CSc 4220/6220 – Fall 2018

Assignment #1

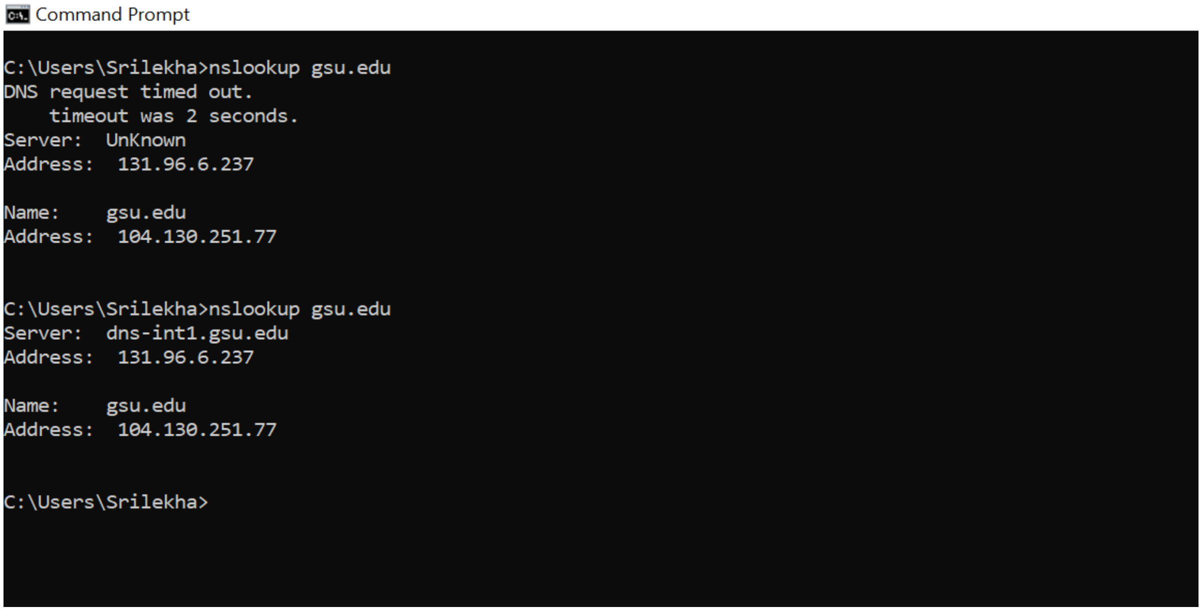
Due Friday, Aug 31st 11:59 pm
Late Deadline Tuesday, Sept 4th 11:59 pm

Section 1: Packet Sniffing with Wireshark

Wireshark is a free packet-sniffing tool where you can see how a packet is being transferred between source and destination and also the protocols involved in it.
Wireshark can be downloaded in both Mac/linux and windows.

- Start packet capturing by using Wireshark.
- Do a request for some website in your browser or do nslookup(DNS query) for some server in your command-prompt.
- Stop packet capture in wireshark.

You should get a trace that looks something like the following for a nslookup for gsu.edu (DNS query) in command prompt:



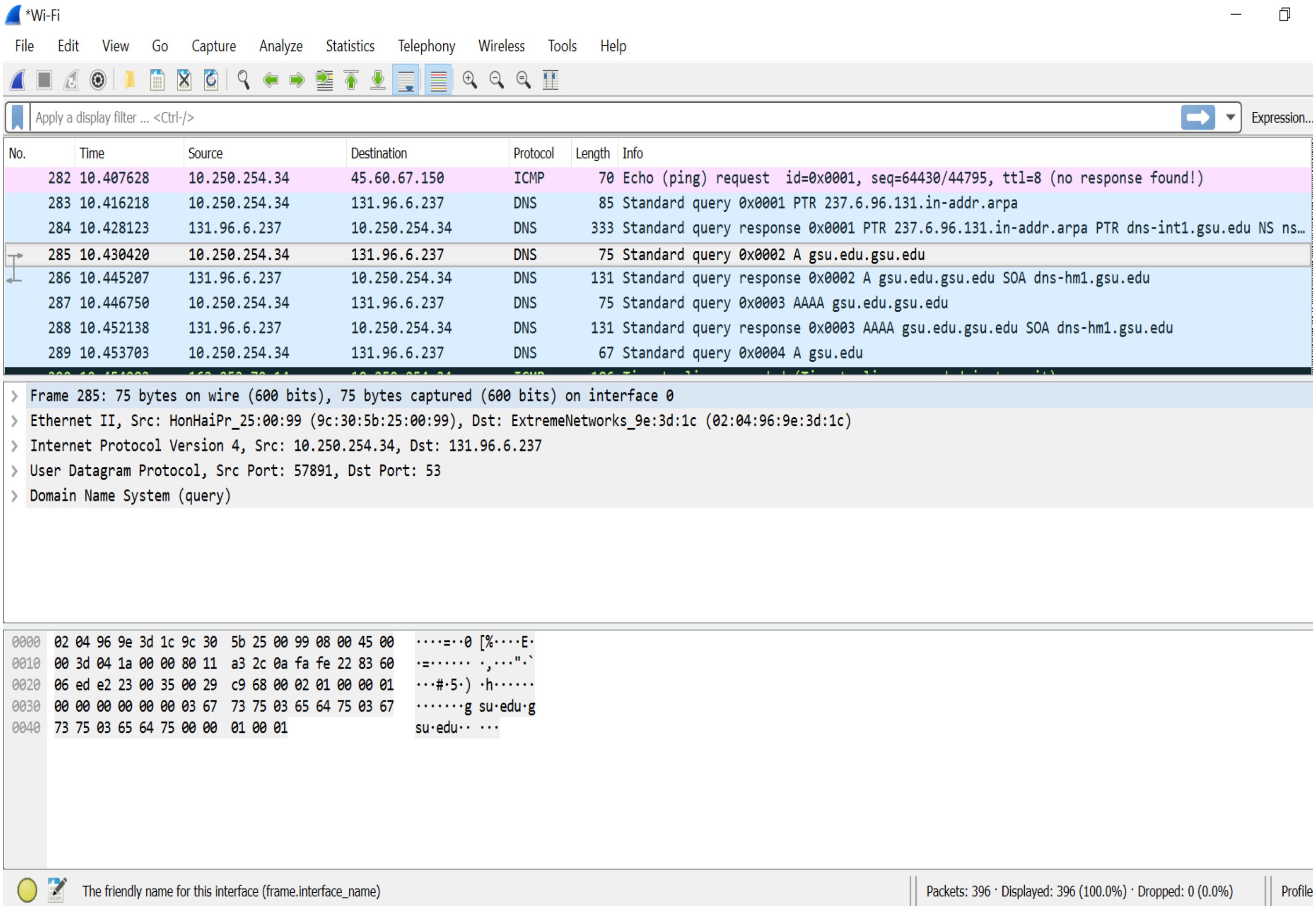
```
C:\Users\Srilekha>nslookup gsu.edu
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  131.96.6.237

Name:    gsu.edu
Address:  104.130.251.77


C:\Users\Srilekha>nslookup gsu.edu
Server:  dns-int1.gsu.edu
Address:  131.96.6.237

Name:    gsu.edu
Address:  104.130.251.77


C:\Users\Srilekha>
```



```
No.   Time        Source          Destination     Protocol Length Info
  282 10.407628   10.250.254.34   45.60.67.150    ICMP       70 Echo (ping) request  id=0x0001, seq=64430/44795, ttl=8 (no response found!)
  283 10.416218   10.250.254.34   131.96.6.237    DNS        85 Standard query 0x0001 PTR 237.6.96.131.in-addr.arpa
  284 10.428123   131.96.6.237    10.250.254.34   DNS       333 Standard query response 0x0001 PTR 237.6.96.131.in-addr.arpa PTR dns-int1.gsu.edu NS ns…
  285 10.430420   10.250.254.34   131.96.6.237    DNS        75 Standard query 0x0002 A gsu.edu.gsu.edu
  286 10.445207   131.96.6.237    10.250.254.34   DNS       131 Standard query response 0x0002 A gsu.edu.gsu.edu SOA dns-hm1.gsu.edu
  287 10.446750   10.250.254.34   131.96.6.237    DNS        75 Standard query 0x0003 AAAA gsu.edu.gsu.edu
  288 10.452138   131.96.6.237    10.250.254.34   DNS       131 Standard query response 0x0003 AAAA gsu.edu.gsu.edu SOA dns-hm1.gsu.edu
  289 10.453703   10.250.254.34   131.96.6.237    DNS        67 Standard query 0x0004 A gsu.edu
```

```
> Frame 285: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface 0
> Ethernet II, Src: HonHaiPr_25:00:99 (9c:30:5b:25:00:99), Dst: ExtremeNetworks_9e:3d:1c (02:04:96:9e:3d:1c)
> Internet Protocol Version 4, Src: 10.250.254.34, Dst: 131.96.6.237
> User Datagram Protocol, Src Port: 57891, Dst Port: 53
> Domain Name System (query)
```

```
0000  02 04 96 9e 3d 1c 9c 30  5b 25 00 99 08 00 45 00   ····=··0 [%····E·
0010  00 3d 04 1a 00 00 80 11  a3 2c 0a fa fe 22 83 60   ·=······ ·,···"·`
0020  06 ed e2 23 00 35 00 29  c9 68 00 02 01 00 00 01   ···#·5·) ·h······
0030  00 00 00 00 00 00 03 67  73 75 03 65 64 75 03 67   ·······g su·edu·g
0040  73 75 03 65 64 75 00 00  01 00 01                  su·edu·· ···
```

The friendly name for this interface (frame.interface_name)

Packets: 396 · Displayed: 396 (100.0%) · Dropped: 0 (0.0%)

Profile

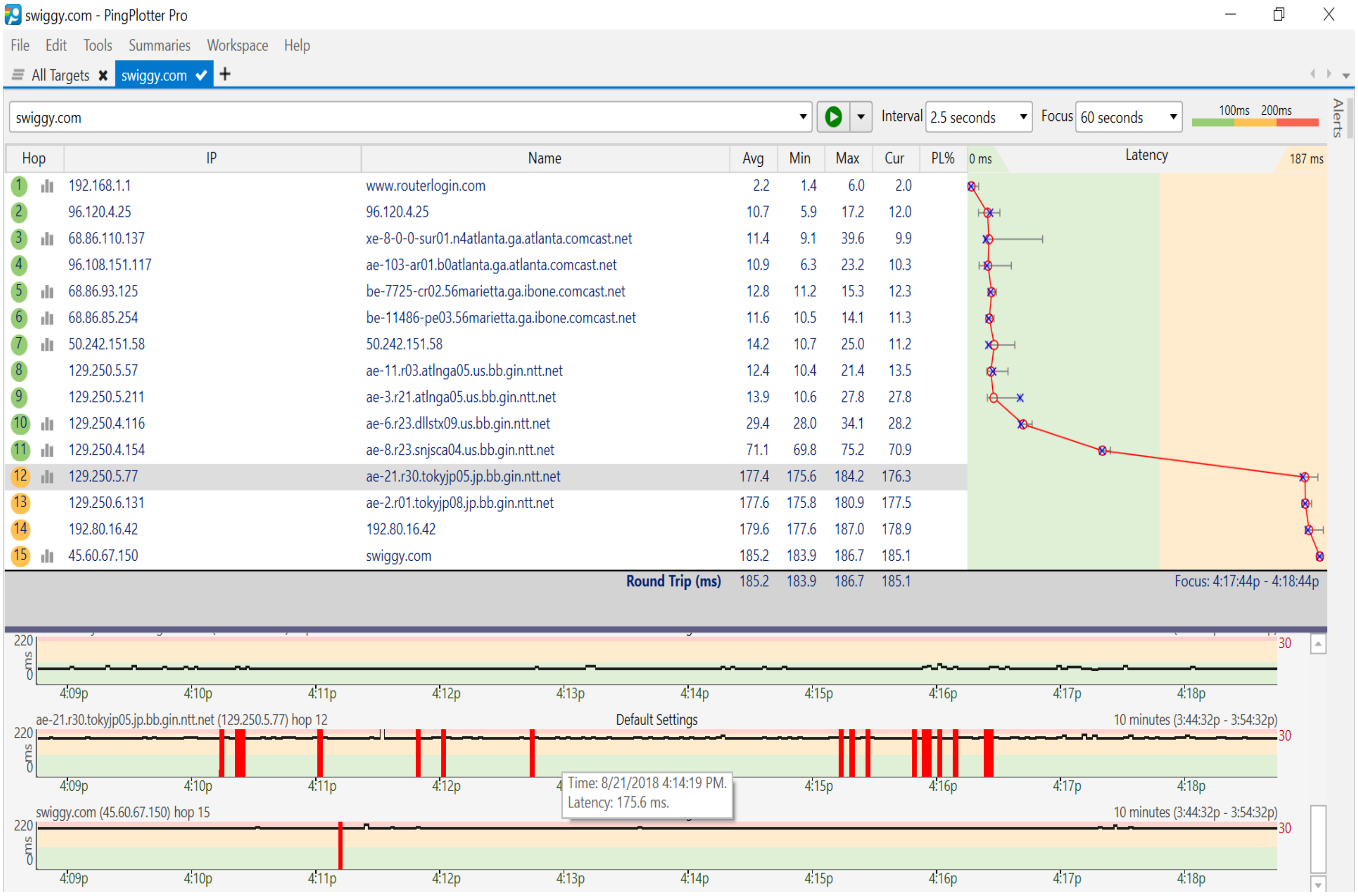
So, from above example, I can see the query request packet from my command-prompt in Wireshark. So, for the query in the second-level, the protocols that are used are seen and the last section gives the packet encoded in ASCII.

With the above example as a reference try to do the following things as an assignment.

1. Capture the packets with Wireshark when you access a website and see how many protocols are involved in the packet-transfer and list them by making screenshot like above with filters for each.
2. Find your own IP address in the screenshot that you take and provide a screenshot of it too.
3. Try to get the screenshots of your http messages with both GET and OK for one of the requested services and measure the time difference between those messages. (To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)
4. Do packet capture in Wireshark and make a DNS query with *nslookup* in command-prompt like the one that has been mentioned above and provide the packet transfer screenshot for wireshark and DNS query in the command-prompt.

Section 2: Traceroute the packets

In this section we will discuss about traceroute program that has been discussed in the class. Traceroute is a program which gives detailed information about the path and delay occurred between source and destination when we make a request. Traceroute in windows is possible through "*tracert*" command in the command-line but it won't allow to update the packet-size i.e. the length of datagrams that is needed in further lessons. It maintains a default size of 56 bytes. So, use "Pingplotter" in windows to see the traceroute program. It provides graphical representation of network connection and any packet loss can be determined through it.



swiggy.com

Interval 2.5 seconds

Focus 60 seconds

Hop	IP	Name	Avg	Min	Max	Cur	PL%
1	192.168.1.1	www.routerlogin.com	2.2	1.4	6.0	2.0	
2	96.120.4.25	96.120.4.25	10.7	5.9	17.2	12.0	
3	68.86.110.137	xe-8-0-0-sur01.n4atlanta.ga.atlanta.comcast.net	11.4	9.1	39.6	9.9	
4	96.108.151.117	ae-103-ar01.b0atlanta.ga.atlanta.comcast.net	10.9	6.3	23.2	10.3	
5	68.86.93.125	be-7725-cr02.56marietta.ga.ibone.comcast.net	12.8	11.2	15.3	12.3	
6	68.86.85.254	be-11486-pe03.56marietta.ga.ibone.comcast.net	11.6	10.5	14.1	11.3	
7	50.242.151.58	50.242.151.58	14.2	10.7	25.0	11.2	
8	129.250.5.57	ae-11.r03.atlnga05.us.bb.gin.ntt.net	12.4	10.4	21.4	13.5	
9	129.250.5.211	ae-3.r21.atlnga05.us.bb.gin.ntt.net	13.9	10.6	27.8	27.8	
10	129.250.4.116	ae-6.r23.dllstx09.us.bb.gin.ntt.net	29.4	28.0	34.1	28.2	
11	129.250.4.154	ae-8.r23.snjsca04.us.bb.gin.ntt.net	71.1	69.8	75.2	70.9	
12	129.250.5.77	ae-21.r30.tokyjp05.jp.bb.gin.ntt.net	177.4	175.6	184.2	176.3	
13	129.250.6.131	ae-2.r01.tokyjp08.jp.bb.gin.ntt.net	177.6	175.8	180.9	177.5	
14	192.80.16.42	192.80.16.42	179.6	177.6	187.0	178.9	
15	45.60.67.150	swiggy.com	185.2	183.9	186.7	185.1	
		Round Trip (ms)	185.2	183.9	186.7	185.1	

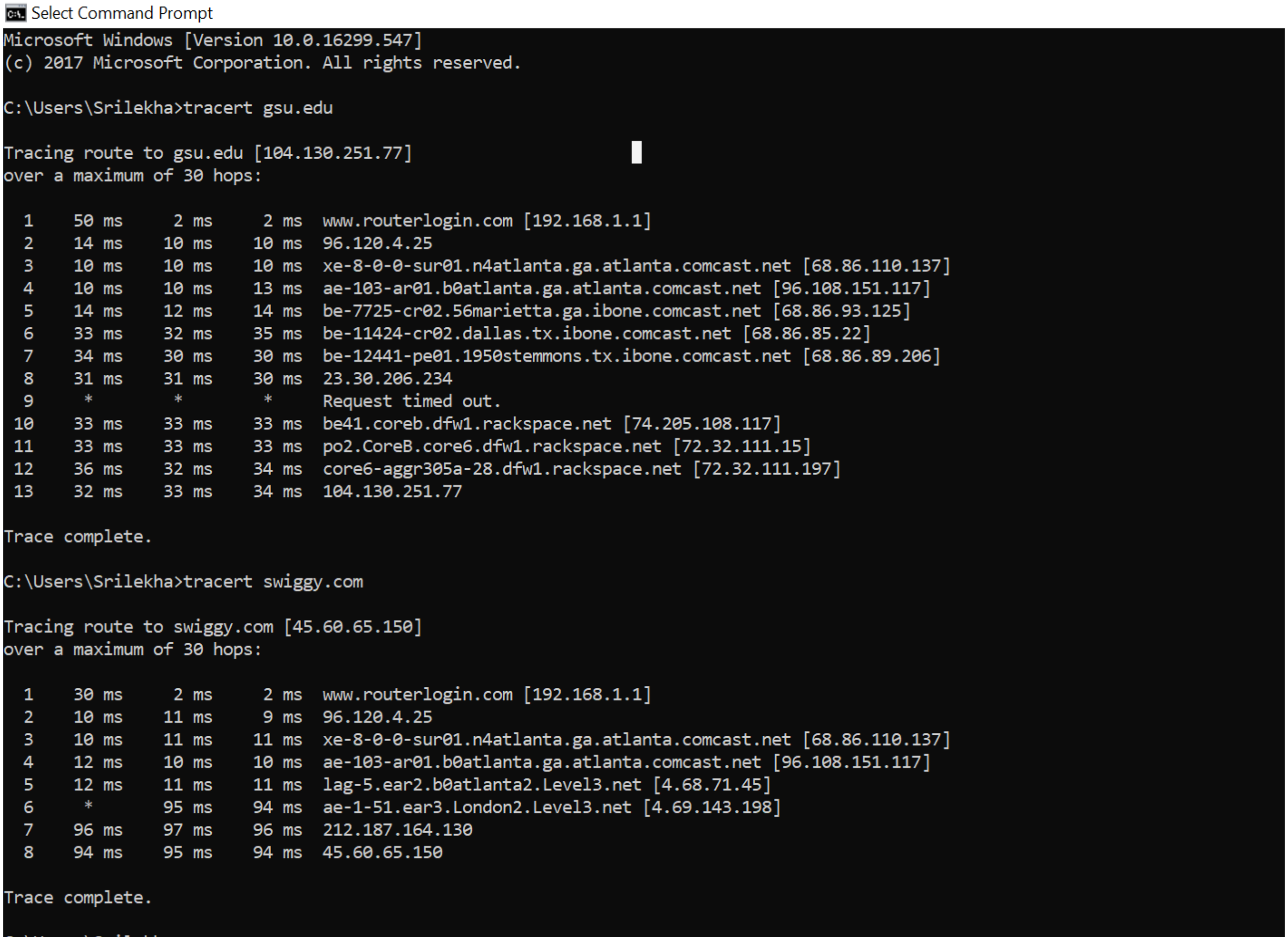
Focus: 4:17:44p - 4:18:44p

220
ms
0
30
4:09p
4:10p
4:11p
4:12p
4:13p
4:14p
4:15p
4:16p
4:17p
4:18p
ae-21.r30.tokyjp05.jp.bb.gin.ntt.net (129.250.5.77) hop 12
Default Settings
10 minutes (3:44:32p - 3:54:32p)
swiggy.com (45.60.67.150) hop 15

Time: 8/21/2018 4:14:19 PM.
Latency: 175.6 ms.

You can see in the above example I have requested a destination as a swiggy server in Asia and asked to traceroute the network to it from my PC. I can see 15 routers connected in between to serve my request and each router having their packet delays included above as an average for every request made to it and listed the overall round-trip delay. And for each router I see some packet loss as red mark in the below section.

You can also use traceroute program on the command-line in windows as below, the '*' symbol indicates packet-loss and the overall delays for each router for three attempts are shown as below.



Select Command Prompt

```
Microsoft Windows [Version 10.0.16299.547]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Srilekha>tracert gsu.edu

Tracing route to gsu.edu [104.130.251.77]
over a maximum of 30 hops:

  1    50 ms     2 ms     2 ms  www.routerlogin.com [192.168.1.1]
  2    14 ms    10 ms    10 ms  96.120.4.25
  3    10 ms    10 ms    10 ms  xe-8-0-0-sur01.n4atlanta.ga.atlanta.comcast.net [68.86.110.137]
  4    10 ms    10 ms    13 ms  ae-103-ar01.b0atlanta.ga.atlanta.comcast.net [96.108.151.117]
  5    14 ms    12 ms    14 ms  be-7725-cr02.56marietta.ga.ibone.comcast.net [68.86.93.125]
  6    33 ms    32 ms    35 ms  be-11424-cr02.dallas.tx.ibone.comcast.net [68.86.85.22]
  7    34 ms    30 ms    30 ms  be-12441-pe01.1950stemmons.tx.ibone.comcast.net [68.86.89.206]
  8    31 ms    31 ms    30 ms  23.30.206.234
  9     *        *        *     Request timed out.
 10    33 ms    33 ms    33 ms  be41.coreb.dfw1.rackspace.net [74.205.108.117]
 11    33 ms    33 ms    33 ms  po2.CoreB.core6.dfw1.rackspace.net [72.32.111.15]
 12    36 ms    32 ms    34 ms  core6-aggr305a-28.dfw1.rackspace.net [72.32.111.197]
 13    32 ms    33 ms    34 ms  104.130.251.77

Trace complete.

C:\Users\Srilekha>tracert swiggy.com

Tracing route to swiggy.com [45.60.65.150]
over a maximum of 30 hops:

  1    30 ms     2 ms     2 ms  www.routerlogin.com [192.168.1.1]
  2    10 ms    11 ms     9 ms  96.120.4.25
  3    10 ms    11 ms    11 ms  xe-8-0-0-sur01.n4atlanta.ga.atlanta.comcast.net [68.86.110.137]
  4    12 ms    10 ms    10 ms  ae-103-ar01.b0atlanta.ga.atlanta.comcast.net [96.108.151.117]
  5    12 ms    11 ms    11 ms  lag-5.ear2.b0atlanta2.Level3.net [4.68.71.45]
  6     *       95 ms    94 ms  ae-1-51.ear3.London2.Level3.net [4.69.143.198]
  7    96 ms    97 ms    96 ms  212.187.164.130
  8    94 ms    95 ms    94 ms  45.60.65.150

Trace complete.
```

- Linux/Unix/MacOS. With the Unix/MacOS *traceroute* command, the size of the UDP datagram sent towards the destination can be explicitly set by indicating the number of bytes in the datagram; this value is entered in the *traceroute* command line immediately after the name or address of the destination. For example, to send *traceroute* datagrams of 2000 bytes towards gsu.edu, the command would be:
  %traceroute gsu.edu 2000
  And on a default if we don't mention any packet-size in the above command, it will take 56bytes which is needed for the problems below and the '*' symbol in the trace shows packet-loss like windows.

So, from above information as reference, I want you to try the following

5. Perform a traceroute with destination on the same continent and try to see the traffic at three different hours of the day and provide screenshots of each attempt.
6. Find the numbers of routers at each of the three hours
7. Observe any path change in any attempt and observe which of the peer ISP's has the largest delay in all three attempts
8. Do the above (5,6,7) for some destination outside the continent and provide screenshots as well.
9. Observe any packet-loss with any router in between source and destination and screenshot them like the one in the above in information section.

Section 3:

10. Briefly give an example for each of the delay in the network that have been mentioned in the class.

NOTE: You can learn many interesting things like analyzing network, any intruder in your network etc .. Please learn to use wireshark and pingplotter through Youtube vidoes so that it comes in handy for the following chapters.