Lab Assignment & Solution

Cybersecurity Professional Program

Introductory Course

Network Fundamentals

IC-04-LS3
Ping and Traceroute Practice

Note: Solutions for the instructor are shown inside the green box.

Lab Objective

Gain a better understanding of the ***ping*** and ***traceroute*** commands for troubleshooting purposes. In this lab, you will use ***ping***, ***traceroute***, and ***pathping*** (Microsoft Windows only) to learn more about how data moves through the internet.

Lab Mission

Use the ***ping*** and ***traceroute*** commands in multiple scenarios.
Note: Task 1 is for Windows OS users; Task 2 is for Mac OS X users.

Lab Duration

10–15 minutes

Requirements

- Basic knowledge of the ***ping*** and ***traceroute*** commands

Resources

- Environment & Tools
  - Windows OS
    - Command Prompt
  - Mac OS X
    - Terminal

Textbook References

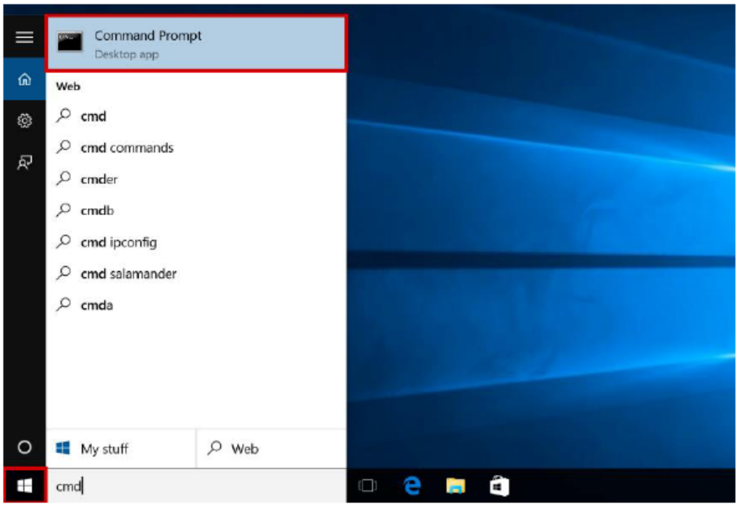
- Chapter 3: Network Fundamentals
  - Section 4: Protocols and Communication

Lab Task 1: Windows 10 Ping and Traceroute Execution

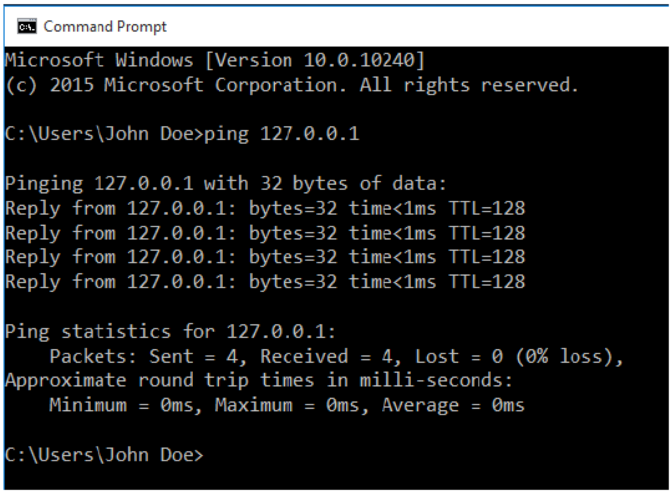
Interview Tip
A popular question in an interview is, “How would you check if a device is up?” Your answer should be ***ping***.

This lab will help you practice using ***ping*** and some of its options.

1 Press the ***Windows*** key, type **cmd**, and select **Command Prompt**.



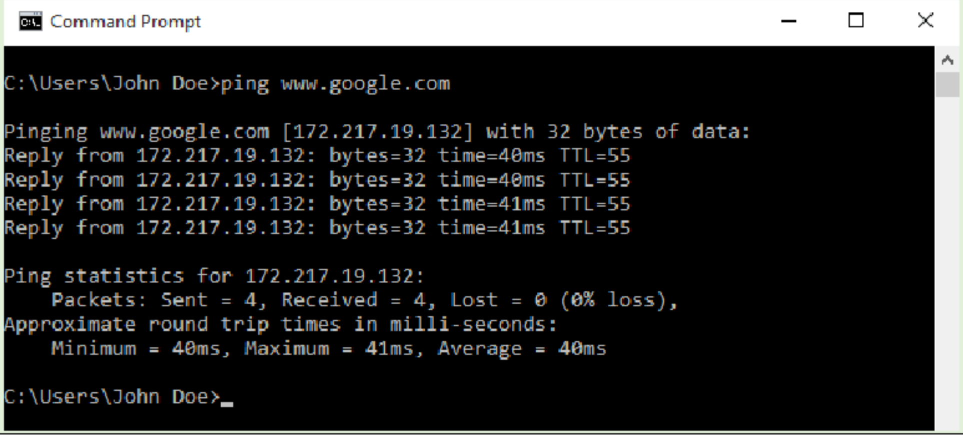
2 In the command prompt, enter **ping 127.0.0.1**. This will check that your NIC works properly.



```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\John Doe>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

C:\Users\John Doe>
```

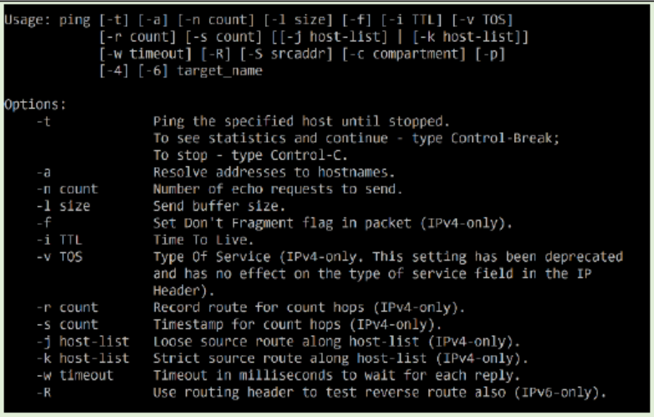
3 In the command prompt, enter **ping www.google.com**
Why is it possible to ping ***google.com***?

It is possible to ping the domain name because we are using a DNS server that translates domain names to IP addresses.



```
C:\Users\John Doe>ping www.google.com

Pinging www.google.com [172.217.19.132] with 32 bytes of data:
Reply from 172.217.19.132: bytes=32 time=40ms TTL=55
Reply from 172.217.19.132: bytes=32 time=40ms TTL=55
Reply from 172.217.19.132: bytes=32 time=41ms TTL=55
Reply from 172.217.19.132: bytes=32 time=41ms TTL=55

Ping statistics for 172.217.19.132:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 40ms, Maximum = 41ms, Average = 40ms

C:\Users\John Doe>_
```

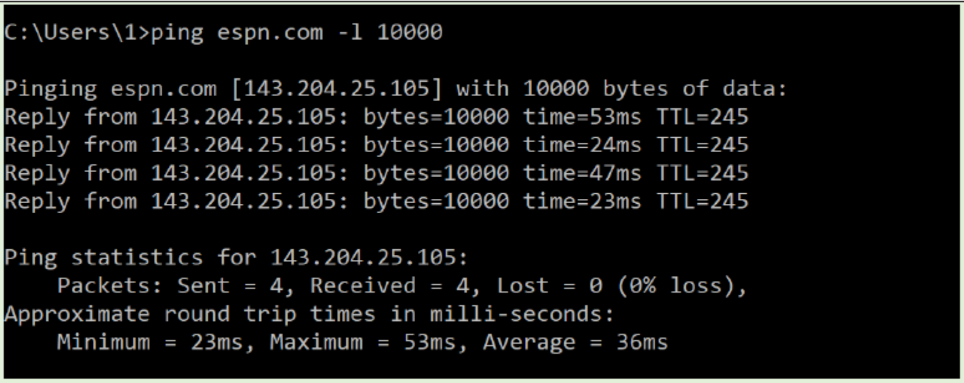
4 Run ***ping /?*** to find the syntax of the command. The syntax shows you how to write the command and what options are available for it.



```
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).
```

Tip
You can learn about almost any Microsoft Windows command by simply typing ***/?*** after the command.

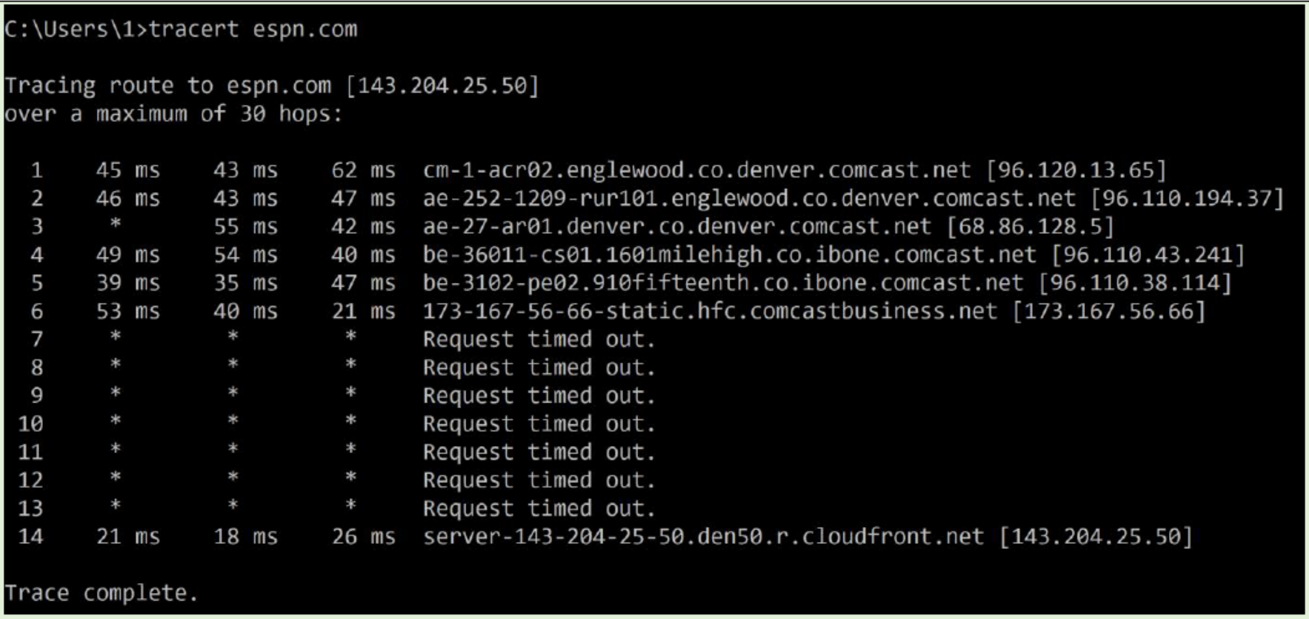
5 Ping packets are small and may not detect any problems with larger traffic like streaming video. To send a larger ping packet, use the ***-l*** option and a value from 0–65,535. For example, send a larger packet by typing **ping espn.com -l 10000**



```
C:\Users\1>ping espn.com -l 10000

Pinging espn.com [143.204.25.105] with 10000 bytes of data:
Reply from 143.204.25.105: bytes=10000 time=53ms TTL=245
Reply from 143.204.25.105: bytes=10000 time=24ms TTL=245
Reply from 143.204.25.105: bytes=10000 time=47ms TTL=245
Reply from 143.204.25.105: bytes=10000 time=23ms TTL=245

Ping statistics for 143.204.25.105:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 23ms, Maximum = 53ms, Average = 36ms
```

6 The ping utility is basic, so you can use the traceroute utility to show a path to the destination. In Microsoft Windows, the command is ***tracert***. Use the ***tracert*** command to traceroute ***espn.com***.

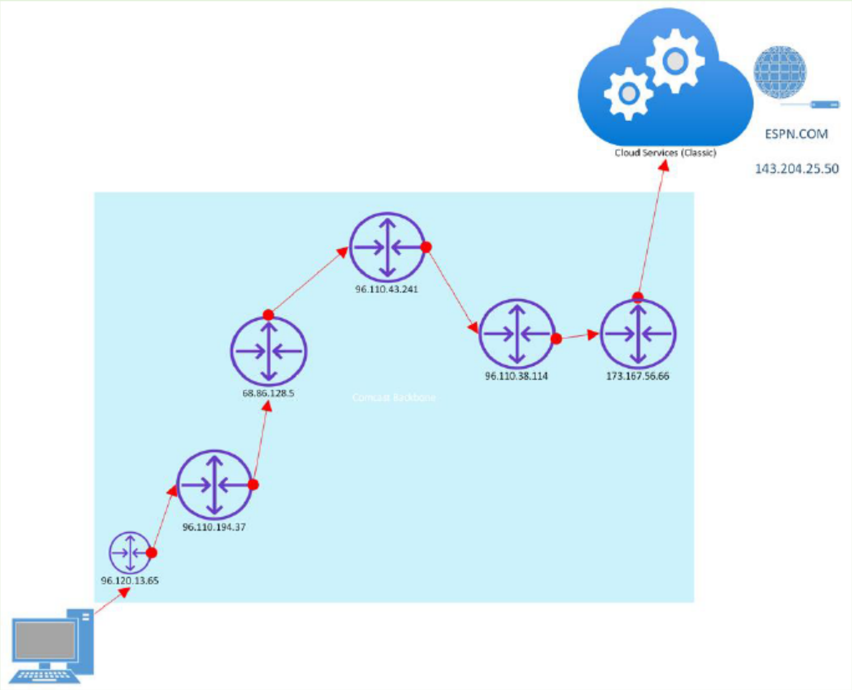
Good to Know
Microsoft uses the command ***tracert*** versus ***traceroute*** to maintain the original 8.3 naming convention from older OSs like Windows 95.



```
C:\Users\1>tracert espn.com

Tracing route to espn.com [143.204.25.50]
over a maximum of 30 hops:

  1    45 ms    43 ms    62 ms  cm-1-acr02.englewood.co.denver.comcast.net [96.120.13.65]
  2    46 ms    43 ms    47 ms  ae-252-1209-rur101.englewood.co.denver.comcast.net [96.110.194.37]
  3     *       55 ms    42 ms  ae-27-ar01.denver.co.denver.comcast.net [68.86.128.5]
  4    49 ms    54 ms    40 ms  be-36011-cs01.1601milehigh.co.ibone.comcast.net [96.110.43.241]
  5    39 ms    35 ms    47 ms  be-3102-pe02.910fifteenth.co.ibone.comcast.net [96.110.38.114]
  6    53 ms    40 ms    21 ms  173-167-56-66-static.hfc.comcastbusiness.net [173.167.56.66]
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12     *        *        *     Request timed out.
 13     *        *        *     Request timed out.
 14    21 ms    18 ms    26 ms  server-143-204-25-50.den50.r.cloudfront.net [143.204.25.50]

Trace complete.
```

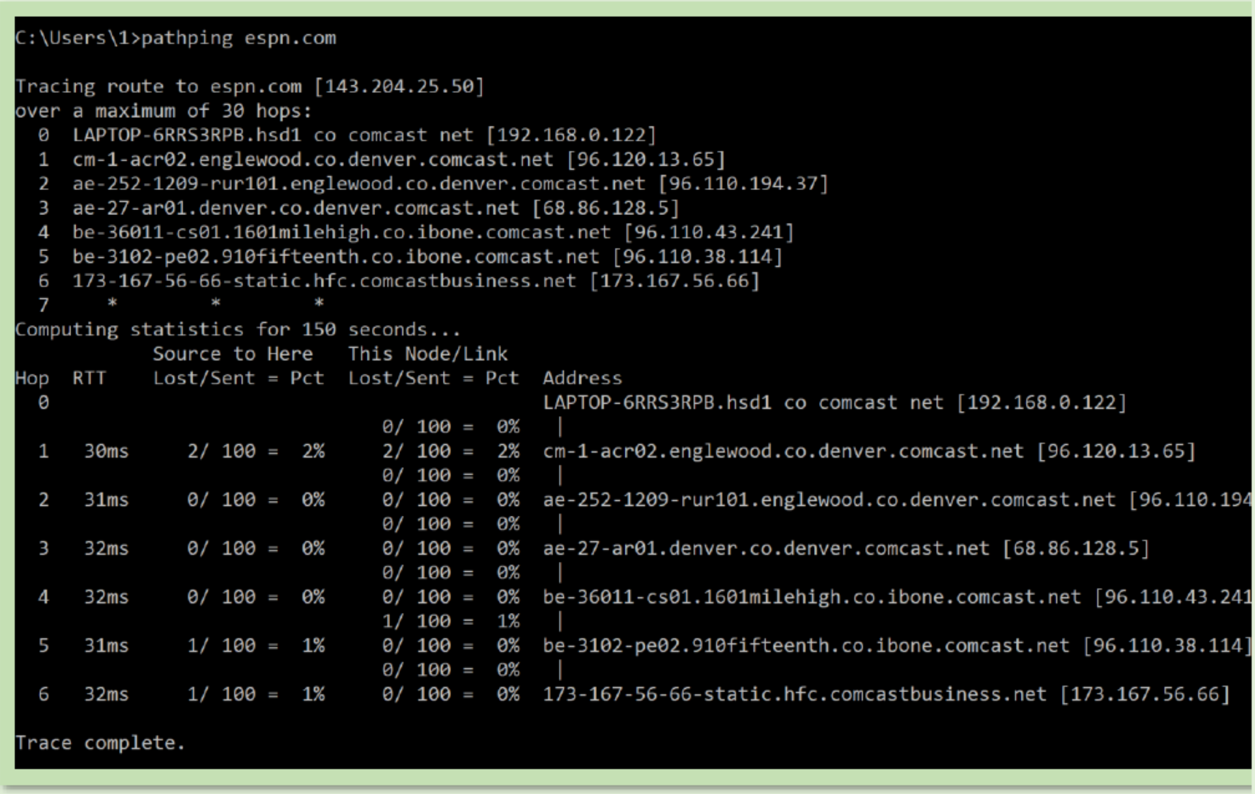
In this example, hops 7–13 do not show up because the network administrator has disabled that feature.

If the trace completely times out, it could indicate a routing problem.

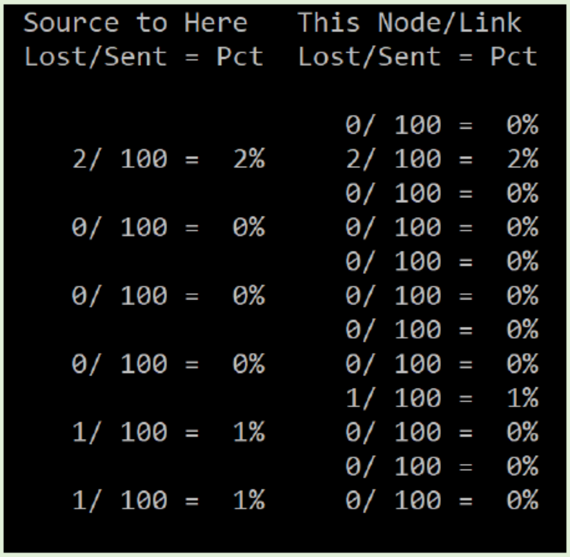


This diagram is a visual representation of a traceroute from above.

7 Even if the packet does get through, there could be some loss. Microsoft Windows does have the utility to measure packet loss. In the command prompt, type **pathping espn.com**



```
C:\Users\1>pathping espn.com

Tracing route to espn.com [143.204.25.50]
over a maximum of 30 hops:
  0  LAPTOP-6RRS3RPB.hsd1 co comcast net [192.168.0.122]
  1  cm-1-acr02.englewood.co.denver.comcast.net [96.120.13.65]
  2  ae-252-1209-rur101.englewood.co.denver.comcast.net [96.110.194.37]
  3  ae-27-ar01.denver.co.denver.comcast.net [68.86.128.5]
  4  be-36011-cs01.1601milehigh.co.ibone.comcast.net [96.110.43.241]
  5  be-3102-pe02.910fifteenth.co.ibone.comcast.net [96.110.38.114]
  6  173-167-56-66-static.hfc.comcastbusiness.net [173.167.56.66]
  7     *        *        *
Computing statistics for 150 seconds...
            Source to Here   This Node/Link
Hop  RTT    Lost/Sent = Pct  Lost/Sent = Pct  Address
  0                                           LAPTOP-6RRS3RPB.hsd1 co comcast net [192.168.0.122]
                                0/ 100 =  0%   |
  1   30ms     2/ 100 =  2%     2/ 100 =  2%  cm-1-acr02.englewood.co.denver.comcast.net [96.120.13.65]
                                0/ 100 =  0%   |
  2   31ms     0/ 100 =  0%     0/ 100 =  0%  ae-252-1209-rur101.englewood.co.denver.comcast.net [96.110.194
                                0/ 100 =  0%   |
  3   32ms     0/ 100 =  0%     0/ 100 =  0%  ae-27-ar01.denver.co.denver.comcast.net [68.86.128.5]
                                0/ 100 =  0%   |
  4   32ms     0/ 100 =  0%     0/ 100 =  0%  be-36011-cs01.1601milehigh.co.ibone.comcast.net [96.110.43.241
                                1/ 100 =  1%   |
  5   31ms     1/ 100 =  1%     0/ 100 =  0%  be-3102-pe02.910fifteenth.co.ibone.comcast.net [96.110.38.114]
                                0/ 100 =  0%   |
  6   32ms     1/ 100 =  1%     0/ 100 =  0%  173-167-56-66-static.hfc.comcastbusiness.net [173.167.56.66]

Trace complete.
```



```
Source to Here   This Node/Link
Lost/Sent = Pct  Lost/Sent = Pct

                    0/ 100 =  0%
   2/ 100 =  2%     2/ 100 =  2%
                    0/ 100 =  0%
   0/ 100 =  0%     0/ 100 =  0%
                    0/ 100 =  0%
   0/ 100 =  0%     0/ 100 =  0%
                    0/ 100 =  0%
   0/ 100 =  0%     0/ 100 =  0%
                    1/ 100 =  1%
   1/ 100 =  1%     0/ 100 =  0%
                    0/ 100 =  0%
   1/ 100 =  1%     0/ 100 =  0%
```

In this example, you can see that there is some packet loss. If the percentages were higher, it could attribute to lag in videos or transactions.

Type **exit** to exit the command prompt.

Mission Completed

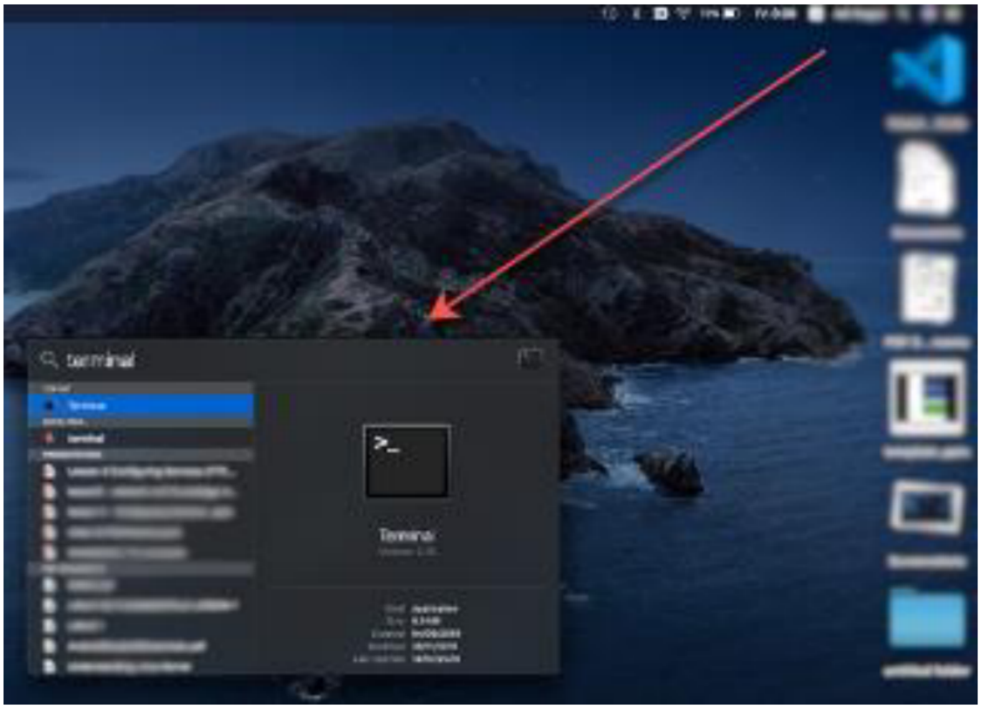
In this lab, you have accomplished the following tasks:

- Verified that the network adapter is working, and that DNS is integrated with network commands by pinging ***google.com***
- Used the traceroute command to see the path a packet takes to ***espn.com***
- Discovered that there was a 4% packet loss to ***espn.com*** using the path ping utility

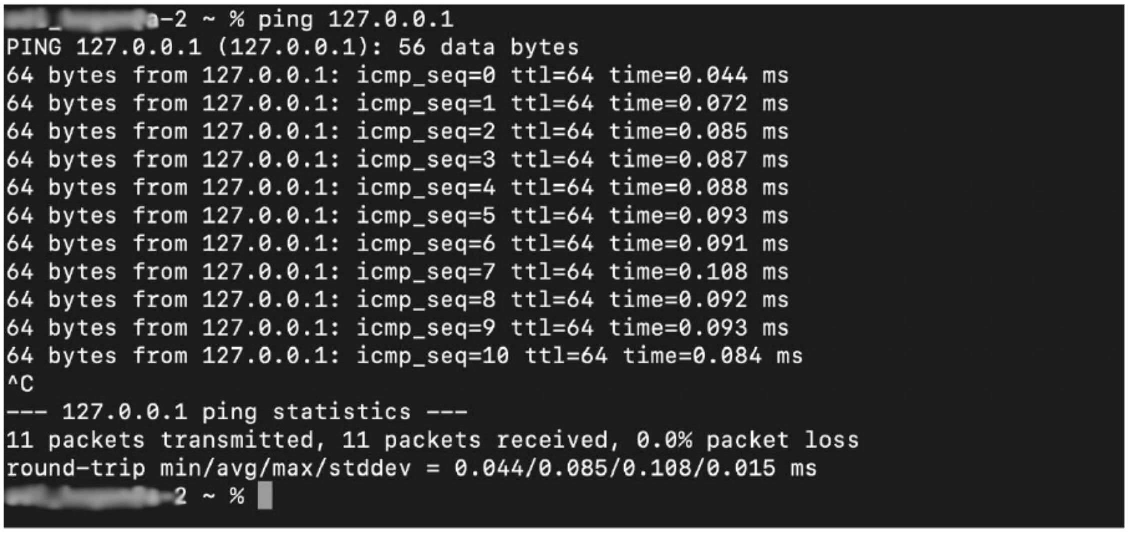
Lab Task 2: Mac OS X Ping Execution

This lab will help you practice using ***ping*** and some of its options.

1 Press ***Command+Spacebar*** to open the spotlight search and type in **terminal**. Alternatively, you can bring up the spotlight by clicking on the magnifying glass in the top right corner.



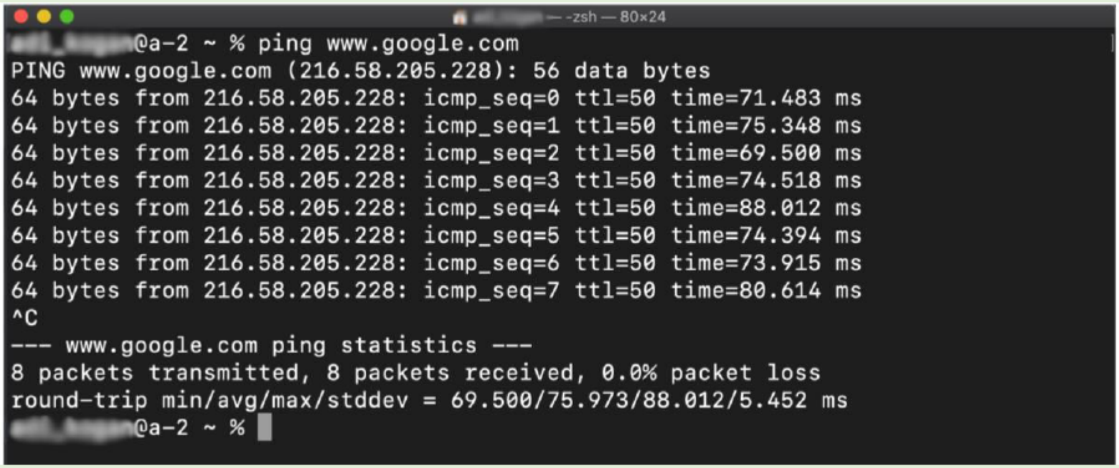
2 In the terminal, type **ping 127.0.0.1**. This will check that your NIC works properly. Note that, unlike Microsoft Windows, the ping does not stop. To stop the ping, use ***Control+C***.



```
[redacted]a-2 ~ % ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.072 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.085 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.087 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.088 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.093 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.091 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.108 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.092 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.093 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.084 ms
^C
--- 127.0.0.1 ping statistics ---
11 packets transmitted, 11 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.044/0.085/0.108/0.015 ms
[redacted]-2 ~ %
```

3 In the terminal, run ***ping www.google.com***
Why is it possible to ping [google.com]?

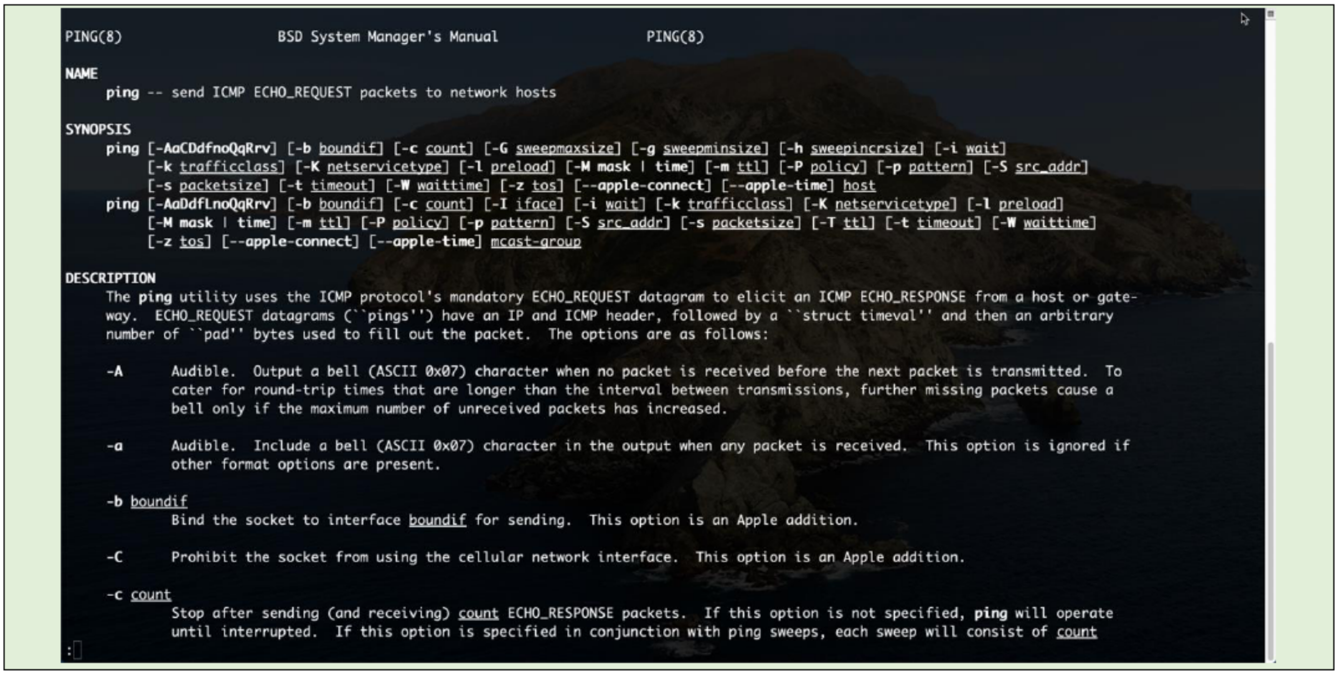
It is possible to ping the domain name because we are using a DNS server that translates domain names to IP addresses.



```
[redacted]@a-2 ~ % ping www.google.com
PING www.google.com (216.58.205.228): 56 data bytes
64 bytes from 216.58.205.228: icmp_seq=0 ttl=50 time=71.483 ms
64 bytes from 216.58.205.228: icmp_seq=1 ttl=50 time=75.348 ms
64 bytes from 216.58.205.228: icmp_seq=2 ttl=50 time=69.500 ms
64 bytes from 216.58.205.228: icmp_seq=3 ttl=50 time=74.518 ms
64 bytes from 216.58.205.228: icmp_seq=4 ttl=50 time=88.012 ms
64 bytes from 216.58.205.228: icmp_seq=5 ttl=50 time=74.394 ms
64 bytes from 216.58.205.228: icmp_seq=6 ttl=50 time=73.915 ms
64 bytes from 216.58.205.228: icmp_seq=7 ttl=50 time=80.614 ms
^C
--- www.google.com ping statistics ---
8 packets transmitted, 8 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 69.500/75.973/88.012/5.452 ms
[redacted]@a-2 ~ %
```

4 Learn more about the ***ping*** command by reviewing the manual. Type ***man ping*** to bring up the manual.

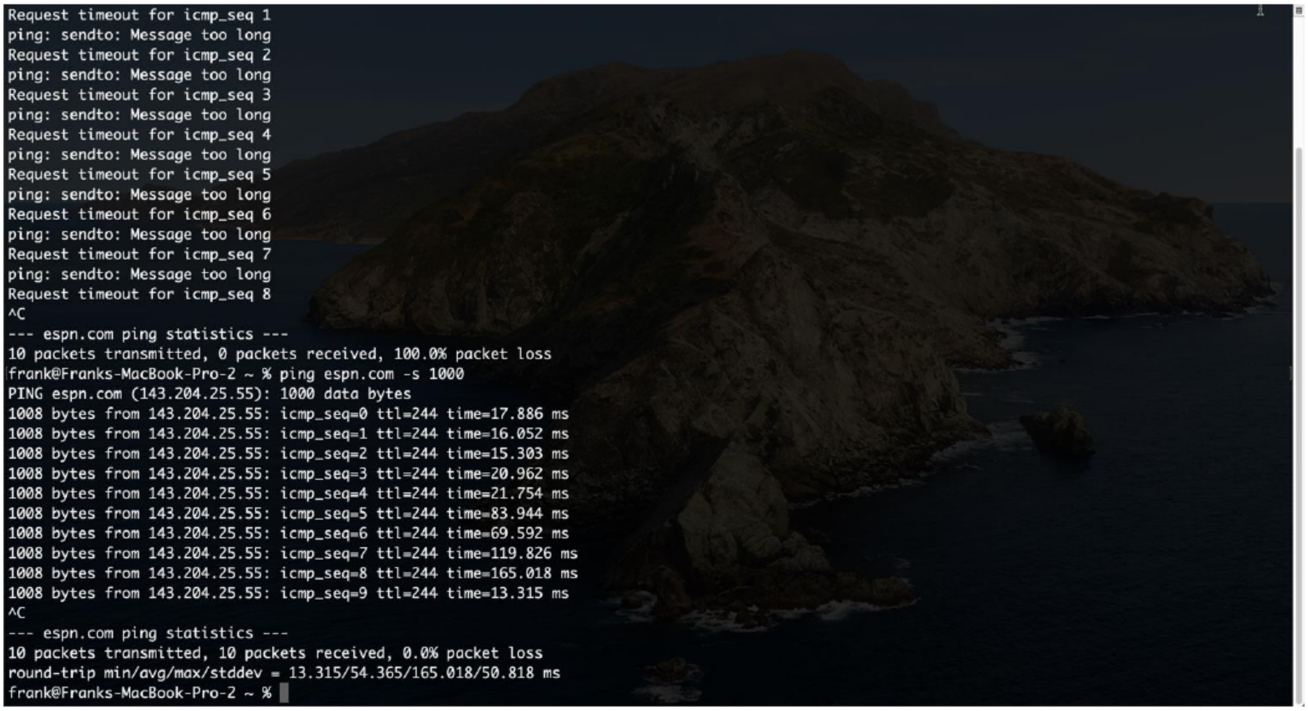
> **Tip**
> You can learn about almost any terminal command by simply typing **man** before the command. Note that in the Mac OS X terminal, there are case-sensitive options. As you can see below, there is a capital **C** option for cellular but a lowercase **c** for count.



```
PING(8)                  BSD System Manager's Manual                  PING(8)

NAME
     ping -- send ICMP ECHO_REQUEST packets to network hosts

SYNOPSIS
     ping [-AaCDdfnoQqRrv] [-b boundif] [-c count] [-G sweepmaxsize] [-g sweepminsize] [-h sweepincrsize] [-i wait]
          [-k trafficclass] [-K netservicetype] [-l preload] [-M mask | time] [-m ttl] [-P policy] [-p pattern] [-S src_addr]
          [-s packetsize] [-t timeout] [-W waittime] [-z tos] [--apple-connect] [--apple-time] host
     ping [-AaDdfLnoQqRrv] [-b boundif] [-c count] [-I iface] [-i wait] [-k trafficclass] [-K netservicetype] [-l preload]
          [-M mask | time] [-m ttl] [-P policy] [-p pattern] [-S src_addr] [-s packetsize] [-T ttl] [-t timeout] [-W waittime]
          [-z tos] [--apple-connect] [--apple-time] mcast-group

DESCRIPTION
     The ping utility uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gate-
     way.  ECHO_REQUEST datagrams (``pings'') have an IP and ICMP header, followed by a ``struct timeval'' and then an arbitrary
     number of ``pad'' bytes used to fill out the packet.  The options are as follows:

     -A      Audible.  Output a bell (ASCII 0x07) character when no packet is received before the next packet is transmitted.  To
             cater for round-trip times that are longer than the interval between transmissions, further missing packets cause a
             bell only if the maximum number of unreceived packets has increased.

     -a      Audible.  Include a bell (ASCII 0x07) character in the output when any packet is received.  This option is ignored if
             other format options are present.

     -b boundif
             Bind the socket to interface boundif for sending.  This option is an Apple addition.

     -C      Prohibit the socket from using the cellular network interface.  This option is an Apple addition.

     -c count
             Stop after sending (and receiving) count ECHO_RESPONSE packets.  If this option is not specified, ping will operate
             until interrupted.  If this option is specified in conjunction with ping sweeps, each sweep will consist of count
:
```

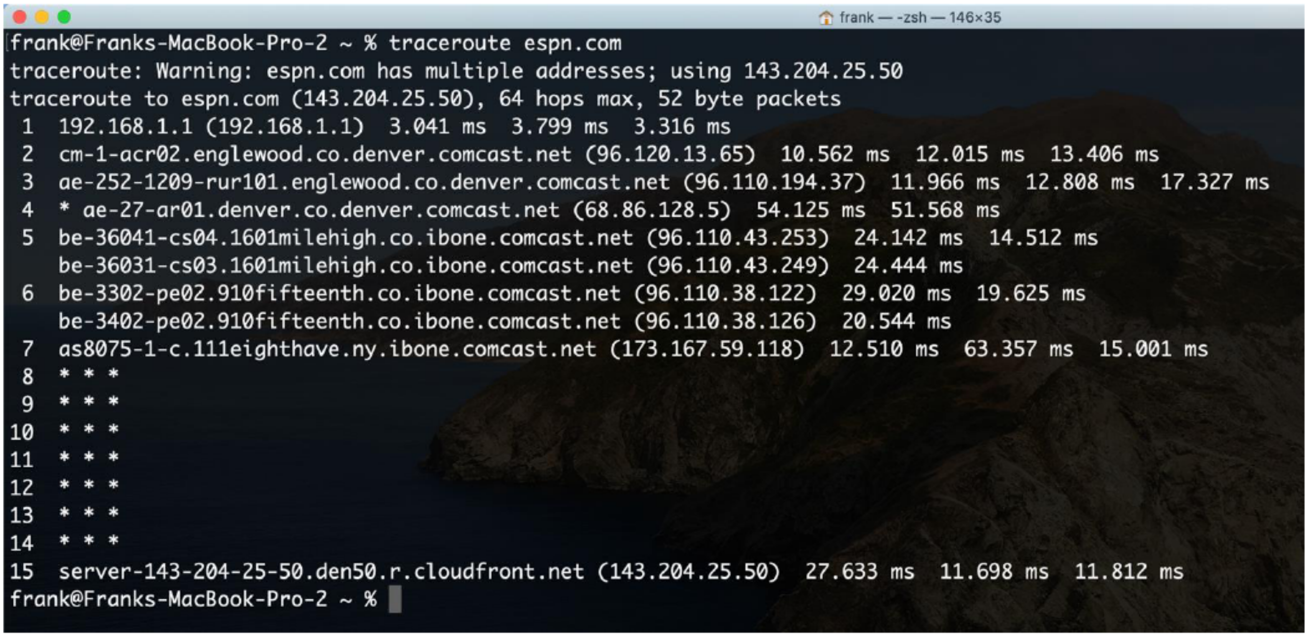
5 Use ***z*** to scroll down one page until you see ***-s packetsize***. Note what all the flags do for the ***ping*** command. Type ***q*** to exit the manual page. Ping packets are small and may not detect any problems with larger traffic like streaming video. To send a larger ping packet, type: **ping espn.com -s 10000**

You may receive an error because Mac OS X will not let you send a large ping packet. Reduce the size of the packet by typing **ping espn.com -s 1000**

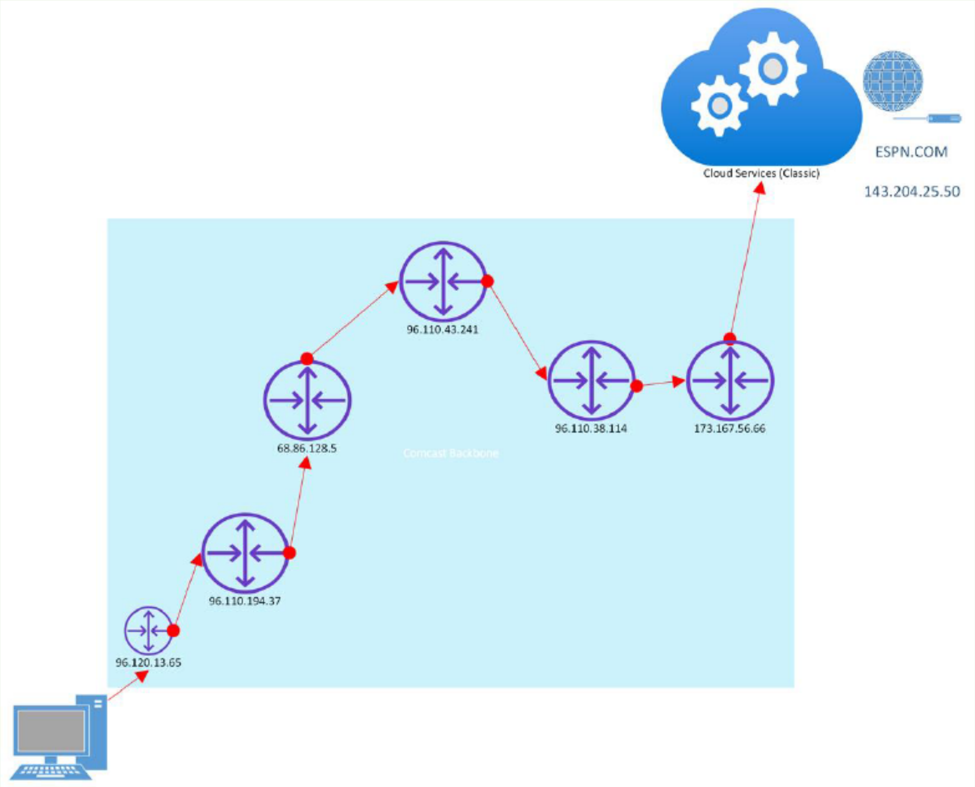


```
Request timeout for icmp_seq 1
ping: sendto: Message too long
Request timeout for icmp_seq 2
ping: sendto: Message too long
Request timeout for icmp_seq 3
ping: sendto: Message too long
Request timeout for icmp_seq 4
ping: sendto: Message too long
Request timeout for icmp_seq 5
ping: sendto: Message too long
Request timeout for icmp_seq 6
ping: sendto: Message too long
Request timeout for icmp_seq 7
ping: sendto: Message too long
Request timeout for icmp_seq 8
^C
--- espn.com ping statistics ---
10 packets transmitted, 0 packets received, 100.0% packet loss
frank@Franks-MacBook-Pro-2 ~ % ping espn.com -s 1000
PING espn.com (143.204.25.55): 1000 data bytes
1008 bytes from 143.204.25.55: icmp_seq=0 ttl=244 time=17.886 ms
1008 bytes from 143.204.25.55: icmp_seq=1 ttl=244 time=16.052 ms
1008 bytes from 143.204.25.55: icmp_seq=2 ttl=244 time=15.303 ms
1008 bytes from 143.204.25.55: icmp_seq=3 ttl=244 time=20.962 ms
1008 bytes from 143.204.25.55: icmp_seq=4 ttl=244 time=21.754 ms
1008 bytes from 143.204.25.55: icmp_seq=5 ttl=244 time=83.944 ms
1008 bytes from 143.204.25.55: icmp_seq=6 ttl=244 time=69.592 ms
1008 bytes from 143.204.25.55: icmp_seq=7 ttl=244 time=119.826 ms
1008 bytes from 143.204.25.55: icmp_seq=8 ttl=244 time=165.018 ms
1008 bytes from 143.204.25.55: icmp_seq=9 ttl=244 time=13.315 ms
^C
--- espn.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 13.315/54.365/165.018/50.818 ms
frank@Franks-MacBook-Pro-2 ~ %
```

6 The ping utility is basic, so you can use the traceroute utility to show a path to the destination. In Mac OS X, the command is spelled out with ***traceroute***.



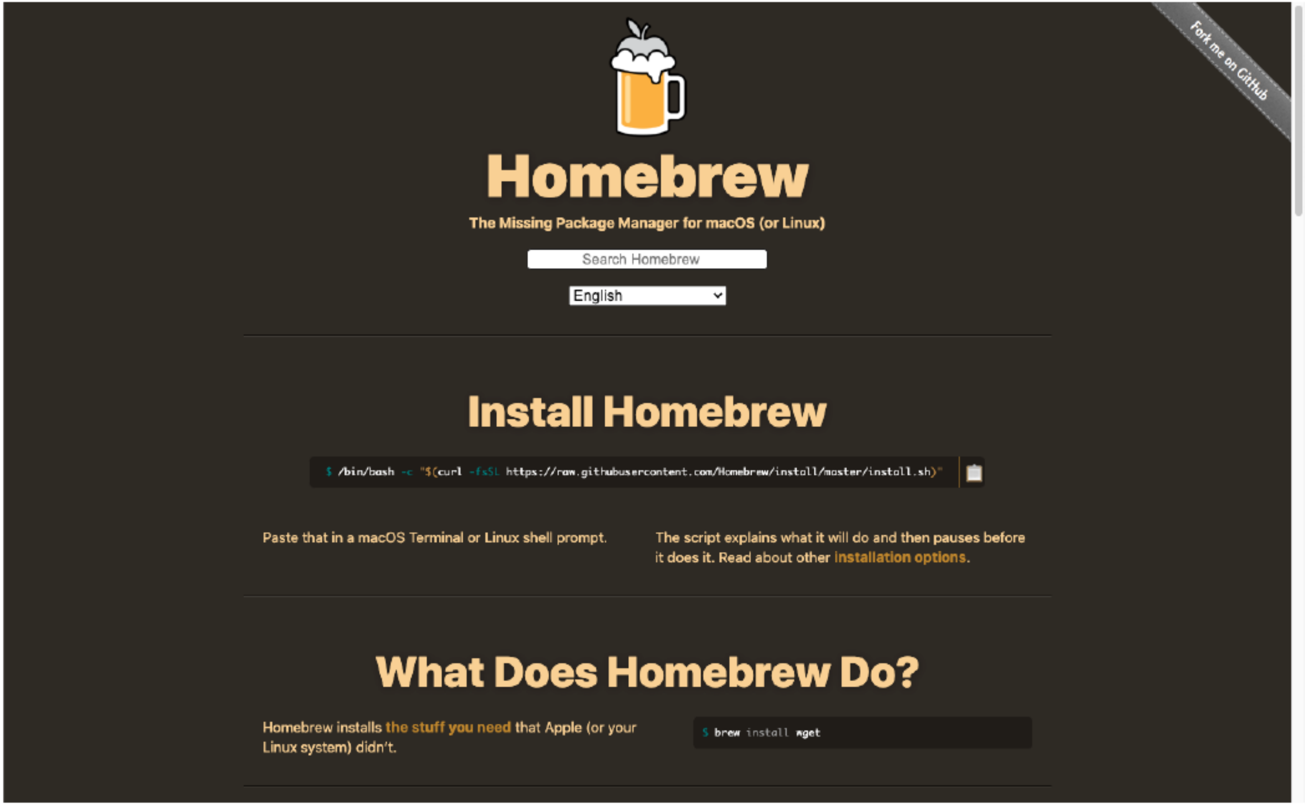
```
frank@Franks-MacBook-Pro-2 ~ % traceroute espn.com
traceroute: Warning: espn.com has multiple addresses; using 143.204.25.50
traceroute to espn.com (143.204.25.50), 64 hops max, 52 byte packets
 1  192.168.1.1 (192.168.1.1)  3.041 ms  3.799 ms  3.316 ms
 2  cm-1-acr02.englewood.co.denver.comcast.net (96.120.13.65)  10.562 ms  12.015 ms  13.406 ms
 3  ae-252-1209-rur101.englewood.co.denver.comcast.net (96.110.194.37)  11.966 ms  12.808 ms  17.327 ms
 4  * ae-27-ar01.denver.co.denver.comcast.net (68.86.128.5)  54.125 ms  51.568 ms
 5  be-36041-cs04.1601milehigh.co.ibone.comcast.net (96.110.43.253)  24.142 ms  14.512 ms
    be-36031-cs03.1601milehigh.co.ibone.comcast.net (96.110.43.249)  24.444 ms
 6  be-3302-pe02.910fifteenth.co.ibone.comcast.net (96.110.38.122)  29.020 ms  19.625 ms
    be-3402-pe02.910fifteenth.co.ibone.comcast.net (96.110.38.126)  20.544 ms
 7  as8075-1-c.111eighthave.ny.ibone.comcast.net (173.167.59.118)  12.510 ms  63.357 ms  15.001 ms
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  server-143-204-25-50.den50.r.cloudfront.net (143.204.25.50)  27.633 ms  11.698 ms  11.812 ms
frank@Franks-MacBook-Pro-2 ~ %
```



This diagram is a visual representation of a traceroute from above.

7 Mac OS X uses a Linux utility called MTR to measure packet loss. It is optional to install this, as it is not a native OS X tool.

8 Navigate to the Homebrew webpage at [*http://brew.sh*](http://brew.sh)



Homebrew

The Missing Package Manager for macOS (or Linux)

Install Homebrew

```
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

Paste that in a macOS Terminal or Linux shell prompt.

The script explains what it will do and then pauses before it does it. Read about other installation options.

What Does Homebrew Do?

Homebrew installs the stuff you need that Apple (or your Linux system) didn't.

```
$ brew install wget
```

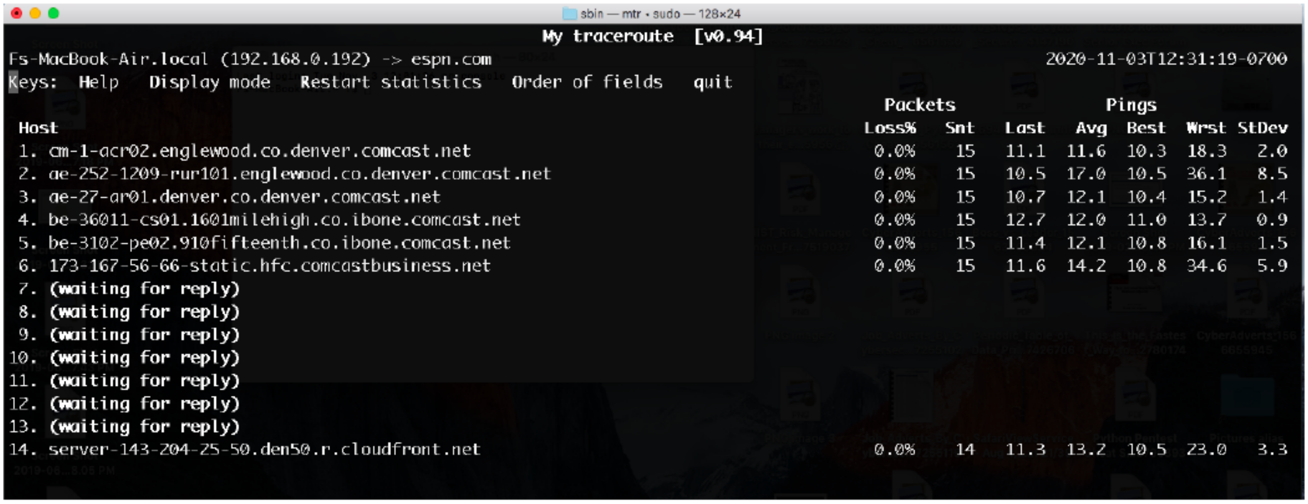
9 Per the homepage, copy and paste the following into your terminal window ***/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"***

10 The installation may take 5–15 minutes to complete. Once the installation is complete, run the command: ***brew install mtr***

11 Once ***mtr*** is installed, run the command: ***cd /usr/local/Cellar/mtr/0.92/sbin*** **Note:** Versions may have changed; therefore, if you get stuck after ***mtr***, enter a slash (***/***) and press the ***Tab*** key.

12 Make the ***mtr*** command available to your system by running ***cp mtr /usr/local/bin/***

13 To run an ***mtr*** trace, run the command: ***sudo mtr espn.com***



```
                                            My traceroute  [v0.94]
Fs-MacBook-Air.local (192.168.0.192) -> espn.com                                             2020-11-03T12:31:19-0700
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                                                        Packets               Pings
 Host                                                                 Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. cm-1-acr02.englewood.co.denver.comcast.net                         0.0%    15   11.1  11.6  10.3  18.3   2.0
 2. ae-252-1209-rur101.englewood.co.denver.comcast.net                 0.0%    15   10.5  17.0  10.5  36.1   8.5
 3. ae-27-ar01.denver.co.denver.comcast.net                            0.0%    15   10.7  12.1  10.4  15.2   1.4
 4. be-36011-cs01.1601milehigh.co.ibone.comcast.net                    0.0%    15   12.7  12.0  11.0  13.7   0.9
 5. be-3102-pe02.910fifteenth.co.ibone.comcast.net                     0.0%    15   11.4  12.1  10.8  16.1   1.5
 6. 173-167-56-66-static.hfc.comcastbusiness.net                       0.0%    15   11.6  14.2  10.8  34.6   5.9
 7. (waiting for reply)
 8. (waiting for reply)
 9. (waiting for reply)
10. (waiting for reply)
11. (waiting for reply)
12. (waiting for reply)
13. (waiting for reply)
14. server-143-204-25-50.den50.r.cloudfront.net                        0.0%    14   11.3  13.2  10.5  23.0   3.3
```

14 In this capture, you can see that there is no packet loss. MTR is a real-time utility and will continue until you tell it to stop with ***Ctrl+C***

15 Type **exit** to exit the utility, and then press ***Command+Q*** to close the terminal window.

Mission Completed

In this lab, you have accomplished the following tasks:

- Verified that the network adapter is working, and that DNS is integrated with network commands by pinging ***google.com***
- Used the traceroute command to see the path a packet takes to ***espn.com***
- *Optionally* installed and used the ***mtr*** utility to get a real-time view of data loss