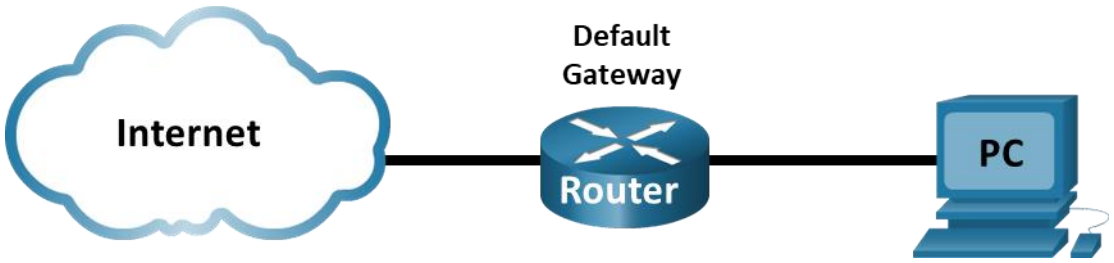
Lab - Test Network Latency with Ping and Traceroute

Topology



Objectives

Part 1: Use Ping to Document Network Latency

Part 2: Use Traceroute to Document Network Latency

Background / Scenario

To obtain realistic network latency statistics, this activity must be performed on a live network. Be sure to check with your instructor for any local security restrictions against using the **ping** command on the network.

The purpose of this lab is to measure and evaluate network latency over time and during different periods of the day to capture a representative sample of typical network activity. This will be accomplished by analyzing the return delay from a distant computer with the **ping** command. Return delay times, measured in milliseconds, will be summarized by computing the average latency (mean) and the range (maximum and minimum) of the delay times.

Required Resources

- 1 PC with Internet access

Instructions

Part 1: Use Ping to Document Network Latency

In Part 1, you will examine network latency of several websites in different parts of the globe. This process can be used in an enterprise production network to create a performance baseline.

Step 1: Verify connectivity.

Ping the following Regional Internet Registry (RIR) websites to verify connectivity:

```
C:\Users\User1> ping www.lacnic.net
C:\Users\User1> ping www.afrinic.net
C:\Users\User1> ping www.apnic.net
```

Note: Because www.ripe.net and www.arin.net do not reply to ICMP requests, they cannot be used for this lab.

Note: If the websites are resolved to IPv6 addresses, the option -4 can be used to resolve to IPv4 addresses if desired. The command becomes **ping -4 www.apnic.net**.

Step 2: Collect network data.

You will collect a sufficient amount of data to compute statistics on the **ping** output by sending out 25 echo requests to each address listed in Step 1. This step may require administrative privileges, depending upon your operating system. Record the results for each website to text files.

a. At the command prompt, type **ping** to list the available options.

```
C:\Users\User1> ping
```

b. Using the **ping** command with the count option, you can send 25 echo requests to the destination as illustrated below. Furthermore, it will create a text file with filename of **lacnic.txt** in the current directory. This text file will contain the results of the echo requests.

```
C:\Users\User1> ping –n 25 www.lacnic.net > lacnic.txt
```

Note: The terminal remains blank until the command has finished, because the output has been redirected to a text file, **lacnic.txt**, in this example. The **>** symbol is used to redirect the screen output to the file and overwrite the file if it already exists. If appending more results to the file is desired, replace **>** with **>>** in the command.

c. Repeat the **ping** command for the other websites.

```
C:\Users\User1> ping –n 25 www.afrinic.net > afrinic.txt
C:\Users\User1> ping –n 25 www.apnic.net > apnic.txt
```

Step 3: Verify data collection.

To verify that the files have been created, use the dir command to list the files in the directory. Also the wildcard * can be used to filter only the text files.

```
C:\Users\User1> dir *.txt
Volume in drive C is OS
 Volume Serial Number is 0A97-D265

 Directory of C:\Users\User1

02/07/2013  12:59 PM             1,642 afrinic.txt
02/07/2013  01:00 PM             1,615 apnic.txt
02/07/2013  12:58 PM             1,589 lacnic.txt
```

To see the results in the file created, use the **more** command at the command prompt.

```
C:\Users\User1> more lacnic.txt

Pinging www.lacnic.net [200.3.14.184] with 32 bytes of data:
Reply from 200.3.14.184: bytes=32 time=220ms TTL=51
Reply from 200.3.14.184: bytes=32 time=231ms TTL=51
Reply from 200.3.14.184: bytes=32 time=243ms TTL=51
Reply from 200.3.14.184: bytes=32 time=255ms TTL=51
Reply from 200.3.14.184: bytes=32 time=266ms TTL=51
<output omitted>
Reply from 200.3.14.184: bytes=32 time=522ms TTL=51
Reply from 200.3.14.184: bytes=32 time=195ms TTL=51
Reply from 200.3.14.184: bytes=32 time=207ms TTL=51
Reply from 200.3.14.184: bytes=32 time=219ms TTL=51
Reply from 200.3.14.184: bytes=32 time=232ms TTL=51

Ping statistics for 200.3.14.184:
    Packets: Sent = 25, Received = 24, Lost = 1 (4% loss),
Approximate round trip times in milli-seconds:
    Minimum = 175ms, Maximum = 522ms, Average = 253ms
```

Note: Press the Spacebar to display the rest of the file or press **q** to exit.

Record your results in the following table.

	Minimum	Maximum	Average
www.afrinic.net			
www.apnic.net			
www.lacnic.net			

Compare the delay results. How is delay affected by geographical location?

Part 2: Use Traceroute to Document Network Latency

The routes traced may go through many hops and a number of different ISPs depending on the size of the ISPs and the location of the source and destination hosts. The **traceroute** commands can also be used to observe network latency. In Part 2, the **tracert** command is used to trace the path to the same destinations in Part 1. The command **tracert** is the Windows version of the traceroute command.

The **tracert** command uses ICMP TTL Exceed packets and ICMP echo replies to trace the path.

Step 1: Use the tracert command and record the output to text files.

Copy the following commands to create the traceroute files:

```
C:\Users\User1> tracert www.lacnic.net > traceroute_lacnic.txt
C:\Users\User1> tracert www.afrinic.net > traceroute_afrinic.txt
C:\Users\User1> tracert www.apnic.net > traceroute_apnic.txt
```

Note: If the websites are resolved to IPv6 addresses, the option -4 can be used to resolve to IPv4 addresses if desired. The command becomes **tracert -4 www.lacnic**.net **> traceroute_lacnic.txt**.

Step 2: Use the more command to examine the traced path.

a. Use the **more** command to access the content of these files:

```
C:\Users\User1> more traceroute_lacnic.txt

Tracing route to www.lacnic.net [200.3.14.184]
over a maximum of 30 hops:

  1     3 ms     1 ms     2 ms  192.168.0.1
  2     *        *        *     Request timed out.
  3    14 ms    10 ms     9 ms  173-219-1-12.suddenlink.net [173.219.1.12]
  4    39 ms    38 ms    45 ms  173-219-1-232.suddenlink.net [173.219.1.232]
  5     *       38 ms    40 ms  173-219-1-98.suddenlink.net [173.219.1.98]
  6     *       35 ms    38 ms  lag-102.ear1.Chicago3.Level3.net [4.28.58.177]
  7     *        *        *     Request timed out.
  8    80 ms    79 ms    77 ms  GLOBAL-CROS.ear3.Miami2.Level3.net [4.15.156.54]
  9   341 ms   221 ms   222 ms  et-0-0-4-0.ptx-b.spo-piaf.algartelecom.com.br
[168.197.23.182]
 10     *        *        *     Request timed out.
 11   197 ms   222 ms   334 ms  201-048-035-089.static.ctbctelecom.com.br
[201.48.35.89]
 12   225 ms   175 ms   176 ms  xe-4-2-1-0.core1.nu.registro.br [200.160.0.180]
 13   269 ms   222 ms   221 ms  xe-0-0-0.ar3.nu.registro.br [200.160.0.249]
 14   217 ms   228 ms   218 ms  ae0-0.gw1.jd.lacnic.net [200.160.0.212]
 15     *      281 ms   220 ms  200.3.12.34
 16   231 ms   233 ms   212 ms  www.lacnic.net [200.3.14.184]

Trace complete.
```

In this example, it took less than 1 ms to receive a reply from the default gateway (192.168.0.1). In hop count 6, the round trip to 4.28.58.177 took an average of 37 ms. For the round trip to the final destination at www.lacnic .net took an average of 225 ms.

Between lines 8 and 9, there is more network delay as indicated by the round trip time increase from an average of 78 ms to 298 ms

b. Perform the same analysis with the rest of the tracert results.

What can you conclude regarding the relationship between the roundtrip time and geographical location?

Part 3: Extended Traceroute

Although **traceroute** has different implementations depending on the platform, all versions allow the user to adjust its behavior. In Windows, this can be done providing options and switches in the **tracert** command line.

a. Reverse name resolution (resolving an IP address to a domain name) can add a delay to **tracert** results and yield inaccurate results. To ensure **tracert** won't attempt to reverse resolve hop IP addresses, add the **–d** option to the **tracert** command line:

```
C:\Users\User1> tracert -d www.lacnic.net > traceroute_d_lacnic.txt
C:\Users\User1> tracert -d www.afrinic.net > traceroute_d_afrinic.txt
C:\Users\User1> tracert -d www.apnic.net > traceroute_d_apnic.txt
```

b. Use the **more** command to access the content of these files:

```
C:\Users\User1> more traceroute_d_lacnic.txt

Tracing route to www.lacnic.net [200.3.14.184]
over a maximum of 30 hops:

  1     4 ms     1 ms     1 ms  192.168.0.1
  2     *        *        *     Request timed out.
  3     *      931 ms   111 ms  173.219.221.12
  4    42 ms    41 ms    40 ms  173.219.17.232
  5    40 ms    37 ms    36 ms  173.219.234.108
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8    90 ms    81 ms    83 ms  4.15.156.54
  9   238 ms   221 ms   223 ms  168.197.23.182
 10     *        *        *     Request timed out.
 11   190 ms   246 ms   224 ms  201.48.35.89
 12   227 ms   222 ms   222 ms  200.160.0.180
 13   226 ms   222 ms   224 ms  200.160.0.249
 14   248 ms   199 ms   223 ms  200.160.0.212
 15   180 ms   270 ms   224 ms  200.3.12.34
 16   231 ms   218 ms   223 ms  200.3.14.184

Trace complete..
```

What is different about the **tracert** output when the **–d** option was added?

Note: Windows **tracert** will present a list of available options and their descriptions when issued without any options.

Note: Cisco IOS implementation of **traceroute** also allows for fine tuning but it does not rely on command line options. Cisco IOS extended traceroute presents a number of simple questions to allow the administrator to provide values for the desired parameters.

Reflection Questions

1. The **tracert** and **ping** results can provide important network latency information. What do you need to do if you want an accurate baseline picture regarding network latency for your network?

2. How can you use the baseline information?