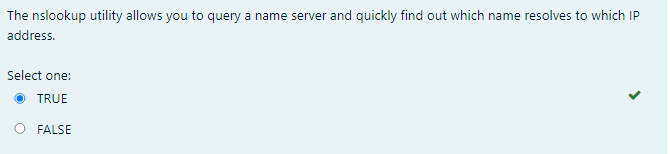
Non-authoritative answer:

Name: clarusway.com

Addresses: 3.225.75.90

54.164.151.235

What this tells you is that the server that returned the information is not responsible (authoritative) for the zone information of the domain for which you requested an address and that the name server for the domain clarusway.com is located at the IP address 3.225.75.90.



### Using the mtr Utility

Mtr or **My traceroute** is a computer program that combines the functions of the traceroute and ping utilities in a single network diagnostic tool. It also adds round-trip time and packet loss to the output. Mtr probes routers on the route path by limiting the number of hops individual packets are allowed to traverse and listening to news of their termination. It will regularly repeat this process (usually once per second) and keep track of the response times of the hops along the path.

Mtr is available for Linux or Unix. Third-party applications of Mtr are available to install on Windows, but Microsoft did respond with its own version of Mtr —it’s called pathping and it provides the same functions as Mtr. Here’s a look at the output and the options:

C:\Users\clarusway>pathping

Usage: pathping [-g host-list] [-h maximum\_hops] [-i address] [-n]

[-p period] [-q num\_queries] [-w timeout]

[-4] [-6] target\_name

Options:

-g host-list Loose source route along host-list.

-h maximum\_hops Maximum number of hops to search for target.

-i address Use the specified source address.

-n Do not resolve addresses to hostnames.

-p period Wait period milliseconds between pings.

-q num\_queries Number of queries per hop.

-w timeout Wait timeout milliseconds for each reply.

-4 Force using IPv4.

-6 Force using IPv6.

Using the Nmap Utility

**Nmap** is one of the most popular port scanning tools used today. After performing scans with certain flags set in the scan packets, security analysts (and hackers) can make certain assumptions based on the responses received. These flags are used to control the TCP connection process and so are present only in TCP packets. The below figure shows a TCP header with the important flags circled. Normally flags are “turned on” because of the normal TCP process, but hackers can craft packets to check the flags they want to check.

|  |
| --- |
| *TCP flags* |

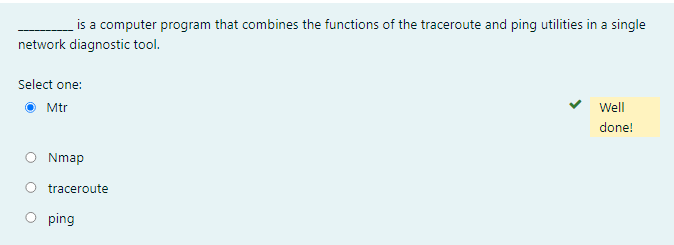
* URG: Urgent pointer field significant
* ACK: Acknowledgment field significant
* PSH: Push function
* RST: Reset the connection
* SYN: Synchronize sequence numbers
* FIN: No more data from sender

Security analysts and hackers alike can perform scans with these flags set in the scan packets to get responses that allow them to determine the following information:

* If a port is open on a device
* If the port is blocked by a firewall before its gets to the device

Nmap can also be used as follows:

* To determine the live hosts on a network
* To create a logical “map” of the network



### Using The route Command

The biggest reason for manipulating the routing table on a server is to create a firewall. For instance, let’s say we’re running an Application layer firewall on a server located between the demilitarized zone (DMZ) and the internal network.

This scenario would mean the routing that’s happening on the server or hosts located in the DMZ wouldn’t be able to reach the internal network’s hosts and vice versa. To circumvent this problem, we would need to employ both static and default routing because running routing protocols on hosts and servers wouldn’t be a good solution for today’s networks.

To view the routing table on a Windows device, use the route print command, as shown below.

C:\Users\clarusway>route print

===========================================================================

Interface List

14...9c 5c 8e ce d9 c9 ......Intel(R) I211 Gigabit Network Connection

18...9c 5c 8e ce d9 ca ......Intel(R) Ethernet Connection (2) I219-V

15...76 c6 3b 00 62 86 ......Microsoft Wi-Fi Direct Virtual Adapter

8...76 c6 3b 00 6a 86 ......Microsoft Wi-Fi Direct Virtual Adapter #2

10...74 c6 3b 00 62 86 ......Broadcom 802.11ac Network Adapter

1...........................Software Loopback Interface 1

17...00 00 00 00 00 00 00 e0 Microsoft Teredo Tunneling Adapter

===========================================================================

IPv4 Route Table

===========================================================================

Active Routes:

Network Destination Netmask Gateway Interface Metric

0.0.0.0 0.0.0.0 192.168.1.1 192.168.1.22 35

127.0.0.0 255.0.0.0 On-link 127.0.0.1 331

127.0.0.1 255.255.255.255 On-link 127.0.0.1 331

127.255.255.255 255.255.255.255 On-link 127.0.0.1 331

192.168.1.0 255.255.255.0 On-link 192.168.1.22 291

192.168.1.22 255.255.255.255 On-link 192.168.1.22 291

192.168.1.255 255.255.255.255 On-link 192.168.1.22 291

224.0.0.0 240.0.0.0 On-link 127.0.0.1 331

224.0.0.0 240.0.0.0 On-link 192.168.1.22 291

255.255.255.255 255.255.255.255 On-link 127.0.0.1 331

255.255.255.255 255.255.255.255 On-link 192.168.1.22 291

===========================================================================

Persistent Routes:

None

IPv6 Route Table

===========================================================================

Active Routes:

If Metric Network Destination Gateway

17 331 ::/0 On-link

1 331 ::1/128 On-link

17 331 2001::/32 On-link

17 331 2001:0:2851:782c:148e:f3fd:6aff:55b8/128

On-link

10 291 fe80::/64 On-link

17 331 fe80::/64 On-link

17 331 fe80::148e:f3fd:6aff:55b8/128

On-link

10 291 fe80::19ac:8efb:2c6e:f512/128

On-link

1 331 ff00::/8 On-link

10 291 ff00::/8 On-link

17 331 ff00::/8 On-link

===========================================================================

Persistent Routes:

None

In this output, you can see that each of the routes was added automatically when the system booted up. To see all the options available with the route command, type route.

route Command Options

To add a route to your routing table, use the following syntax:

route [-f] [-p] [Command] [Destination] [mask Netmask] [Gateway] [metric Metric] [if Interface]

* -f: Using this command with any of the options like add, change, or delete will clear the routing table of all entries that aren’t host routes, the loopback network route or routes, and any multicast routes
* -p: If you use this with the add command, the individual route will be added to the Registry and then used to initialize the IP routing table whenever TCP/IP is started. Important to remember is that by default, the routes you’ve statically added won’t remain in the routing table the next time TCP/IP boots. And if you use -p with the print command, you’ll get shown a list of the persistent routes that are stored in the Registry location of HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes.

Now, let’s take a look at how and when you would use the route command. The below table shows the command options available and what they do when you are using the route command with them.

| **Command** | **Purpose** |
| --- | --- |
| add | Adds a route |
| change | Modifies an existing route |
| delete | Deletes a route (or routes) |
| print | Prints a route (or routes) |

Here’s a description of some other tasks you can accomplish via the rest of the command’s options:

* Destination: This will give you the network destination of a given route. If the host bits of the network address are set to 0, it will be depicted with the destination’s IP network address, an IP address for a specific host route, or the default route of 0.0.0.0.
* mask netmask: This will provide you with the subnet mask that’s associated with the destination network. The default destination subnet mask is 0.0.0.0, and typically you’ll see 255.255.255.255 representing a host route.
* Gateway: The gateway depends on the network address and subnet mask. It defines the next-hop IP address. For routes located on a local subnet, the gateway address maps directly to a particular interface. If the destination is on a remote network, the gateway IP address will direct packets to the router.
* metric: Metric refers to the cost of a given route from the sender to the receiver device, and it has a value between 1 and 9999. Devices use this value to choose the best, or most efficient, routes among those in its routing table—the route with the lowest value wins. This decision can also include factors like the number of hops and the speed, reliability, and available bandwidth of the path being considered.
* if interface: This tool depends on information from the gateway address and determines the interface index for the specific interface that needs to receive the data. You can get a list of interfaces along with their relevant interface indexes by typing the route print command.
* /?: Using this will allow you to view help at the command prompt.

Some Examples of The route Command

It is recommended that you spend some time practicing them on a nonproduction server.

* To display the entire IP routing table, type:

route print

* To add a default route with the default gateway address 192.168.10.1, type:

route add 0.0.0.0 mask 0.0.0.0 192.168.10.1

* To add a route to the destination 10.1.1.0 with the subnet mask 255.255.255.0 and the next-hop address 10.2.2.2, type:

route add 10.1.1.0 mask 255.255.255.0 10.2.2.2

* If you want to add a persistent route to the destination 10.100.0.0 with the subnet mask 255.255.0.0 and the next-hop address 10.2.0.1, type:

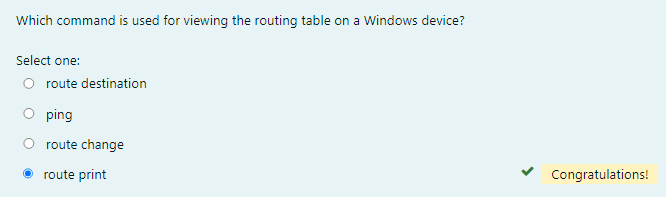
route -p add 10.100.0.0 mask 255.255.0.0 10.2.0.1

* If you want to delete the route to the destination 10.100.0.0 with the subnet mask 255.255.0.0, enter:

route delete 10.100.0.0 mask 255.255.0.0

* If you want to change the next-hop address of a route with the destination 10.100.0.0 and the subnet mask 255.255.0.0 from 10.2.0.1 to 10.7.0.5, type:

route change 10.100.0.0 mask 255.255.0.0 10.7.0.5



Using The nbtstat Utility

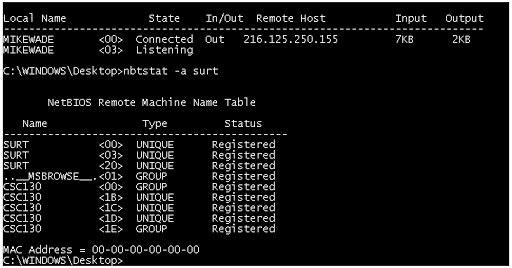
The -a Switch

Microsoft Windows uses an interface called **Network Basic Input/Output System (NetBIOS)**, which relates names with workstations and is an upper-layer interface that requires a transport protocol—usually TCP/IP. But IPv6 can be used as well. Deploying the **nbtstat** utility will achieve these three important things:

1. Track NetBIOS over TCP/IP statistics
2. Show the details of incoming and outgoing NetBIOS over TCP/IP connections
3. Resolve NetBIOS names

Understand that because NetBIOS name resolution is primarily a Windows network utility, the nbtstat command is available only in Windows-based operating systems.

Making use of the –a switch will get you a remote machine’s NetBIOS name table consisting of a list of every NetBIOS name the machine from which you’ve deployed the switch knows of. The –a switch produced the output from server surt shown below.



|  |
| --- |
| *Sample output of the nbtstat –a command* |

So, using this switch arranges the NetBIOS name-table information in table form with output in four columns. The **Name column** displays the NetBIOS name entry for the remote host machine. The next column gives you a unique two-digit hexadecimal identifier for the NetBIOS name. This identifier represents the last byte of the NetBIOS name depicted in the Name column, and it’s important because the same name could actually be used several times for the same machine. Plus, it identifies the specific service on the particular host that the name is referencing. The below tables list the hexadecimal identifiers for unique and group hostnames. The **Type column** refers to the type of NetBIOS name being referenced. Unique NetBIOS names refer to individual hosts, and group names refer to logical groupings of workstations—either domains or workgroups. The **Status column** gives you information about the status of a host’s NetBIOS even if it hasn’t been registered with the rest of the network.

| **Hex ID** | **Description** |
| --- | --- |
| 00 | General name for the computer. |
| 03 | Messenger service ID used to send messages between a WINS server and a workstation. This is the ID registered with a WINS server. |
| 06 | Remote Access Server (RAS) server service ID. |
| 21 | RAS client. |
| 53 | DNS. |
| 123 | Network Time Protocol (NTP). |
| 1B | Domain master browser ID. A NetBIOS name with this ID indicates the domain master browser. |
| 1F | Network Dynamic Data Exchange (NetDDE) service ID. |
| BE | Network monitor agent ID. |
| BF | Network monitor utility ID. |
| 01 | Master browser for a domain to other master browsers. |
| 20 | Internet group name ID. This ID is registered with the WINS server to indicate which computers are used for administrative purposes. |
| 1C | Domain group name ID. |
| 1D | Master browser name. |
| 1E | Normal group name. |

### The -A Switch

The –A switch works just like the –a switch and will give you the same output, but the syntax of the command is different. Obviously, you use an uppercase A instead of a lowercase one, and you also have to include the host’s IP address instead of its NetBIOS name. To use it, type nbtstat followed by –A and finally the IP address of the specific host whose NetBIOS table you want to check out:

nbtstat –A 199.153.163.2

The -c Switch

Use the –c switch to display the local NetBIOS name cache on the workstation it’s running on. The below figure shows a sample output of the nbtstat –c command.

|  |
| --- |
| *Sample output of the nbtstat –c command* |

Each entry in this display shows the NetBIOS name, the hex ID for the service that was accessed, the type of NetBIOS name (unique or group), the IP address that the name resolves to, and its life. The *Life* value shows how many seconds each entry will live in the cache. When this time expires, the entry will be deleted.

The -n Switch

The –n switch will give you the local NetBIOS name table on a Windows device. The below figure shows an output that’s similar to the output of the –a switch except for one important thing: What you’re seeing is the NetBIOS name table for the machine you’re running the command on instead of that of another host.

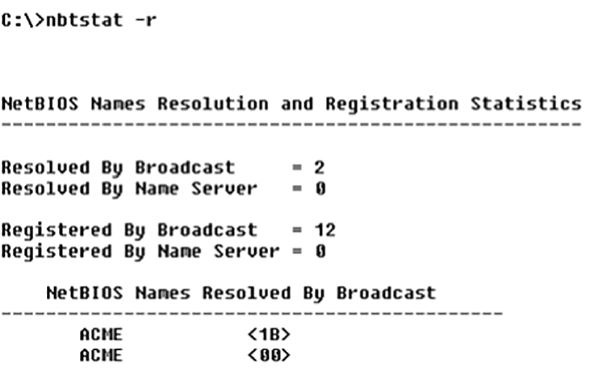
|  |
| --- |
| *Sample output of the nbtstat –n command* |

### The -r Switch

This switch is probably the one you’ll use most often when you want to get a hold of NetBIOS over TCP/IP (NBT) statistics because it tells you exactly how many NetBIOS names have been resolved to TCP/IP addresses. The below figure shows the sample output of the nbtstat –r command.

What you can see here is that the statistics are divided into two categories. First, there are the NetBIOS names resolution and registration statistics. This is how many names have been resolved or registered either by broadcasts on the local segment or via lookup from a WINS name server.

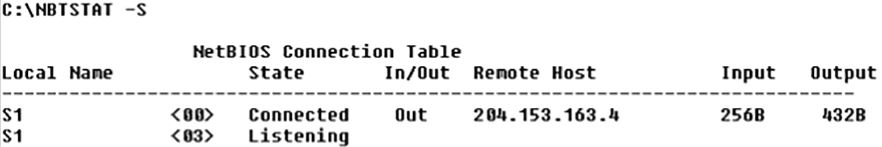
Next, you have the NetBIOS unique and group names and their associated hex IDs that were resolved or registered. In the figure, you can see that there’s a distinct lack of information regarding names resolved by a name server. What this means is that the output is telling you that there’s no WINS server operating—instead, all NetBIOS names were resolved by broadcast only.



Sample output of the *nbtstat –r* command

### The -S Switch

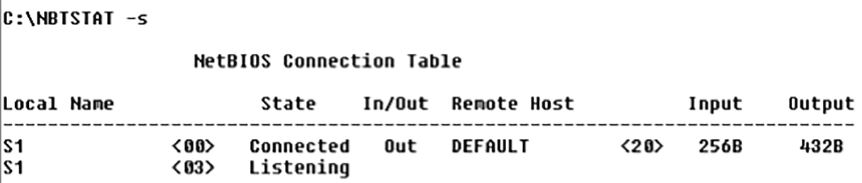
Using the -S switch will display the NetBIOS sessions table that lists all NetBIOS sessions to and from the host from which you issued the command. The -S switch displays both workstation and server sessions but lists remote addresses by IP address only. The below figure shows the sample output of the nbtstat -S command.



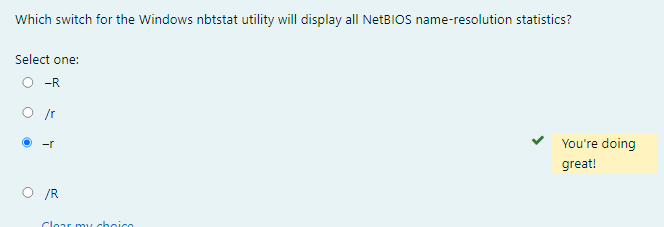
Here you can see the NetBIOS name is displayed along with its hex ID and the status of each session. An entry in the In/Out column determines whether the connection has been initiated from the computer on which you’re running nbtstat (outbound) or whether another computer has initiated the connection (inbound). The numbers in the Input and Output columns indicate in bytes the amount of data transferred between the stations.

### The -s Switch

As with the -A and -a switches, the lowercase -s switch is similar to its uppercase sibling. The nbtstat -s command produces the same output as nbtstat -S except that it will also attempt to resolve remote-host IP addresses into hostnames. The below figure shows the sample output from the nbtstat -s command.



Sample output of the *nbtstat -s* command



The netstat Utility

Using netstat is a great way to **check out the inbound and outbound TCP/IP connections** on your machine. You can also use it **to view packet statistics like** how many packets have been sent and received, the number of errors, and so on. When used without any options, netstat produces output similar to the following, which shows all the outbound TCP/IP connections. This utility is a great tool to use to determine the status of outbound web connections. Take a look:

C:\Users\clarusway>netstat

Active Connections

Proto Local Address Foreign Address State

TCP 192.168.1.22:49812 ec2-35-157-203-133:https ESTABLISHED

TCP 192.168.1.22:49824 ed-in-f188:5228 ESTABLISHED

TCP 192.168.1.22:50322 server-99-86-243-78:https ESTABLISHED

TCP 192.168.1.22:50918 54.239.31.91:https ESTABLISHED

TCP 192.168.1.22:51180 aeab55d76dd13c9bb:https ESTABLISHED

TCP 192.168.1.22:51211 ec2-18-205-93-210:https ESTABLISHED

TCP 192.168.1.22:51212 ec2-52-202-62-236:https CLOSE\_WAIT

TCP 192.168.1.22:51213 ec2-18-205-93-141:https CLOSE\_WAIT

TCP 192.168.1.22:51214 ec2-18-205-93-141:https CLOSE\_WAIT

TCP 192.168.1.22:51215 ec2-18-205-93-141:https CLOSE\_WAIT

TCP 192.168.1.22:51216 ec2-18-205-93-141:https CLOSE\_WAIT

TCP 192.168.1.22:51281 aeab55d76dd13c9bb:https ESTABLISHED

TCP 192.168.1.22:51318 52.46.68.59:https ESTABLISHED

TCP 192.168.1.22:51346 ec2-3-225-75-90:https ESTABLISHED

TCP 192.168.1.22:51377 52.114.128.43:https ESTABLISHED

TCP 192.168.1.22:51391 aeab55d76dd13c9bb:https ESTABLISHED

TCP 192.168.1.22:61298 ec2-52-202-62-228:https ESTABLISHED

TCP 192.168.1.22:61317 ec2-3-120-198-117:https ESTABLISHED

TCP 192.168.1.22:61320 ec2-3-120-198-117:https ESTABLISHED

TCP 192.168.1.22:61330 ec2-3-120-198-117:https ESTABLISHED

TCP 192.168.1.22:62010 51.105.249.228:https ESTABLISHED

The **Proto column** lists the protocol being used. The **Local Address** column lists the source address and the source port (source socket). The **Foreign Address** column lists the address of the destination machine (the hostname if it’s been resolved). If the destination port is known, it will show up as a well-known port. The **State column** indicates the status of each connection. This column shows statistics only for TCP connections because the *User Datagram Protocol (UDP)* establishes no virtual circuit to the remote device. Usually, this column indicates **ESTABLISHED** when a TCP connection between your computer and the destination computer has been established.

**💡Tip:**

* If the address of either your computer or the destination computer can be found in the HOSTS file on your computer, the destination computer’s name, rather than the IP address, will show up in either the Local Address or Foreign Address column.

The output of the netstat utility depends on the switch. By using the netstat /? command, we can see the options available to us.

C:\Users\clarusway>netstat /?

Displays protocol statistics and current TCP/IP network connections.

NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-x] [-t] [interval]

-a Displays all connections and listening ports.

-b Displays the executable involved in creating each connection or

listening port. In some cases well-known executables host

multiple independent components, and in these cases the

sequence of components involved in creating the connection

or listening port is displayed. In this case the executable

name is in [] at the bottom, on top is the component it called,

and so forth until TCP/IP was reached. Note that this option

can be time-consuming and will fail unless you have sufficient

permissions.

-e Displays Ethernet statistics. This may be combined with the -s

option.

-f Displays Fully Qualified Domain Names (FQDN) for foreign

addresses.

-n Displays addresses and port numbers in numerical form.

-o Displays the owning process ID associated with each connection.

-p proto Shows connections for the protocol specified by proto; proto

may be any of: TCP, UDP, TCPv6, or UDPv6. If used with the -s

option to display per-protocol statistics, proto may be any of:

IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.

-q Displays all connections, listening ports, and bound

nonlistening TCP ports. Bound nonlistening ports may or may not

be associated with an active connection.

-r Displays the routing table.

-s Displays per-protocol statistics. By default, statistics are

shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6;

the -p option may be used to specify a subset of the default.

-t Displays the current connection offload state.

-x Displays NetworkDirect connections, listeners, and shared

endpoints.

-y Displays the TCP connection template for all connections.

Cannot be combined with the other options.

interval Redisplays selected statistics, pausing interval seconds

between each display. Press CTRL+C to stop redisplaying

statistics. If omitted, netstat will print the current

configuration information once.

The -a Switch

When you use the -a switch, the netstat utility displays all TCP/IP connections and all UDP connections. Below you can see the output produced by the netstat -a command.

C:\Users\clarusway>netstat -a

Active Connections

Proto Local Address Foreign Address State

TCP 192.168.1.22:49812 ec2-35-157-203-133:https ESTABLISHED

TCP 192.168.1.22:49824 ed-in-f188:5228 ESTABLISHED

TCP 192.168.1.22:50322 server-99-86-243-78:https ESTABLISHED

TCP 192.168.1.22:51211 ec2-18-205-93-210:https ESTABLISHED

TCP 192.168.1.22:51212 ec2-52-202-62-236:https CLOSE\_WAIT

TCP 192.168.1.22:51213 ec2-18-205-93-141:https CLOSE\_WAIT

TCP 192.168.1.22:51214 ec2-18-205-93-141:https CLOSE\_WAIT

TCP 192.168.1.22:51215 ec2-18-205-93-141:https CLOSE\_WAIT

TCP 192.168.1.22:51216 ec2-18-205-93-141:https CLOSE\_WAIT

TCP 192.168.1.22:51518 ec2-54-236-84-111:https ESTABLISHED

TCP 192.168.1.22:51548 185.11.14.41:http TIME\_WAIT

TCP 192.168.1.22:51549 185.11.14.41:http TIME\_WAIT

TCP 192.168.1.22:51550 185.11.14.41:http TIME\_WAIT

TCP 192.168.1.22:51563 99.86.243.5:https ESTABLISHED

TCP 192.168.1.22:51564 ec2-3-225-75-90:https ESTABLISHED

TCP 192.168.1.22:51579 52.114.132.73:https ESTABLISHED

TCP 192.168.1.22:51585 aeab55d76dd13c9bb:https ESTABLISHED

TCP 192.168.1.22:51597 server-99-86-245-89:https ESTABLISHED

TCP 192.168.1.22:61298 ec2-52-202-62-228:https ESTABLISHED

TCP 192.168.1.22:61317 ec2-3-120-198-117:https ESTABLISHED

TCP 192.168.1.22:61320 ec2-3-120-198-117:https ESTABLISHED

TCP 192.168.1.22:61330 ec2-3-120-198-117:https ESTABLISHED

TCP 192.168.1.22:62010 51.105.249.228:https ESTABLISHED

UDP [fe80::19ac:8efb:2c6e:f512%10]:1900 \*:\*

UDP [fe80::19ac:8efb:2c6e:f512%10]:2177 \*:\*

UDP [fe80::19ac:8efb:2c6e:f512%10]:58133 \*:\*

You can tell that UDP connections in the output are broadcasts because the destination address is listed as \* : \* (meaning “any address, any port”).

The most common use for the -a switch is to check the status of a TCP/IP connection that appears to be hung. You can determine if the connection is simply busy or is actually hung and no longer responding.

**💡Tip:**

* The **State** column in the figure has no entry for the UDP rows because UDP is not a connection-oriented protocol and, therefore, has no connection state.

The -e Switch

The -e switch displays a summary of all the packets that have been sent over the Network Interface Card (NIC) as of that instant. The Received and Sent columns show packets coming in as well as being sent:

C:\Users\clarusway>netstat -e

Interface Statistics

Received Sent

Bytes 652308520 724669536

Unicast packets 7476729 5597781

Non-unicast packets 6906 240780

Discards 0 0

Errors 0 1

Unknown protocols 0

You can use the -e switch to display the following categories of statistics:

* **Bytes** - The number of bytes transmitted or received since the computer was turned on. This statistic is useful for finding out if data is actually being transmitted and received or if the network interface isn’t doing anything at all.
* **Unicast Packets** - The number of packets sent from or received at this computer. To register in one of these columns, the packet must be addressed directly from one computer to another and the computer’s address must be in either the source or destination address section of the packet.
* **Non-unicast Packets** - The number of packets that weren’t directly sent from one workstation to another. For example, a broadcast packet is a non-unicast packet. The number of non-unicast packets should be smaller than the number of unicast packets. If the number of non-unicast packets is as high as or higher than that of unicast packets, too many broadcast packets are being sent over your network. Definitely find the source of these packets and make any necessary adjustments to optimize performance.
* **Discards** - The number of packets that were discarded by the NIC during either transmission or reception because they weren’t assembled correctly.
* **Errors** - The number of errors that occurred during transmission or reception. (These numbers may indicate problems with the network card.)
* **Unknown Protocols** - The number of received packets that the Windows networking stack couldn’t interpret. This statistic only shows up in the Received column because if the computer sent them, they wouldn’t be unknown.

Unfortunately, statistics don’t mean much unless they can be colored with time information. For example, if the Errors row shows 1 error, is that a problem? It might be if the computer has been on for only a few minutes. Unfortunately, the netstat utility doesn’t have a way of indicating how much time has elapsed for these statistics.

### The -r Switch

You use the -r switch to display the current route table for a workstation so that you can see exactly how TCP/IP information is being routed.

C:\Users\clarusway>netstat -r

===========================================================================

Interface List

14...9c 5c 8e ce d9 c9 ......Intel(R) I211 Gigabit Network Connection

18...9c 5c 8e ce d9 ca ......Intel(R) Ethernet Connection (2) I219-V

15...76 c6 3b 00 62 86 ......Microsoft Wi-Fi Direct Virtual Adapter

8...76 c6 3b 00 6a 86 ......Microsoft Wi-Fi Direct Virtual Adapter #2

10...74 c6 3b 00 62 86 ......Broadcom 802.11ac Network Adapter

1...........................Software Loopback Interface 1

17...00 00 00 00 00 00 00 e0 Microsoft Teredo Tunneling Adapter

===========================================================================

IPv4 Route Table

===========================================================================

Active Routes:

Network Destination Netmask Gateway Interface Metric

0.0.0.0 0.0.0.0 192.168.1.1 192.168.1.22 35

127.0.0.0 255.0.0.0 On-link 127.0.0.1 331

127.0.0.1 255.255.255.255 On-link 127.0.0.1 331

127.255.255.255 255.255.255.255 On-link 127.0.0.1 331

192.168.1.0 255.255.255.0 On-link 192.168.1.22 291

192.168.1.22 255.255.255.255 On-link 192.168.1.22 291

192.168.1.255 255.255.255.255 On-link 192.168.1.22 291

224.0.0.0 240.0.0.0 On-link 127.0.0.1 331

224.0.0.0 240.0.0.0 On-link 192.168.1.22 291

255.255.255.255 255.255.255.255 On-link 127.0.0.1 331

255.255.255.255 255.255.255.255 On-link 192.168.1.22 291

===========================================================================

Persistent Routes:

None

IPv6 Route Table

===========================================================================

Active Routes:

If Metric Network Destination Gateway

17 331 ::/0 On-link

1 331 ::1/128 On-link

17 331 2001::/32 On-link

17 331 2001:0:2851:782c:148e:f3fd:6aff:55b8/128

On-link

10 291 fe80::/64 On-link

17 331 fe80::/64 On-link

17 331 fe80::148e:f3fd:6aff:55b8/128

On-link

10 291 fe80::19ac:8efb:2c6e:f512/128

On-link

1 331 ff00::/8 On-link

10 291 ff00::/8 On-link

17 331 ff00::/8 On-link

===========================================================================

Persistent Routes:

None

### The -s Switch

Using the -s switch displays a variety of TCP, UDP, IP, and ICMP protocol statistics. But be warned—the output you’ll get is really long, which may or may not be okay for you.

C:\Users\clarusway>netstat -s

IPv4 Statistics

Packets Received = 85199526

Received Header Errors = 0

Received Address Errors = 113

Datagrams Forwarded = 0

Unknown Protocols Received = 49

Received Packets Discarded = 9859

Received Packets Delivered = 85614599

Output Requests = 60765459

Routing Discards = 0

Discarded Output Packets = 5954

Output Packet No Route = 202

Reassembly Required = 10

Reassembly Successful = 4

Reassembly Failures = 0

Datagrams Successfully Fragmented = 0

Datagrams Failing Fragmentation = 0

Fragments Created = 0

IPv6 Statistics

Packets Received = 261062

Received Header Errors = 0

Received Address Errors = 321

Datagrams Forwarded = 0

Unknown Protocols Received = 0

Received Packets Discarded = 981

Received Packets Delivered = 263904

Output Requests = 244350

Routing Discards = 0

Discarded Output Packets = 539

Output Packet No Route = 0

Reassembly Required = 0

Reassembly Successful = 0

Reassembly Failures = 0

Datagrams Successfully Fragmented = 0

Datagrams Failing Fragmentation = 0

Fragments Created = 0

ICMPv4 Statistics

Received Sent

Messages 4104 8133

Errors 0 0

Destination Unreachable 2443 5570

Time Exceeded 437 0

Parameter Problems 0 0

Source Quenches 0 0

Redirects 0 0

Echo Replies 1224 0

Echos 0 2563

Timestamps 0 0

Timestamp Replies 0 0

Address Masks 0 0

Address Mask Replies 0 0

Router Solicitations 0 0

Router Advertisements 0 0

ICMPv6 Statistics

Received Sent

Messages 934 2128

Errors 0 0

Destination Unreachable 184 185

Packet Too Big 0 0

Time Exceeded 11 0

Parameter Problems 0 0

Echos 0 24

Echo Replies 0 0

MLD Queries 0 0

MLD Reports 0 0

MLD Dones 0 0

Router Solicitations 0 92

Router Advertisements 27 0

Neighbor Solicitations 360 1445

Neighbor Advertisements 376 382

Redirects 0 0

Router Renumberings 0 0

TCP Statistics for IPv4

Active Opens = 67905

Passive Opens = 1709

Failed Connection Attempts = 40609

Reset Connections = 5402

Current Connections = 43

Segments Received = 64039477

Segments Sent = 47069056

Segments Retransmitted = 162012

TCP Statistics for IPv6

Active Opens = 258

Passive Opens = 86

Failed Connection Attempts = 222

Reset Connections = 56

Current Connections = 0

Segments Received = 5728

Segments Sent = 4332

Segments Retransmitted = 717

UDP Statistics for IPv4

Datagrams Received = 22383699

No Ports = 6416

Receive Errors = 260328

Datagrams Sent = 14243567

UDP Statistics for IPv6

Datagrams Received = 267432

No Ports = 966

Receive Errors = 15

Datagrams Sent = 239189

### The -p Switch

Like the -n switch, the -p switch is a modifier that’s usually used with the -s switch to specify which protocol statistics to list in the output (IP, TCP, UDP, or ICMP). For example, if you want to view only ICMP statistics, you use the -p switch like so:

netstat -s -p ICMP

The netstat utility then displays the ICMP statistics instead of the entire gamut of TCP/IP statistics that the -s switch will typically flood you with. For a different example, let’s use the -s and -p switches to retrieve some IPv6 information:

C:\Users\clarusway>netstat -s -p IPV6

IPv6 Statistics

Packets Received = 261062

Received Header Errors = 0

Received Address Errors = 321

Datagrams Forwarded = 0

Unknown Protocols Received = 0

Received Packets Discarded = 981

Received Packets Delivered = 263904

Output Requests = 244359

Routing Discards = 0

Discarded Output Packets = 539

Output Packet No Route = 0

Reassembly Required = 0

Reassembly Successful = 0

Reassembly Failures = 0

Datagrams Successfully Fragmented = 0

Datagrams Failing Fragmentation = 0

Fragments Created = 0

### The -n Switch

The -n switch is a modifier for the other switches. When used with them, it reverses the natural tendency of netstat to use names instead of network addresses. In other words, when you use the -n switch, the output always displays network addresses instead of their associated network names. Following is output from the netstat command used with the netstat -n command. It’s showing the same information but with IP addresses instead of names:

C:\Users\clarusway>netstat -n

Active Connections

Proto Local Address Foreign Address State

TCP 192.168.1.22:49812 35.157.203.133:443 ESTABLISHED

TCP 192.168.1.22:49824 74.125.143.188:5228 ESTABLISHED

TCP 192.168.1.22:52352 18.205.93.208:443 ESTABLISHED

TCP 192.168.1.22:52354 18.205.93.141:443 CLOSE\_WAIT

TCP 192.168.1.22:52355 52.202.62.236:443 CLOSE\_WAIT

TCP 192.168.1.22:52356 18.205.93.141:443 CLOSE\_WAIT

TCP 192.168.1.22:52357 18.205.93.141:443 CLOSE\_WAIT

TCP 192.168.1.22:52358 18.205.93.141:443 CLOSE\_WAIT

TCP 192.168.1.22:52458 99.86.243.78:443 ESTABLISHED

TCP 192.168.1.22:52499 52.114.77.34:443 ESTABLISHED

TCP 192.168.1.22:52508 18.205.44.71:443 ESTABLISHED

TCP 192.168.1.22:52515 75.2.53.94:443 ESTABLISHED

TCP 192.168.1.22:52517 75.2.53.94:443 ESTABLISHED

TCP 192.168.1.22:52566 99.83.135.170:443 ESTABLISHED

TCP 192.168.1.22:52584 99.86.245.105:443 ESTABLISHED

TCP 192.168.1.22:52602 99.86.243.98:443 ESTABLISHED

TCP 192.168.1.22:52617 99.83.135.170:443 ESTABLISHED

TCP 192.168.1.22:52668 52.114.7.36:443 TIME\_WAIT

TCP 192.168.1.22:52679 3.225.75.90:443 ESTABLISHED

TCP 192.168.1.22:52688 99.83.135.170:443 ESTABLISHED

TCP 192.168.1.22:52705 52.155.169.137:443 TIME\_WAIT

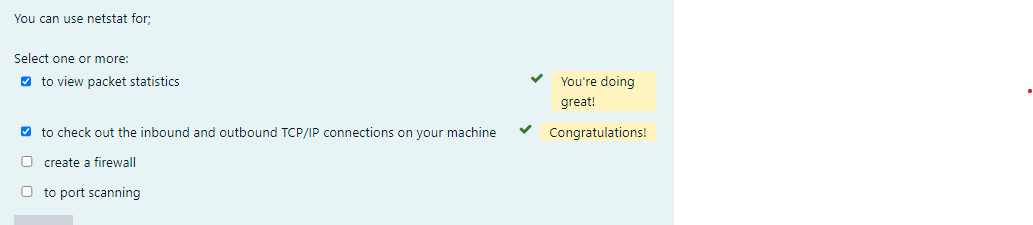
TCP 192.168.1.22:61298 52.202.62.228:443 ESTABLISHED

TCP 192.168.1.22:61317 3.120.198.117:443 ESTABLISHED

TCP 192.168.1.22:61320 3.120.198.117:443 ESTABLISHED

TCP 192.168.1.22:61330 3.120.198.117:443 ESTABLISHED

TCP 192.168.1.22:62010 51.105.249.228:443 ESTABLISHED

Formun Üstü

Using tcpdump

The tcpdump utility is used to read either packets captured live from a network or packets that have been saved to a file. Although there is a Windows version called windump, tcpdump only works on Unix-like operating systems.

* Use this command to capture traffic on all interfaces:

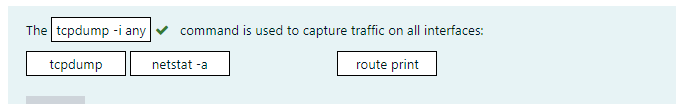
# tcpdump -i any

* Here is the command to capture traffic on a particular interface:

# tcpdump -i eth0

* And to filter traffic by IP, whether it’s the source or the destination, use this command:

# tcpdump host 192.168.5.5



Using the File Transfer Protocol

**File Transfer Protocol (FTP)** is a subset of TCP/IP and that FTP is used for the transfer of files. In recent years, FTP has become a truly cross-platform protocol for transferring files. Almost every client and server platform has implemented FTP. Windows is no exception. Its TCP/IP stack comes with a command-line ftp utility.

To start the ftp utility, enter ftp at a command prompt. The result is an ftp command prompt:

C:\Users\clarusway>ftp

ftp>

From this prompt, you can open a connection to an FTP server and upload and download files as well as change the way FTP operates. To display a list of all the commands you can use at the ftp command prompt, type help or ? and press Enter. To get help on a specific command, type help, a space, and then the name of the command. Here is some output from the help command:

C:\Users\clarusway>ftp

ftp> ?

Commands may be abbreviated. Commands are:

! delete literal prompt send

? debug ls put status

append dir mdelete pwd trace

ascii disconnect mdir quit type

bell get mget quote user

binary glob mkdir recv verbose

bye hash mls remotehelp

cd help mput rename

close lcd open rmdir

**💡Tip:**

* Third-party applications are available that provide a GUI interface for FTP, which is easier to use than a command line.

Starting FTP and Logging In to an FTP Server

Of the two FTP file operations (download and upload), the ability to download files is definitely the more crucial for you to have down as a network technician or sysadmin.

The first steps in starting an FTP download session are to determine the address of the FTP site and start the ftp utility. The FTP site typically has the same name as the website except that the first three characters are ftp instead of www. For example, Microsoft’s website is www.microsoft.com. Its FTP site, on the other hand, is ftp.microsoft.com.

First, start the ftp utility as demonstrated in the preceding section, and then follow these steps:

1. At the ftp command prompt, type open, a space, and the name of the FTP server, like this:

C:\Users\clarusway> ftp

ftp> open ftp.claruswaytrainer.com

Connected to ftp.claruswaytrainer.com.

220---------- Welcome to Pure-FTPd [TLS] ----------

220-You are user number 1 of 100 allowed.

220-Local time is now 11:45. Server port: 21.

220-IPv6 connections are also welcome on this server.

220 You will be disconnected after 15 minutes of inactivity.

User (ftp.claruswaytrainer.com:(none)): enter

230 Anonymous user logged in

ftp>

As shown here, if the FTP server is available and running, you’ll receive a response welcoming you to the server and asking you for a username. Right now, we used *Anonymous* as the username (enabled by default on the FTP server), which means that anyone can log in to it.

You can also start an FTP session by typing ftp, a space, and the address of the FTP server. This allows you to start the ftp utility and open a connection in one step. Here’s an example:

C:\Users\clarusway> ftp ftp.claruswaytrainer.com

1. Enter a valid username, and press Enter.
2. Enter your password, and press Enter.

If you enter the wrong username and/or password, the server will tell you so by displaying the following and leaving you at the ftp command prompt:

530 Login Incorrect

Login failed.

This means you’ve got to try again and must start the login process over. If you’re successful, the FTP server will welcome you and drop you back at the ftp command prompt.

Downloading Files

After you log in to the FTP server, you’ll navigate to the directory that contains the files you want. The FTP command-line interface is similar to the DOS command-line interface.

The below table lists and describes the common navigation commands for FTP. After you navigate to the directory and find the file you want to download, it’s time to set the parameters for the type of file. Files come in two types:

* ASCII, which contains text
* Binary, which is all other files

If you set ftp to the wrong type, the file you download will contain gibberish. So if you’re in doubt, set ftp to download files as binary files. Check out the below table.

| **Command** | **Description** |
| --- | --- |
| ls | Short for list. Displays a directory listing. Very similar to the DIR command in MS-DOS. |
| cd | Short for change directory. Works almost identically to the MS-DOS CD command. Use it to change to a different directory and navigate the server’s directory structure. |
| pwd | Short for print working directory. Displays the current directory on the server. Useful if you forget where you are when changing to several locations on the server. |
| lcd | Short for local change directory. Displays and changes the current directory on the local machine. Useful when you are downloading a file and aren’t in the directory where you want to put the file. |

To set the file type to ASCII, type ascii at the ftp command prompt. ftp will respond by telling you that the file type has been set to A (ASCII):

ftp>ascii

Type set to A

To set the file type to binary, type binary at the ftp command prompt. ftp will respond by telling you that the file type has been set to I (binary):

ftp>binary

Type set to I

To download the file, just use the get command like this:

ftp>get test.exe

200 PORT command successful.

150 Opening BINARY mode data connection for 'test.exe'

(567018 bytes).

The file will start downloading to your hard drive. Unfortunately, with its default settings, the ftp utility doesn’t give you any indication of the progress of the transfer. When the file has downloaded, the ftp utility will display the following message and return you to the ftp command prompt:

226 Transfer complete.

567018 bytes received in 116.27 seconds (4.88 Kbytes/sec)

Uploading Files

To upload a file to an FTP server, you’ve got to have rights on that specific server. These rights are assigned on a directory-by-directory basis. To upload a file, log in and then follow these steps:

1. At the ftp command prompt, type lcd to navigate to the directory on the local machine where the file resides.
2. Type cd to navigate to the destination directory.
3. Set the file type to ASCII or binary.
4. Use the put command to upload the file.

The syntax of the put command looks like this:

ftp> put local file destination file

Let’s say you want to upload a file called test.txt on the local server but you want it to be called my.txt on the destination server. To accomplish that, use the following command:

ftp> put test.txt my.txt

You’ll get the following response:

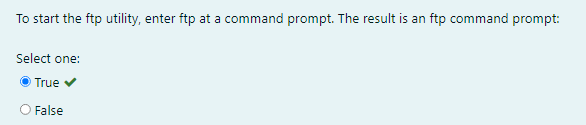
200 PORT command successful.

150 Opening BINARY mode data connection for my.txt

226 Transfer complete.

743622 bytes sent in 0.55 seconds (1352.04 Kbytes/sec)

**💡Tips:**

* You can upload multiple files using the **mput** command. Simply type **mput**, a space, and then a wildcard that specifies the files. For example, to upload all the text files in a directory, type **mput \*.txt**. And in the same way, you can download multiple files with **mget** command.
* **get** and **put** commands require the perfect match of the file names intented to be downloaded and uploaded. On the other hand, **mget** and **mput** commands can run on partial match, a pattern with wildcards or asteriks.
* 

Formun Altı

Using the Telnet Utility

Part of the TCP/IP protocol suite, **Telnet** is a virtual terminal protocol utility that allows you to make connections to remote devices, gather information, and run programs. Telnet was originally developed to open terminal sessions from remote Unix workstations to Unix servers. Although it’s still used for that purpose, we now use it as a troubleshooting tool as well. The below figure shows the basic Telnet interface as it’s being used to start a terminal session on a remote Unix host.

|  |
| --- |
| *The Telnet* |

In today’s Windows environments, Telnet is a basic command-line tool for testing TCP connections. You can telnet to any TCP port to see if it’s responding—something that’s especially useful when checking *Simple Mail Transfer Protocol (SMTP)* and *HTTP* (web) ports.

How to Enable Telnet in Windows

Because most people have the Windows 10 operating system running on their PCs these days, it’s good to know that, by default, these operating systems install without Telnet available. But there’s a way around that one—if you really must have a Telnet client enabled in these operating systems, here’s how to do it:

1. Open Control Panel.
2. Select Programs And Features.
3. In the left column, select Turn Windows Features On Or Off
4. Select the Telnet checkbox (and any other obscure services you may want enabled), and wait while Windows installs for a while and then reboots.

Now you can go to Start and then type telnet in the Start search box to get a Telnet window to open for you. You can also open a DOS prompt and just type telnet from there. Here are the options that Windows provides with Telnet:

Microsoft Telnet> ?

Commands may be abbreviated. Supported commands are:

c - close close current connection

d - display display operating parameters

o - open hostname [port] connect to hostname (default port 23).

q - quit exit telnet

set - set set options (type 'set ?' for a list)

sen - send send strings to server

st - status print status information

u - unset unset options (type 'unset ?' for a list)

?/h - help print help information

Don’t Use Telnet, Use Secure Shell

Telnet is totally insecure because it sends all data in crystal-clear text—including your name and password. If Microsoft doesn’t even enable it on its latest OSs, then you know it really must be insecure.

So if you shouldn’t use Telnet, what should you use instead? **Secure Shell (SSH)** is your answer. It provides the same options as Telnet, plus a lot more; but most importantly, it doesn’t send any data in cleartext. The thing is, your servers, routers, and other devices need to be enabled with SSH, and it’s not configured by default on most devices.

Some configuration is usually necessary if you want things to work as they really should, and yes, sometimes it’s a little painful to get everything running smoothly, but it’s all worth it in the long run.

### Secure Shell (ssh)

The **secure shell** or **ssh** is a collection of tools using a secure protocol for communications with remote computers.

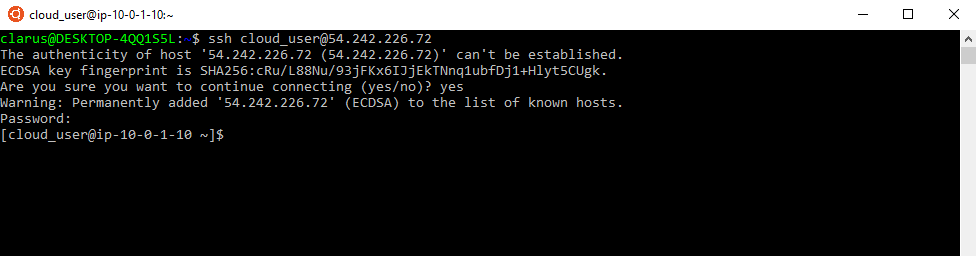
It is a protocol used to securely connect to a remote server/system. ssh is secure in the sense that it transfers the data in encrypted form between the host and the client. It transfers inputs from the client to the host and relays back the output. ssh runs at TCP/IP port 22.

ssh user\_name@host(IP/Domain-name)

ssh command instructs the system to establish an encrypted secure connection with the host machine.

* user\_name represents the account that is being accessed on the host.
* host refers to the machine which can be a computer or a router that is being accessed. It can be an IP address (e.g. 54.164.151.235) or domain name(e.g. www.clarusway.com).

**Example:**



Q: How do you use **ssh** to connect to a **remote** server in Linux.  
A: Most servers in the world are run on Linux servers. They’re dependable, affordable and highly configurable. However, servers aren’t always accessed, nor accessible, directly. Hence they require remote access. The most frequently used, and secure, method of accessing servers remotely is via SSH, otherwise known as Secure Shell.  
For connect to remote server, first we should open to linux terminal and then type  
ssh user@www.remote\_server\_name.com  
or  
ssh [user@82.178.72.19](mailto:user@82.178.72.19)

scp Command

scp (Secure Copy) is a command-line tool that is used to transfer files and directories across the systems securely over the network. When we use scp command to copy files and directories from our local system to a remote system then in the backend it makes **ssh connection** to a remote system.

Syntax:

scp <options> <files\_or\_directories> user@target\_host:/<folder>

scp <options> user@target\_host:/files <folder\_local\_system>

Some of the most widely used options in scp command are listed below:

| **command** | **Explanation** |
| --- | --- |
| -C | Enable Compression |
| -i | identity File or private key |
| -l | limit the bandwidth while copying |
| -P | ssh port number of the target host |
| -r | Copy files and directories recursively |
| -p | Preserves modification times, access times, and modes from the original file |
| -q | Disables the progress meter |

Examples:

* Copies the file "test.txt" from a remote host to the localhost:

scp your\_username@hostname:text.txt /some/local/directory

* Copies the file "test.txt" from the local host to a remote host:

scp text.txt your\_username@hostname:/some/local/directory

* Copies multiple files from the remote host to your current directory on the localhost:

scp text.txt your\_username@hostname: /some/local/directory/\{a,b,c\}

Q: How to copy a file from a remote server to a local machine?  
A: If you are on the local computer wanting to receive file from a remote computer:  
**scp** username@remote:/file/to/send /where/to/put

Using The scp and curl Utility

curl Command

curl is a command-line tool to transfer data to or from a server, using any of the supported protocols.

Syntax:

curl [options] [URL...]

user@clarusway:~$ curl https://www.clarusway.com

* -o: Saves the downloaded file on the local machine with the name provided in the parameters.
* curl -o [file\_name] [URL...]
* user@clarusway:~$ curl -o hello.zip ftp://speedtest.tele2.net/1MB.zip

### Network Configuration Files

The graphical help tools use a few basic commands to edit a specific set of network configuration files. The exact names and location of the configuration files in the file system depend largely on your distribution and version of Linux.

| **File** | **Description** |
| --- | --- |
| /etc/resolv.conf | List DNS servers for internet domain name resolution. Manual page for: /etc/resolv.conf |
| /etc/hosts | Lists hosts to be resolved locally (not by DNS). Manual page for: /etc/hosts |
| /etc/nsswitch.conf | List order of host name search. Typically look at local files, then NIS server, then DNS server. Manual page for: /etc/nsswitch.conf |
| Red Hat/Fedora/CentOS: /etc/sysconfig/network | Specify network configuration. eg. Static IP, DHCP, NIS, etc. |
| Red Hat/Fedora/CentOS: /etc/sysconfig/network-scripts/ifcfg-device | Specify TCP network information. |
| Ubuntu/Debian: /etc/network/interfaces | Specify network configuration and devices. eg. Static IP and info, DHCP, etc. |

### The /etc/sysconfig/network file

* The /etc/sysconfig/network file is a global (across all network cards) configuration file. It allows us to define whether we want networking (NETWORKING=yes|no), what the hostname should be (HOSTNAME=) and which gateway to use (GATEWAY=).
* Note that this file contains no settings at all in a default RHEL7 install (with networking enabled).

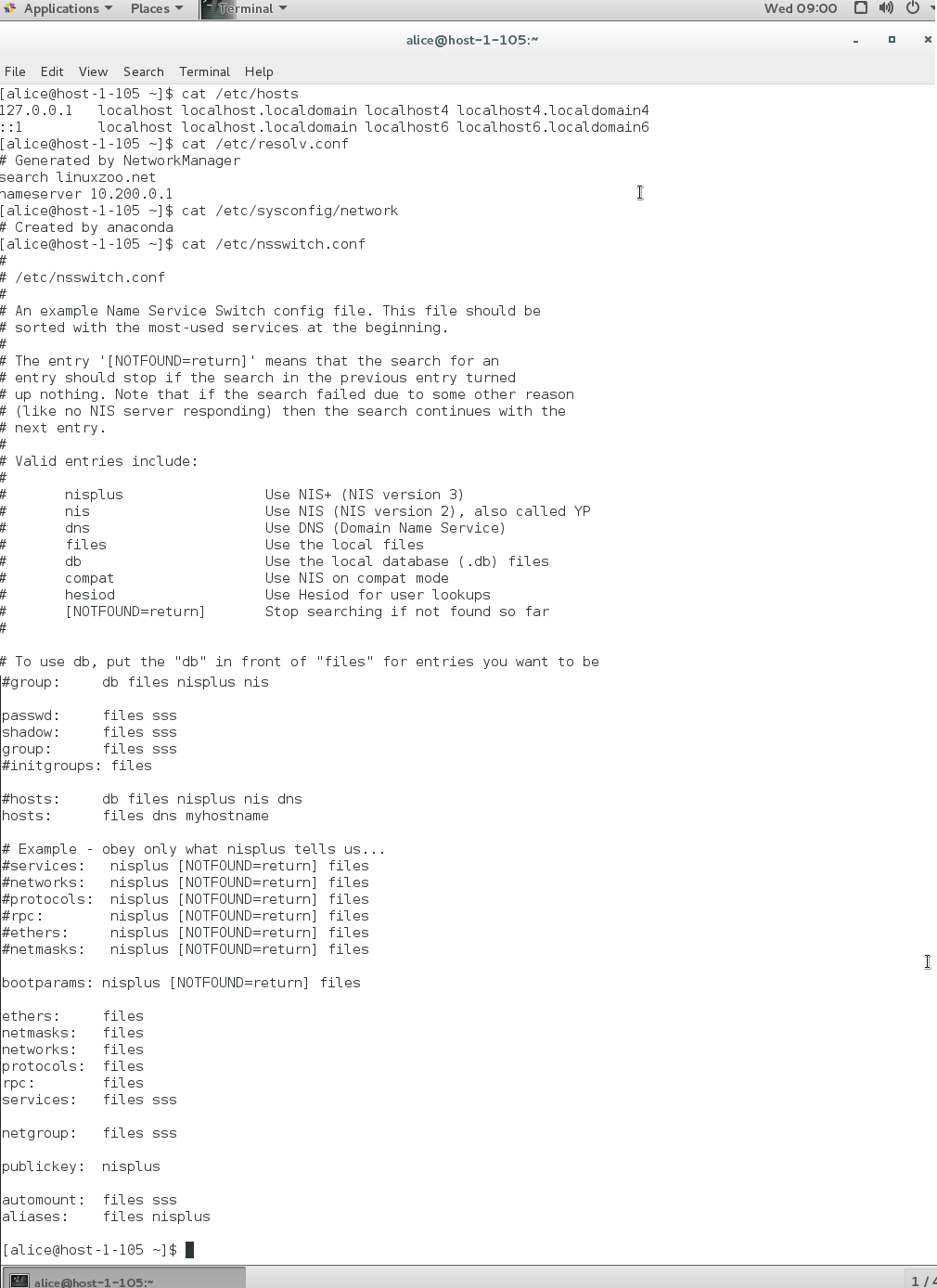
### The /etc/hosts file

The main purpose of /etc/hosts configuration file is to resolve hostnames that cannot be resolved any other way. It can also be used to resolve hostnames on small networks with no DNS server.

### The /etc/resolv.conf file

This file is used for configuring the DNS (Domain Name System) resolver library. The resolv.conf configuration file contains information parameters used by the DNS resolver. The DNS resolver allows for the operating system to translate domain names into IP addresses.

Example:

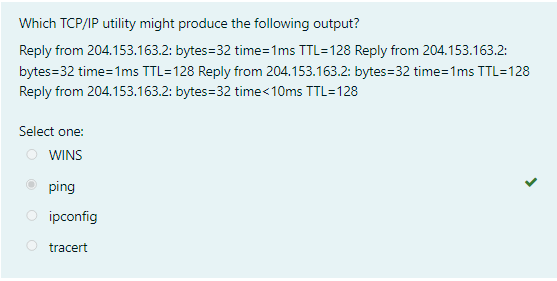


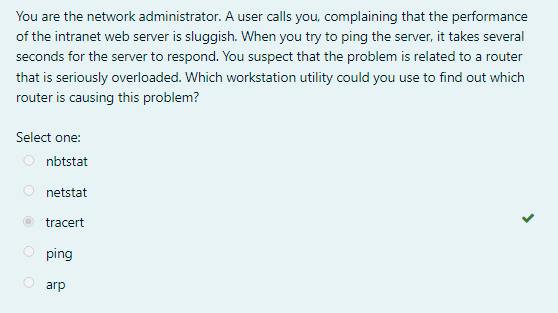
Typical default contents:

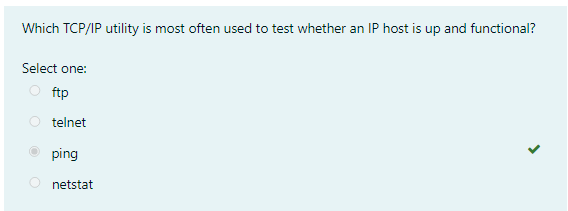
| **Directive** | **Description** |
| --- | --- |
| auto | Indicates the device should be setup at boot time |
| lo | Loopback interface |
| iface | Interface |
| eth0 | Ethernet device 0, typically the primary network adaptor |
| inet | Indicates network adaptor has an IPv4 address space |
| dhcp | Network adaptor gets its configuration from a DHCP server |
| static | Indicates the adaptor uses fixed IP configuration |
| address | The IP address of the host |
| netmask | Network subnet mask |
| gateway | Gateway Address |
| network | The network portion of the IP address |
| nameserver | The IP of a DNS |

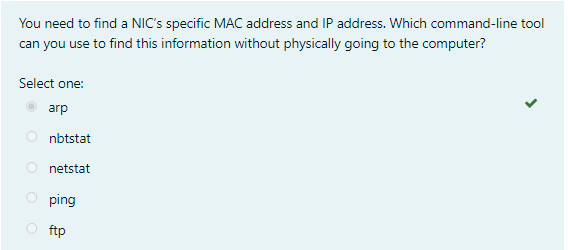
**💡Tips:**

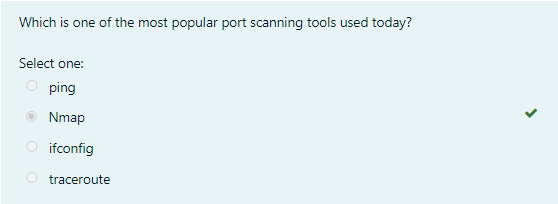
* The loopback (lo) interface will have an IP address of 127.0.0.1, which represents the host itself. Suppose you want to open a web page running on the same Linux server you are on. You could open http://127.0.0.1 in your web browser. That IP address won’t be accessible over the network.
* The ethernet 0 (eth0) interface is typically the connection to the local network. Even if you are running Linux in a virtual machine (VM), you’ll still have an eth0 interface that connects to the physical network interface of the host. Most commonly, you should ensure that eth0 is in an IP state and has an IP address so that you can communicate with the local network and likely over the Internet.

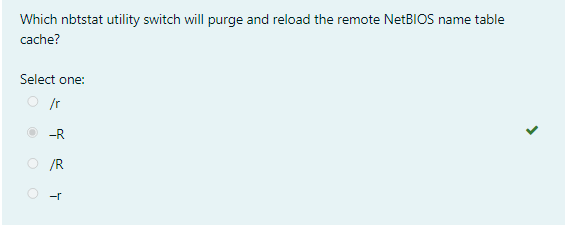


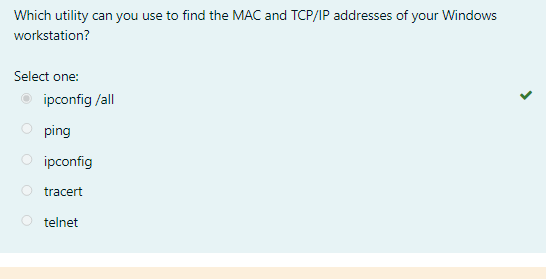


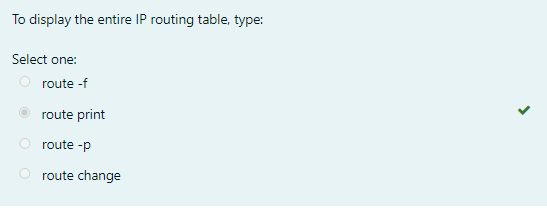
Formun Üstü









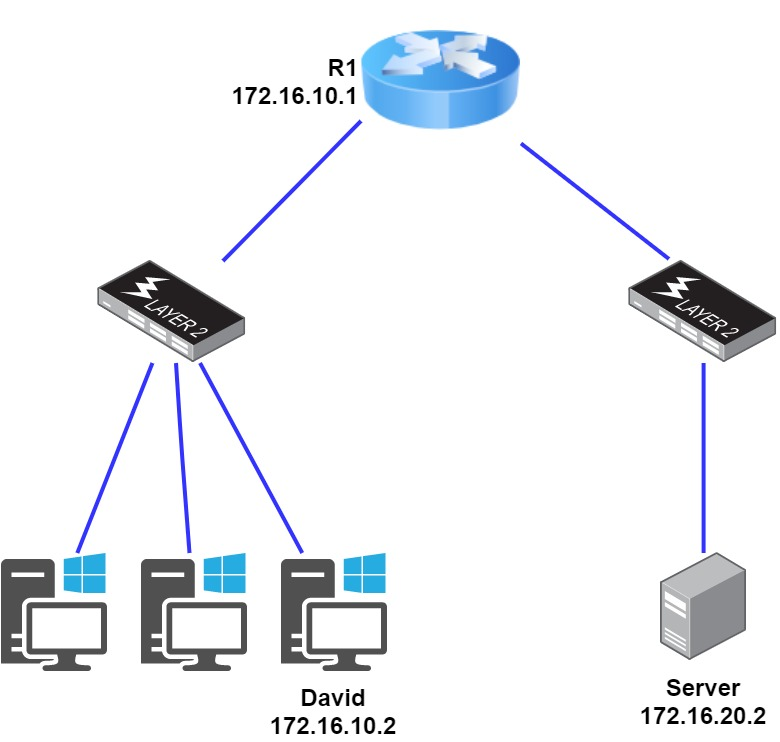


## Troubleshooting IP Addressing (Optional)

### Troubleshooting IP Addressing

Troubleshooting [IP addressing](https://lms.clarusway.com/mod/resource/view.php?id=17542) is obviously an important skill because running into trouble somewhere along the way is pretty much a sure thing, and it’s going to happen to you.

Let’s use the below figure as an example of basic IP trouble— David can’t log in to the Windows server.



|  |
| --- |
| *Basic IP troubleshooting* |

Let’s get started by going over the basic troubleshooting steps.

1. Open a command prompt window on David’s host, and ping 127.0.0.1.

C:\>ping 127.0.0.1

Pinging 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

This is the diagnostic, or loopback address, and if you get a successful ping, your IP stack is considered to be initialized. If it fails, then you have an IP stack failure and need to reinstall TCP/IP on the host.

**💡Tips:**

* If you ping the loopback address and receive an **“unable to contact IP driver, error code 2”** message, you need to reinstall the TCP/IP protocol suite on the host.

1. Now, from the same command prompt window, ping the IP address of the localhost.

C:\>ping 172.16.10.2

Pinging 172.16.10.2 with 32 bytes of data:

Reply from 172.16.10.2: bytes=32 time<1ms TTL=128

Reply from 172.16.10.2: bytes=32 time<1ms TTL=128

Reply from 172.16.10.2: bytes=32 time<1ms TTL=128

Reply from 172.16.10.2: bytes=32 time<1ms TTL=128

Ping statistics for 172.16.10.2:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

If that’s successful, your network interface card (NIC) is functioning. If it fails, there is a problem with the NIC.

1. From the command prompt window, ping the default gateway (router).

C:\>ping 172.16.10.1

Pinging 172.16.10.1 with 32 bytes of data:

Reply from 172.16.10.1: bytes=32 time<1ms TTL=128

Reply from 172.16.10.1: bytes=32 time<1ms TTL=128

Reply from 172.16.10.1: bytes=32 time<1ms TTL=128

Reply from 172.16.10.1: bytes=32 time<1ms TTL=128

Ping statistics for 172.16.10.1:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

If the ping works, it means that the NIC is plugged into the network and can communicate on the local network. If it fails, you have a local physical network problem that could be anywhere from the NIC to the router.

1. If steps 1 through 3 were successful, try to ping the remote server.

C:\>ping 172.16.20.2

Pinging 172.16.20.2 with 32 bytes of data:

Reply from 172.16.20.2: bytes=32 time<1ms TTL=128

Reply from 172.16.20.2: bytes=32 time<1ms TTL=128

Reply from 172.16.20.2: bytes=32 time<1ms TTL=128

Reply from 172.16.20.2: bytes=32 time<1ms TTL=128

Ping statistics for 172.16.20.2:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

If that works, then you know that you have IP communication between the local host and the remote server. You also know that the remote physical network is working.

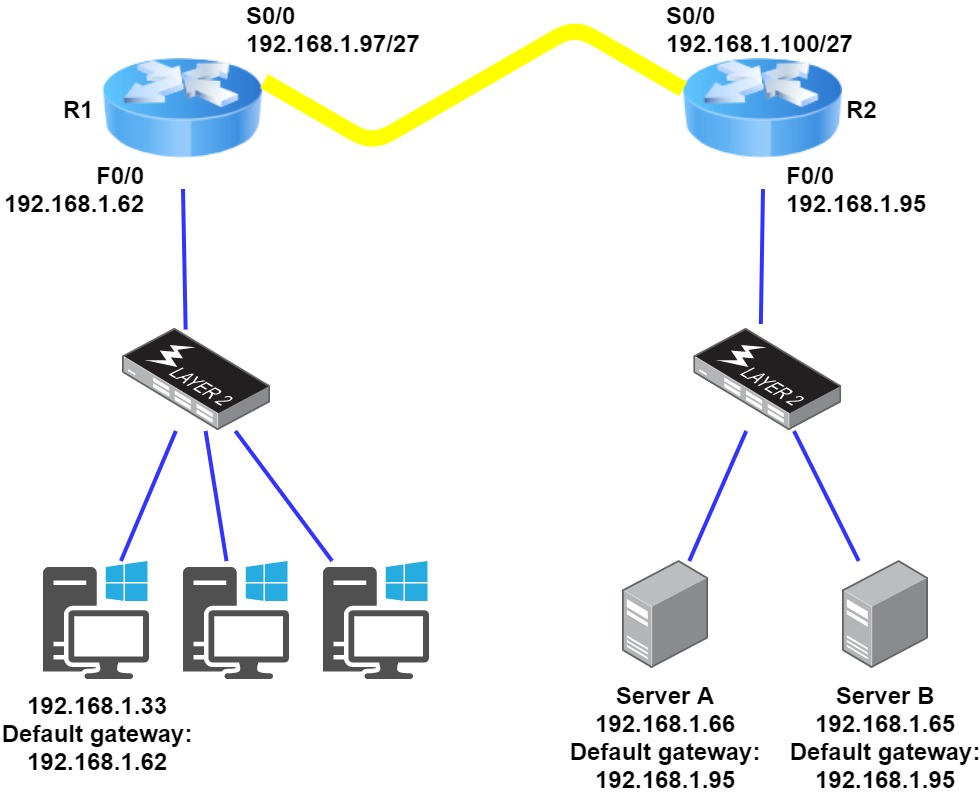
If the user still can’t communicate with the server after steps 1 through 4 are successful, you probably have some type of name resolution problem and need to check your **Domain Name System (DNS)** settings. But if the ping to the remote server fails, then you know you have some type of remote physical network problem and need to go to the server and work through steps 1 through 3 until you find the snag.

## Troubleshooting IP Addressing (Optional)

### Determining IP Address Problems-1

It’s common for a host, router, or other network devices to be configured with the wrong IP address, subnet mask, or default gateway. Because this happens way too often, in this example you will learn how to both determine and fix IP address configuration errors.

After you’ve worked through the four basic steps of troubleshooting and determined there’s a problem, you obviously then need to find and fix it. Once you have your network accurately drawn out, including the [IP addressing](https://lms.clarusway.com/mod/resource/view.php?id=17542) scheme, you need to verify each host’s IP address, mask, and default gateway address to determine the problem. Let’s check out the below example illustration. A user calls and tells you that he can’t get to Server A. You ask him if he can get to Server B in the marketing department, but he doesn’t know because he doesn’t have the rights to log on to that server.



|  |
| --- |
| *IP address problem* |

You ask the client to go through the four troubleshooting steps that you learned about in the preceding section. Steps 1 through 3 work, but step 4 fails. First, the WAN link between the R1 and the R2 shows the mask as a /27. You should already know that this mask is 255.255.255.224 and then determine that all networks are using this mask. The network address is 192.168.1.0. What are our valid subnets and hosts?

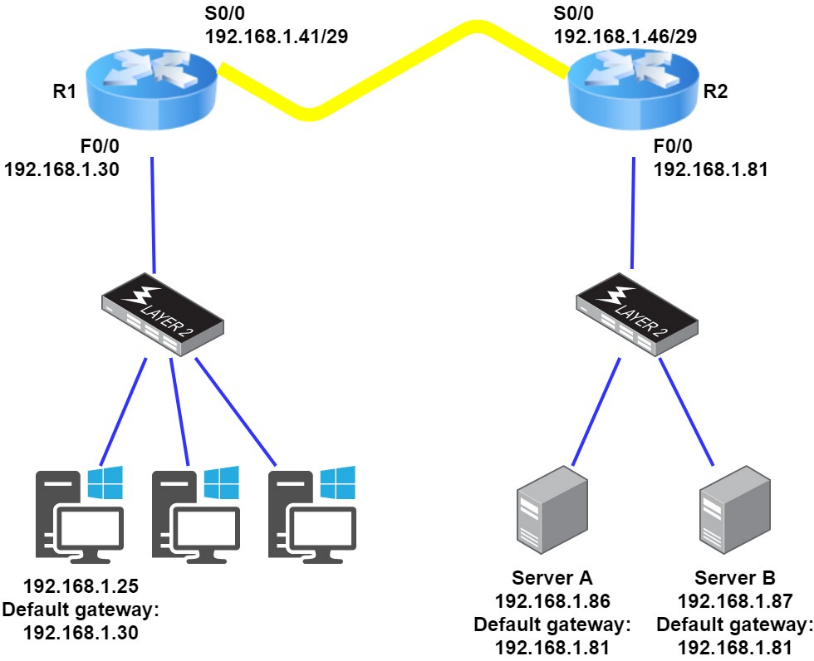
256 – 224 = 32,

so this makes our subnets 0, 32, 64, 96, 128, and so on. So, by looking at the figure, you can see that subnet 32 is being used by the clients, the WAN link is using subnet 96, and the servers are using subnet 64. Now you have to determine what the valid host ranges are for each subnet. The valid hosts for the Sales LAN are 33 through 62—the broadcast address is 63 because the next subnet is 64. For the servers, the valid hosts are 65 through 94 (broadcast 95), and for the WAN link, 97 through 126 (broadcast 127). By looking at the figure, you can determine that the default gateway on the Lab\_B router is incorrect. That address is the broadcast address of the 64 subnet, so there’s no way it could be a valid host.

## Troubleshooting IP Addressing (Optional)

### Determining IP Address Problems-2

The below figure shows a network problem. A user can’t get to Server B. You have the user run through the four basic troubleshooting steps and find that the host can communicate to the local network but not to the remote network. Find and define the [IP addressing](https://lms.clarusway.com/mod/resource/view.php?id=17542) problem.



|  |
| --- |
| *IP address problem* |

If you use the same steps used to solve the last problem, you can see first that the WAN link again provides the subnet mask to use— /29, or 255.255.255.248. You need to determine what the valid subnets, broadcast addresses, and valid host ranges are to solve this problem.

The 248 mask is a block size of 8 (256 – 248 = 8), so the subnets both start and increment in multiples of 8. By looking at the figure, you see that the user is in the 24 subnet, the WAN is in the 40 subnet, and the servers are in the 80 subnet. Can you see the problem yet? The valid host range for the user's LAN is 25–30, and the configuration appears correct. The valid host range for the WAN link is 41–46, and this also appears correct. The valid host range for the 80 subnet is 81–86, with a broadcast address of 87 because the next subnet is 88. Server B has been configured with the broadcast address of the subnet. Now that you can figure out misconfigured IP addresses on hosts, what do you do if a host doesn’t have an IP address and you need to assign one? What you need to do is look at other hosts on the LAN and figure out the network, mask, and default gateway.

Let’s take a look at a couple of examples of how to find and apply valid IP addresses to hosts. You need to assign a server and router IP address on a LAN. The subnet assigned on that segment is 192.168.1.24/29, and the router needs to be assigned the first usable address and the server the last valid host ID. What are the IP address, mask, and default gateway assigned to the server?

To answer this, you must know that a /29 is a 255.255.255.248 mask, which provides a block size of 8. The subnet is known as 24, the next subnet in a block of 8 is 32, so the broadcast address of the 24 subnet is 31, which makes the valid host range 25–30:

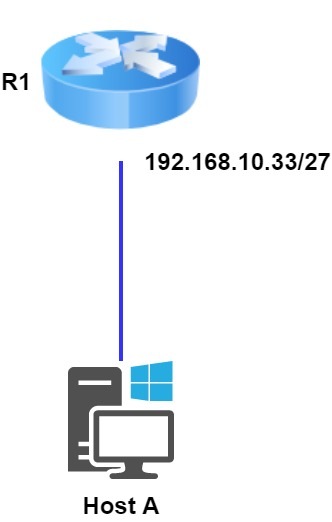
Server IP address: 192.168.1.30

Server mask: 255.255.255.248

Default gateway: 192.168.1.25 (router’s IP address)

## Troubleshooting IP Addressing (Optional)

### Determining IP Address Problems-3



|  |
| --- |
| *Invalid Host* |

Look at the router’s IP address on Ethernet0. What IP address, subnet mask, and valid host range could be assigned to the host?

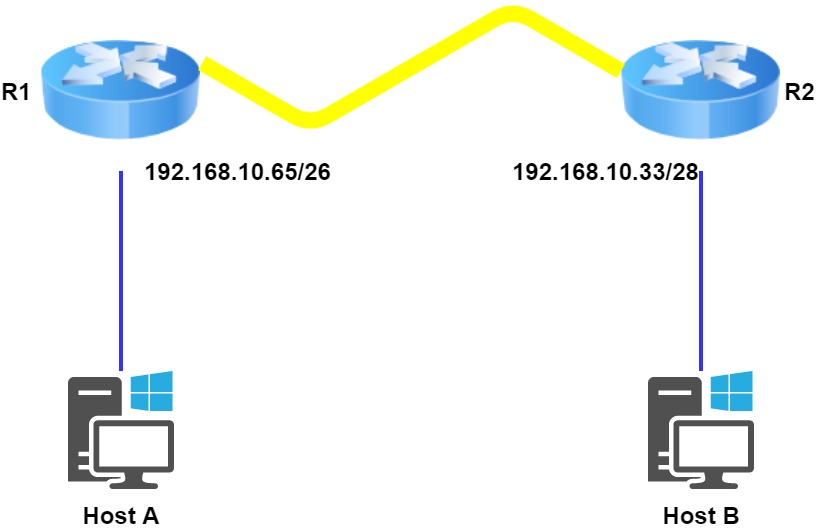
The IP address of the router’s Ethernet0 is 192.168.10.33/27. As you already know, a /27 is a 224 mask with a block size of 32. The router’s interface is in the 32 subnet. The next subnet is 64, so that makes the broadcast address of the 32 subnet 63 and the valid host range 33–62:

Host IP address: 192.168.10.34–62 (any address in the range except for 33, which is assigned to the router)

Mask: 255.255.255.224

Default gateway: 192.168.10.33

The below figure shows two routers with Ethernet configurations already assigned. What are the host addresses and subnet masks of hosts A and B?



|  |
| --- |
| *nvalid Host* |

Router 1 has an IP address of 192.168.10.65/26 and Router 2 has an IP address of 192.168.10.33/28. What are the host configurations?

Router 1 Ethernet0 is in the 192.168.10.64 subnet, and Router 2 Ethernet0 is in the 192.168.10.32 network:

HostA IP address: 192.168.10.66–126

HostA mask: 255.255.255.192

HostA default gateway: 192.168.10.65

HostB IP address: 192.168.10.34–46

HostB mask: 255.255.255.240

HostB default gateway: 192.168.10.33