File transfer and login using IPv6,
plus “What to do when things don’t work”

Introduction

Usually file transfers to remote computers and logins just work. But sometimes they don’t. This article reviews the use of file transfer and login applications. It also describes ways to identify and solve problems using the dig, nslookup, ping, and traceroute/tracert commands.

Internet Protocols (**IP**) and domain names

To access remote computers, a language called an IP is used. For many years there has been only one version called IP version 4 (**IPv4**). In the last few years, a new version called IP version 6 (**IPv6**) has appeared. Remote computers can be accessed using either IPv4 or IPv6 so long as your computer, the local network, the internet in between, the remote network, and the remote computer all agree on which IP version to use. Problems arise when all five don’t agree.

People usually refer to a computer by its **domain name**, such as ftp.microsoft.com. Applications refer to a computer using its **IP address**. An IPv4 address looks like 64.4.30.33, which consists of up to 4 groups of 3-digit decimal numbers separated by periods (or “dots”). An IPv6 address looks like 2001:4860:8011::63, which consists of up to 8 groups of 4-digit hexadecimal numbers separated by colons.

File transfer and login applications can access a remote computer using its domain name, its IPv4 address, or one of its IPv6 addresses. A computer that uses IPv6 always has more than one IPv6 addresses. Sometimes a computer may have more than one IPv4 address. Most of the time it is only necessary to know one address, but occasionally it is helpful to be aware that there may be more than one when problem solving.

Getting application help

Applications use either a command line interface or a graphical user interface (**GUI**). The general form of a command line is:

```
% command_name options
```

Most applications using a command line interface have a built-in help function. To display a summary of the application’s options, enter the command_name on a line by itself (without any options). If that just starts the application without displaying anything, exiting the application by entering “**bye**” or “**quit**” or “**exit**” and then entering this command line usually works:

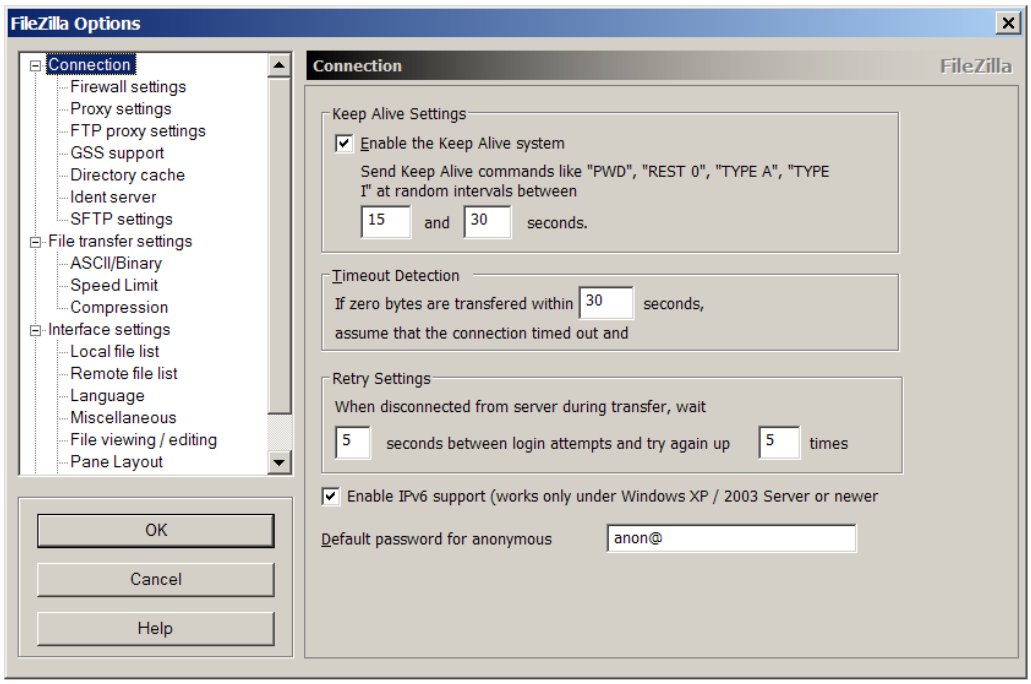
```
% command_name -help
```

Most applications using a GUI also have a built-in help function. To display a summary of the application’s options, start the application and click on the “Help” button. This button is usually (though not always) near the top of the GUI window. Sometimes there will be a “?” in the upper right hand corner of the window. Clicking on the “?” and then on a button or box in the window often displays help for that specific part of the window.

File transfer application review

File transfer applications such as **ftp** and **mpscp** use a command line interface while **FileZilla** uses a graphical user interface (**GUI**).

The **FileZilla** GUI must first be configured to enable the use of IPv6 (if not already configured). Start the application and click on the “Edit” button, then on the “Settings” button. In the Options window that appears next, make sure the “Enable IPv6 support” checkbox is checked as shown below, and then click on “OK”.



GUI parameters and resulting output	Notes
Click on “File”, then “Site Manager”, then “New Site”. Enter ftp.microsoft.com in the “Host” box. Next, select a “Logontype” of “normal”. Then enter your username on the remote computer in the “User:” box and click on “Connect”. Look for “Status: Connected” in the output.	A domain name was used. It may also be necessary to select a “Servertype” of “SFTP using SSH2”
Click on “File”, then “Site Manager”, then “New Site”. Enter 64.4.30.34 in the “Host” box. Next, select a “Logontype” of “normal”. Then enter your username on the remote computer in the “User:” box and click on “Connect”. Look for “Status: Connected” in the output.	An IPv4 address was used. It may also be necessary to select a “Servertype” of “SFTP using SSH2”
Click on “File”, then “Site Manager”, then “New Site”. Enter [2001:480:150:75::5222] in the “Host” box. Next, select a “Logontype” of “normal”. Then enter your username on the remote computer in the “User:” box and click on “Connect”. Look for “Status: Connected” in the output.	An IPv6 address was used. It may also be necessary to select a “Servertype” of “SFTP using SSH2”. Enter the square brackets around the IPv6 address in the “Host” box as shown to the left
As above. Look for “Cannot connect” in the output.	The remote computer may not understand IPv6

Example **ftp** command lines are shown below. If your username on the local computer (user1) and on the remote computer (user2) are not the same, enter “user2” in response to the “Name” prompt from the remote computer. It may also be necessary to enter a password in response to a “Password:” prompt.

Command line and resulting output	Notes
% ftp ftp.microsoft.com Connected to ftp.microsoft.com	A domain name was used
% ftp ftp.microsoft.com Connected to ftp.microsoft.com Name (ftp.microsoft.com:user1):	A domain name was used. The usernames on the local and remote computers are different, so enter “user2” in response to the “Name” prompt
% ftp 64.4.30.34 Connected to 64.4.30.34	An IPv4 address was used
% ftp 2001:480:150:75::5222 Connected to 2001:480:150:75::5222	An IPv6 address was used
% ftp 2001:480:150:75::5037 Unknown error	The remote computer may not understand IPv6

Example **mpscp** command lines are shown below. If your username on the local computer (user1) and on the remote computer (user2) are not the same, place “user2@” in front of the remote computer name. It may also be necessary to enter a password in response to a “Password:” prompt.

Command line and resulting output	Notes
% mpscp file1 ftp.microsoft.com:file2 x bytes received in 0.y seconds (z Kbytes/s)	A domain name was used. A summary of the file transfer will be shown
% mpscp file1 user2@ftp.microsoft.com:file2 x bytes received in 0.y seconds (z Kbytes/s)	A domain name was used. The usernames on the local and remote computers are different, so enter “user2@” in front of the remote computer. A summary of the file transfer will be shown
% mpscp file1 64.4.30.34:file2 x bytes received in 0.y seconds (z Kbytes/s)	An IPv4 address was used. A summary of the file transfer will be shown
% mpscp -o “AddressFamily inet6” file1 2001:480:150:75::5222:./file2 x bytes received in 0.y seconds (z Kbytes/s)	An IPv6 address was used. A summary of the file transfer will be shown. The use of IPv6 addresses may not be supported on all computers
% mpscp -o “AddressFamily inet6” file1 ftp.microsoft.com:file2 x bytes received in 0.y seconds (z Kbytes/s)	mpscp will try to use an IPv6 address for the transfer. A summary of the file transfer will be shown
% mpscp file1 2001:480:150:75::5222:./file2 ssh: connect to 2001:480:150:75::5037 port 22: Connection refused	The remote computer may not understand mpscp or IPv6

Login application review

Login applications like **{k}rlogin**, **{k}rsh**, **{k}telnet,** and **ssh** use a command line interface while **PuTTY** uses a GUI. Note: {k}command_name is used to indicate that on some computers it *may* be necessary to put a “k” in front of the command_name.

Example **{k}rlogin** command lines are shown below. If your username on the local computer (user1) and on the remote computer (user2) are not the same, add a “-l” user2 at the end of the command line. That’s a dash and the letter L, not a dash and the number 1. It may also be necessary to enter a password in response to a “Password:” prompt.

Command line and resulting output	Notes
% {k}rlogin ftp.microsoft.com This rlogin session is encrypting all data transmissions.	A domain name was used
% {k}rlogin ftp.microsoft.com -l user2 This rlogin session is encrypting all data transmissions.	A domain name was used. The usernames on the local and remote computers are different, so put a dash L and your remote username after the computer name
% {k}rlogin 64.4.30.34 This rlogin session is encrypting all data transmissions.	An IPv4 address was used
% {k}rlogin 2001:480:150:75::5222 This rlogin session is encrypting all data transmissions.	An IPv6 address was used
% {k}rlogin 2001:480:150:75::5037 Unknown error	The remote computer may not understand IPv6

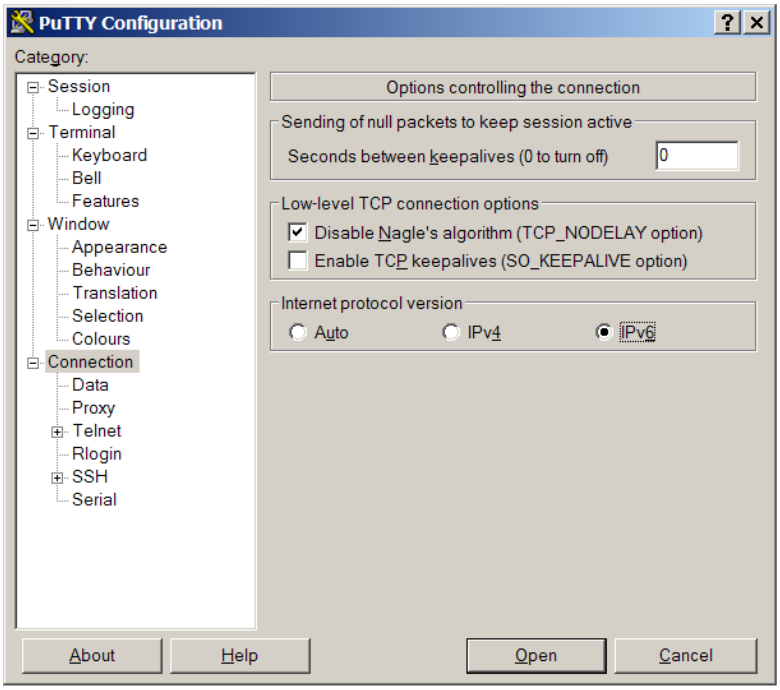
Example **{k}rsh** command lines are shown below. If your username on the local computer (user1) and on the remote computer (user2) are not the same, add a “-l” user2 to the command line. Use a dash and the letter L, not a dash and the number 1. It may also be necessary to enter a password in response to a “Password:” prompt.

Command line and resulting output	Notes
% {k}rsh ftp.microsoft.com *command* This rlogin session is encrypting all data transmissions.	A domain name was used
% {k}rsh ftp.microsoft.com -l user2 *command* This rlogin session is encrypting all data transmissions.	A domain name was used. The usernames on the local and remote computers are different, so put a dash L and your remote username after the remote computer name
% {k}rsh -l user2 ftp.microsoft.com *command* This rlogin session is encrypting all data transmissions.	Alternatively, the dash L and your remote username may precede the computer name
% {k}rsh 64.4.30.34 *command* This rlogin session is encrypting all data transmissions.	An IPv4 address was used.
% {k}rsh 2001:480:150:75::5222 *command* This rlogin session is encrypting all data transmissions.	An IPv6 address was used.
% {k}rsh 2001:480:150:75::5037 *command* Unknown error	The remote computer may not understand IPv6

Example **{k}telnet** command lines are shown below. If your username on the local computer (user1) and on the remote computer (user2) are not the same, add a “-l” user2 to the command line. Use a dash and the letter L, not a dash and the number 1. It may also be necessary to enter a password in response to a “Password:” prompt.

Command line and resulting output	Notes
% {k}telnet ftp.microsoft.com Trying 64.4.30.34… Connected to ftp.microsoft.com (64.4.30.34).	A domain name was used. An IP address and the domain name of the remote computer will be shown.
% {k}telnet -l user2 ftp.microsoft.com Trying 64.4.30.34… Connected to ftp.microsoft.com (64.4.30.34).	A domain name was used. The usernames on the local and remote computers are different, so put a dash L and your remote username before the remote computer name. An IP address and the domain name of the remote computer will be shown
% {k}telnet 64.4.30.34 Trying 64.4.30.34… Connected to ftp.microsoft.com (64.4.30.34).	An IPv4 address was used. An IP address and the domain name of the remote computer will be shown
% {k}telnet 2001:480:150:75::5222 Trying 2001:480:150:75::5222… Connected to pingo.arsc.edu (2001:480:150:75::5222).	An IPv6 address was used. An IP address and the domain name of the remote computer will be shown
% {k}telnet 2001:480:150:75::5037 telnet: Unable to connect to remote host: Connection refused	The remote computer may not allow remote telnet connections or it may not understand IPv6

The **PuTTY** GUI must first be configured to enable the use of IPv6 (if not already configured). Start the application and click on the “Connection” category. The “Internet protocol version” should have either the “Auto” or the “IPv6” radio button selected. When “Auto” is selected, PuTTY will try connecting using IPv6 if it is available, and then it will try IPv4. When “IPv6” is selected, PuTTY will only try to connect using IPv6.



GUI parameters and resulting output	Notes
Click on the “Session” category. Enter ftp.microsoft.com in the “Host name (or IP address)” box. Click on “Open”. Look for a new command line window to open.	A domain name was used
Click on the “Session” category. Enter ftp.microsoft.com in the “Host name (or IP address)” box. Click on “Open”. Look for a new command line window to open.	A domain name was used. If your usernames on the local and remote computers are not the same, click on the “Data” subcategory below the “Connection” category, and enter “user2” in the “Auto-login username” **before** clicking on the “Open” button
Click on the “Session” category. Enter 64.4.30.34 in the “Host name (or IP address)” box. Click on “Open”. Look for a new command line window to open.	An IPv4 address was used
Click on the “Session” category. Enter 2001:480:150:75::5222 in the “Host name (or IP address)” box. Click on “Open”. Look for a new command line window to open.	An IPv6 address was used
Click on the “Session” category. Enter 2001:480:150:75::5222 in the “Host name (or IP address)” box. Click on “Open”. An error is displayed. “Unable to open connection to ftp.microsoft.com. Host does not exist.”	The remote computer may not understand IPv6

Example **ssh** command lines are shown below. The **ssh** command has a lot of options; only a few of which are shown. If your username on the local computer (user1) and on the remote computer (user2) are not the same, add a "-l" user2 to the command line or place "user2@" in front of the remote computer name (but not both). Use a dash and the letter L, not a dash and the number 1. It may also be necessary to enter a password in response to a "Password:" prompt.

Command line and resulting output	Notes
% ssh ftp.microsoft.com	A domain name was used. SSh will silently connect.
% ssh user2@ftp.microsoft.com	A domain name was used. The usernames on the local and remote computers are different, so enter "user2@" in front of the remote computer name. SSH will silently connect
% ssh -l user2 ftp.microsoft.com	Alternatively, put a dash L and "user2" before the remote computer name. SSh will silently connect
% ssh 64.4.30.34	An IPv4 address was used. SSh will silently connect
% ssh -4 ftp.microsoft.com	SSh will try to use an IPv4 address to connect. SSh will silently connect
% ssh 2001:480:150:75::5222	An IPv6 address was used. SSh will silently connect
% ssh -6 ftp.microsoft.com	SSh will try to use an IPv6 address to connect. SSh will silently connect
% ssh -o "AddressFamily inet6" ftp.microsoft.com	SSh will try to use an IPv6 address to connect. SSH will silently connect
% ssh 2001:480:150:75::5037 ssh: Could not resolve hostname midnight.arsc.edu: Name or service not known	The remote computer may not understand IPv6
% ssh ftp.microsoft.com Received disconnect from ftp.microsoft.com: 2: Your ssh version is not allowed. Please install a newer version.	The local computer has an old version of the ssh command in the default path, so the path of a current version of ssh must be specified. Use % /usr/brl/bin/ssh ftp.microsoft.com at the ARL DSRC, for example

What to do when things don't work

When you receive an error message like "Cannot Connect" or "Unknown error" or "Connection refused", first ask some basic questions like "Do I have an account?" and "Is the remote computer up?" If you believe the answers to such basic questions are "yes", then it is time to see if something is wrong with the local network, the internet in between, or the remote network.

Dig, **nslookup**, **ping**, and **traceroute** are applications that can help diagnose such problems. Domain Information Grouper (**dig**) and name server lookup (**nslookup**) can be used to discover Domain Name System (DNS) information about computers and their IP addresses. **Ping** can be used to determine whether a path exists to a remote computer. **Traceroute** (called **tracert** in Microsoft Windows©) shows the network nodes along the path.

In DNS terminology, there are many records types. The four most commonly used ones are:

A records – map domain names to IPv4 addresses,
AAAA records – map domain names to IPv6 addresses ("AAAA" is called "quad A"),
CNAME records – map aliases or nicknames to "real" domain names, and
PTR records – map IP addresses to domain names (in a convoluted manner).

There are more records types, which this article does not attempt to discuss.

Example **dig** command lines are shown below. The **dig** command will query a DNS server and show the results. To access the built-in help function in **dig**, do this:

% dig -help

In the examples below, only the ANSWER SECTION of **dig** output is shown.

Cautions: **Dig** is newer than **nslookup**, and is beginning to replace it. **Dig** is not usually found on Microsoft Windows computers. **Dig** output is quite verbose.

Command line and resulting output	Notes
% dig ftp.microsoft.com ;; ANSWER SECTION: ftp.microsoft.com. 3265 IN CNAME ftp.microsoft.akadns.net. ftp.microsoft.akadns.net. 300 IN A 64.4.30.34	A domain name was used. Since no type was given after the domain name, dig will look for an “A” record. ftp.microsoft.com is an alias of ftp.microsoft.akadns.net which has an IPv4 address of 64.4.30.34
% dig ipv6.google.com AAAA ;; ANSWER SECTION: ipv6.google.com. 3705 IN CNAME ipv6.l.google.com. ipv6.l.google.com. 300 IN AAAA 2001:4860:8010::63	A domain name was used. Since the “AAAA” type was given after the domain name, dig will look for a “AAAA” record. ipv6.google.com is an alias of ipv6.l.google.com, which has an IPv6 address of 2001:4860:8010::63
% dig ipv6.google.com A ;; ANSWER SECTION: ipv6.google.com. 3251 IN CNAME ipv6.l.google.com. % dig ipv6.l.google.com ANY ;; ANSWER SECTION: ipv6.l.google.com. 300 IN AAAA 2001:4860:8010::68	A domain name was used. dig looked for an “A” record but did not find one. ipv6.google.com is an alias for ipv6.l.google.com, which a dig with a type of ANY shows has an IPv6 address of 2001:4860:8010::68
% dig -x 140.32.246.9 ;; ANSWER SECTION: 9.246.32.140.in-addr.arpa. 3447 IN PTR www.hpcmo.hpc.mil.	An IPv4 address was used with the “-x” option. 140.32.246.9 has a domain name of www.hpcmo.hpc.mil
% dig -x 64.4.30.34 (no ANSWER SECTION returned by dig)	An IPv4 address was used. When no ANSWER SECTION is returned, it means that there is no PTR record for that IP address on the default DNS server
% dig -x 2001:4860:8010::93 ;; ANSWER SECTION: 3.9.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.8.0.6.8.4.1.0.0.2.ip6.arpa. 86358IN PTR ipv6.google.com.	An IPv6 address was used with the “-x” option. 2001:4860:8010::93 has a domain name of ipv6.google.com

Example **nslookup** command lines are shown below. The **nslookup** command will query a DNS server and show the results. In order to access the built-in help function in **nslookup**:

```
C:\>nslookup
> help
```

and then use the **exit** command to leave the program. In the examples below, only selected lines of **nslookup** output are shown. If **nslookup** is not installed, try using **dig** instead.

Cautions: The “-q=PTR” option sometimes doesn’t work under Microsoft Windows.

Command line and resulting output	Notes
% nslookup ftp.microsoft.com Non-authoritative answer: Name: ftp.microsoft.akadns.net Address: 64.4.30.34	A domain name was used. Since no “-q=” option was given before the domain name, nslookup will look for an “A” record. ftp.microsoft.com is an alias of ftp.microsoft.akadns.net, which has an IPv4 address of 64.4.30.34
C:\>nslookup -q=AAAA ipv6.google.com ipv6.google.com canonical name = ipv6.l.google.com ipv6.l.google.com AAAA IPv6 address = 2001:4860:800f::68	A domain name was used. Since the “-q=AAAA” option was given, nslookup will look for a “AAAA” record. ipv6.google.com is an alias for ipv6.l.google.com, which has an IPv6 address of 2001:4860:8010::68
% nslookup -q=A ipv6.google.com Non-authoritative answer: ipv6.google.com canonical name = ipv6.l.google.com.	A domain name was used. nslookup looked for an “A” record but did not find one, either for ipv6.google.com or for an alias of ipv6.l.google.com
C:\>nslookup -q=PTR 140.32.246.9 9.246.32.140.in-addr.arpa name = www.hpcmo.hpc.mil	An IPv4 address was used with the “-q=PTR” option. 140.32.246.9 has a domain name of www.hpcmo.hpc.mil
C:\>nslookup -q=PTR 64.4.30.34 DNS request timed out. timeout was 2 seconds. *** Can't find server name for address 64.4.30.34: Timed out	An IPv4 address was used. The local DNS server timed out while trying to get the answer from another DNS server. Try again
% nslookup -q=PTR 64.4.30.34 ** server can't find 34.30.4.64.in-addr.arpa.: NXDOMAIN	An IPv4 address was used with the “-q=PTR” option. When a “can’t find” message is returned, it means that there is no PTR record for that IP address on the default DNS server.
% nslookup -q=PTR 2001:4860:8010::93 3.9.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.8.0.6.8.4.1.0.0.2.ip6.arpa name = ipv6.google.net.	An IPv6 address was used with the “-q=PTR” option. 2001:4860:8010::93 has a domain name of ipv6.google.com

Example **ping** command lines are shown below. The **ping** command will try to send packets to the remote computer and ask for a reply from the remote computer when each packet is received.

Cautions: Sometimes the name of the **ping** command that understands IPv6 may be **ping -6** or **ping6**. On some computers, a default number of packets will be sent, otherwise **ping** will keep sending packets until the command is terminated (usually by entering a Ctrl-C in the command window). A specified number of packets may be sent as shown in the examples. The **ping** command is not always on the default path. Try using the following command to locate the correct path:

% whereis ping

Command line and resulting output	Notes
C:\>ping -n 2 ftp.microsoft.com Pinging ftp.microsoft.com [64.4.30.34] with 32 bytes of data: Reply from 64.4.30.34: bytes=32 time=97ms TTL=51 Reply from 64.4.30.34: bytes=32 time=97ms TTL=51 Ping statistics for 64.4.30.34: Packets: Sent = 2, Received = 2, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 97ms, Maximum = 97ms, Average = 97ms	A domain name was used. The "-n 2" requests that 2 packets be sent. On some computers, the option to request a specific number of packets is "-c" instead of "-n".
% /usr/sbin/ping ftp.microsoft.com –s 24 2 PING www.google.com: 24 data bytes 32 bytes from ftp.microsoft.com (64.4.30.34): icmp_seq=0. time=49.6 ms 32 bytes from ftp.microsoft.com (64.4.30.34): icmp_seq=1. time=47.3 ms ----ftp.microsoft.com PING Statistics---- 2 packets transmitted, 2 packets received, 0% packet loss round-trip (ms) min/avg/max/stddev = 47.2/48.0/49.6/1.4	When the **ping** command is not in the default path, use an explicit path. In this case, the "-s 24" option requests 24 bytes, and the "2" option requests 2 packets
% ping -c 2 ftp.microsoft.com PING 64.4.30.34 (64.4.30.34) 56(84) bytes of data. 64 bytes from 64.4.30.34: icmp_seq=1 ttl=59 time=112 ms 64 bytes from 64.4.30.34: icmp_seq=2 ttl=59 time=111 ms --- 64.4.30.34 ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 1003ms rtt min/avg/max/mdev = 111.856/112.126/112.520/0.480 ms	An IPv4 address was used
% ping -c 2 ipv6.google.com ping: unknown host ipv6.google.com % ping6 -c 2 ipv6.google.com PING ipv6.google.com(2001:4860:800f::67) 56 data bytes 64 bytes from 2001:4860:800f::67: icmp_seq=1 ttl=57 time=66.8 ms 64 bytes from 2001:4860:800f::67: icmp_seq=2 ttl=57 time=67.6 ms --- ipv6.google.com ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 2003ms rtt min/avg/max/mdev = 66.843/67.404/67.688/0.449 ms	An IPv6 address was used. Since **ping** did not understand IPv6, the **ping6** command was used instead. The "-c 2" requests that 2 packets be sent.
C:\>ping -6 ftp.microsoft.com Ping request could not find host ftp.microsoft.com. Please check the name. % ping6 ftp.microsoft.com unknown host % /usr/sbin/ping -s -A inet6 ftp.microsoft.com unknown host ftp.microsfot.com	The remote computer may not understand IPv6
C:\>ping -n 2 ftp.microsoft.com Pinging ftp.microsoft.com [64.4.30.34] with 32 bytes of data: Destination host unreachable. Destination host unreachable. Ping statistics for 64.4.30.34: Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),	A network device between the local and remote computers does not understand IPv4, or is not working. The **traceroute** command may be able to identify the device
% ping –A inet4 ftp.microsoft.com	Ping on Solaris will use IPv4
% ping –A inet6 ftp.microsoft.com	Ping on Solaris will use IPv6
% ping6 -c 2 ipv6.google.com PING ipv6.google.com(2001:4860:800f::67) 56 data bytes Destination unreachable: Address unreachable. Destination unreachable: Address unreachable. --- ipv6.google.com ping statistics --- 2 packets transmitted, 0 received, 100% packet loss, time 1003ms	A network device between the local and remote computers does not understand IPv6, or is not working. The **traceroute** command may be able to identify the device
%ping –i eth0 fe80::21b:63ff:feab:e6a6 PING fe80::21b:63ff:feab:e6a6%eth0(fe80::21b:63ff:feab:e6a6) 56 data bytes 64 bytes from fe80::21b:63ff:feab:e6a6: icmp_seq=1 ttl=64 time=0.041 ms	For a link local address (fe80), an interface name or address must be specified (-i eth0)

Example **traceroute** command lines are shown below. The **traceroute** command will try to send packets along the path to the remote computer and show the reply from each network device along the path. The **traceroute** command usually displays a lot of output. In the examples below, only the first and last few lines have been shown.

Cautions: On Microsoft Windows computers, the traceroute command is called **tracert**. On other computers, the name of the **traceroute** command that understands IPv6 may not be **traceroute**. If **traceroute** doesn't work, try using **traceroute -6** or **traceroute6** instead. The **traceroute** command is not always on the default path. Try using the following command to locate the correct path:

% whereis traceroute

Command line and resulting output	Notes
C:\>tracert ftp.microsoft.com Tracing route to ftp.microsoft.com [64.4.30.34] over a maximum of 30 hops: 1 95 ms 97 ms 97 ms 140.32.241.128 2 97 ms 95 ms 96 ms jupiter.hpcmo.hpc.mil [140.32.240.1] … 10 103 ms 98 ms 97 ms ftp.microsoft.com [64.4.30.34] Trace complete.	A domain name was used
% /usr/sbin/traceroute ftp.microsoft.com traceroute to ftp.microsoft.com (64.4.30.62), 30 hops max, 40 byte packets 1 mana-gw.mhpcc.hpc.mil (140.31.196.225) 0.463 ms 0.519 ms 0.583 ms 2 border.mhpcc.hpc.mil (140.31.192.33) 0.354 ms 0.660 ms 0.695 ms … 10 ftp.microsoft.net (64.4.30.62) 76.112 ms 76.056 ms 75.956 ms	A domain name was used. When the **traceroute** command is not in the default path, an explicit path must used
C:\>tracert 64.4.30.34 Tracing route to 64.4.30.34 [64.4.30.34] over a maximum of 30 hops: 1 95 ms 97 ms 97 ms 140.32.241.128 2 97 ms 95 ms 96 ms jupiter.hpcmo.hpc.mil [140.32.240.1] … 10 103 ms 98 ms 97 ms ftp.microsoft.com [64.4.30.34] Trace complete.	An IPv4 address was used
% traceroute -6 ipv6.google.com traceroute to ipv6.google.com (2001:4860:8010::68), 30 hops max, 40 byte packets 1 2001:480:120:196c::1 (2001:480:120:196c::1) 0.529 ms 0.558 ms 0.629 ms 2 2001:480:120:1::1 (2001:480:120:1::1) 0.500 ms 0.572 ms 0.646 ms … 13 ipv6.google.com (2001:4860:8010::68) 89.029 ms 86.933 ms 86.914 ms	An IPv6 address was used. Since **traceroute** did not understand IPv6, the **traceroute -6** command was used instead
% traceroute -6 ftp.microsoft.com ftp.microsoft.com: No address associated with hostname C:\> tracert -6 ftp.microsoft.com Unable to resolve target system name ftp.microsoft.com.	The remote computer may not understand IPv6
% traceroute –A inet4 ftp.microsoft.com	traceroute on Solaris will use IPv4
% traceroute –A inet6 ftp.microsoft.com	traceroute on Solaris will use IPv6
C:/>tracert ftp.microsoft.com Tracing route to 64.4.30.34 [64.4.30.34] over a maximum of 30 hops: 1 95 ms 97 ms 97 ms 140.32.241.128 2 97 ms 95 ms 96 ms jupiter.hpcmo.hpc.mil [140.32.240.1] … Destination host unreachable.	A network device between the local and remote computers does not understand IPv4, or is not working. It is the next device on the path after hop 2
% traceroute ipv6.google.com traceroute to ipv6.google.com (2001:4860:8010::68), 30 hops max, 40 byte packets 1 2001:480:120:196c::1 (2001:480:120:196c::1) 0.529 ms 0.558 ms 0.629 ms 2 2001:480:120:1::1 (2001:480:120:1::1) 0.500 ms 0.572 ms 0.646 ms … Destination unreachable: Address unreachable.	A network device between the local and remote computers does not understand IPv6, or is not working. It is the next device on the path after hop 2
C:\>tracert -d fe80::8009:ece6:e719:c8d%11 Tracing route to fe80::8009:ece6:e719:c8d%11 over a maximum of 30 hops 1 <1 ms <1 ms <1 ms fe80::8009:ece6:e719:c8d Trace complete.	For a link local address (fe80), a zone ID such as "%11" (interface index) must be appended on a computer with multiple network interfaces