Gaurav Jain
2019CS10349

COL334 : Computer Networks
Assignment 1

22-08-2021

1 Networking Tools

Basic networking tools like *ifconfig, nslookup, ping* and *traceroute* are used for answering the following questions.

a) The IP address of a machine is determined using the *ifconfig* tool. The IP address of my machine when connected to different networks is-

   1. Home Broadband Network : 192.168.1.18
   2. Mobile Hotspot Network : 192.168.43.61
   3. IIT Delhi VPN : 10.52.10.2

b) The IP (IPv4) addresses of *www.google.com* and *www.facebook.com* is determined using *nslookup*. DNS server can be passed as an argument in the *nslookup* command.

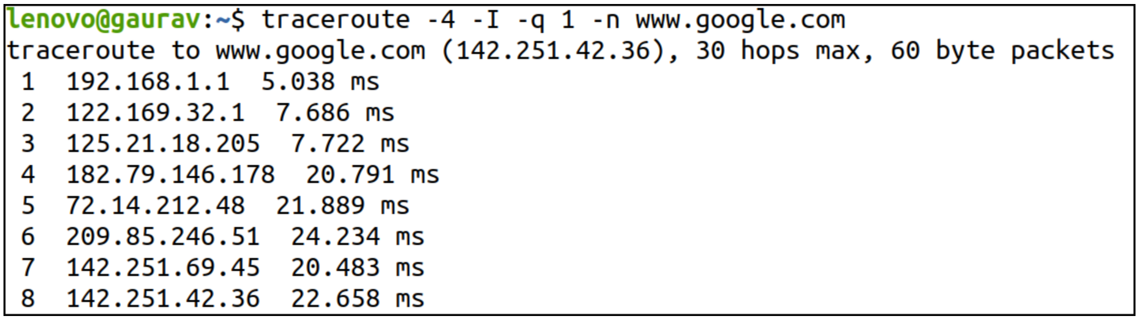
   1. Using local DNS server
      i. *www.google.com* : 142.251.42.36
      ii. *www.facebook.com* : 31.13.79.35
   2. Using Google's DNS server (8.8.8.8)
      i. *www.google.com* : 142.250.183.164
      ii. *www.facebook.com* : 31.13.79.35

c) The *ping* packets are sent with different parameter values like number of requests, packets size, TTL size and timeout. The maximum packets size is determined by running *ping* for different values of packets size manually. The maximum packets size for some destinations are listed below-

   1. *www.iitd.ac.in* : 1464(1492) bytes (size of header is 28 bytes).
   2. *www.google.com* : 68(96) bytes.
   3. *www.facebook.com* : 1464(1492) bytes.

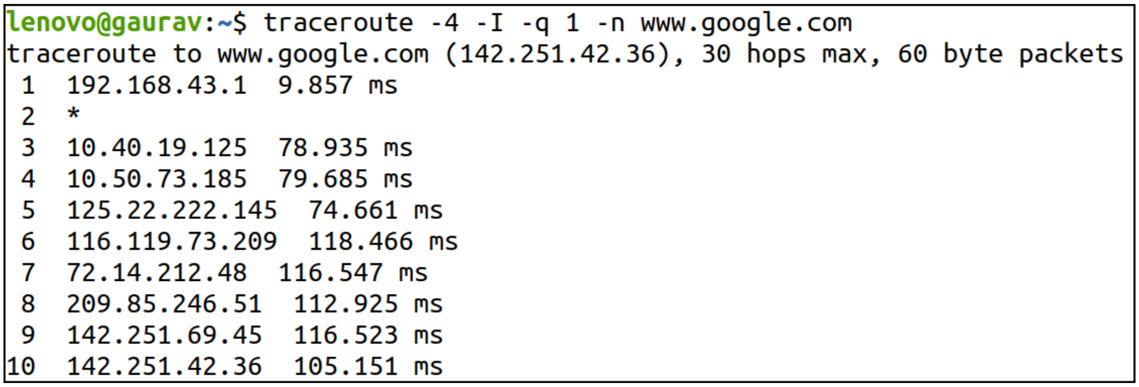
   By observing the maximum packets size of the above destinations, we can say that the maximum ping packets size is not same for all destinations.

d) *Traceroute* for *www.google.com* is shown below with two ISPs.



```
lenovo@gaurav:~$ traceroute -4 -I -q 1 -n www.google.com
traceroute to www.google.com (142.251.42.36), 30 hops max, 60 byte packets
 1  192.168.1.1  5.038 ms
 2  122.169.32.1  7.686 ms
 3  125.21.18.205  7.722 ms
 4  182.79.146.178  20.791 ms
 5  72.14.212.48  21.889 ms
 6  209.85.246.51  24.234 ms
 7  142.251.69.45  20.483 ms
 8  142.251.42.36  22.658 ms
```

Figure 1: *traceroute* with Home Broadband Network



```
lenovo@gaurav:~$ traceroute -4 -I -q 1 -n www.google.com
traceroute to www.google.com (142.251.42.36), 30 hops max, 60 byte packets
 1  192.168.43.1  9.857 ms
 2  *
 3  10.40.19.125  78.935 ms
 4  10.50.73.185  79.685 ms
 5  125.22.222.145  74.661 ms
 6  116.119.73.209  118.466 ms
 7  72.14.212.48  116.547 ms
 8  209.85.246.51  112.925 ms
 9  142.251.69.45  116.523 ms
10  142.251.42.36  105.151 ms
```

Figure 2: *traceroute* with Mobile Hotspot Network

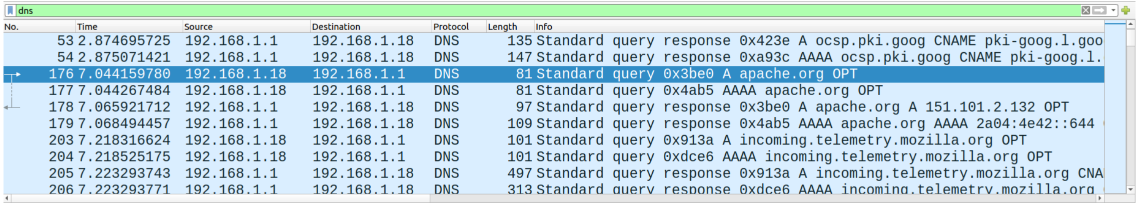
Observations:

1. To force traceroute to use IPv4, -4 flag is added to the command.
2. Some routers don't respond to UDP packets. -I flag is added to use ICMP packets.
3. Queue size is limited by using -q 1.
4. To hide domain names corresponding to IP addresses, -n flag is used.

2 Packet Analysis

Wireshark is used to grab all packets while visiting the website *http://apache.org* and the following results are reported-

a) The "dns" filter is applied on the packet trace of *http://apache.org*. The DNS request is at line 176 and the response to this DNS request is at line 178 of the packet trace present below.
Thus, the time taken to complete the request is 7.065921712 - 7.044159780 = 0.021761932 seconds.



dns

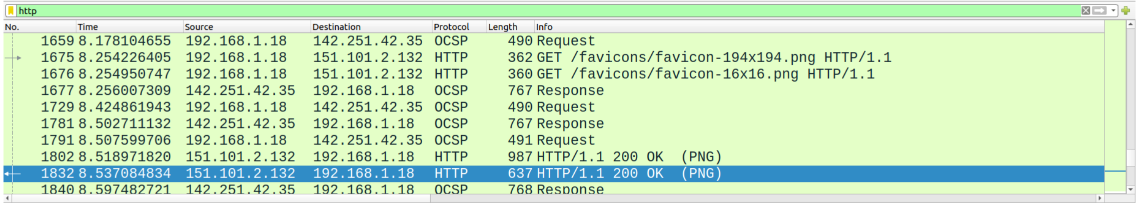
No.	Time	Source	Destination	Protocol	Length	Info
53	2.874695725	192.168.1.1	192.168.1.18	DNS	135	Standard query response 0x423e A ocsp.pki.goog CNAME pki-goog.l.goo
54	2.875071421	192.168.1.1	192.168.1.18	DNS	147	Standard query response 0xa93c AAAA ocsp.pki.goog CNAME pki-goog.l.
176	7.044159780	192.168.1.18	192.168.1.1	DNS	81	Standard query 0x3be0 A apache.org OPT
177	7.044267484	192.168.1.18	192.168.1.1	DNS	81	Standard query 0x4ab5 AAAA apache.org OPT
178	7.065921712	192.168.1.1	192.168.1.18	DNS	97	Standard query response 0x3be0 A apache.org A 151.101.2.132 OPT
179	7.068494457	192.168.1.1	192.168.1.18	DNS	109	Standard query response 0x4ab5 AAAA apache.org AAAA 2a04:4e42::644
203	7.218316624	192.168.1.18	192.168.1.1	DNS	101	Standard query 0x913a A incoming.telemetry.mozilla.org OPT
204	7.218525175	192.168.1.18	192.168.1.1	DNS	101	Standard query 0xdce6 AAAA incoming.telemetry.mozilla.org OPT
205	7.223293743	192.168.1.1	192.168.1.18	DNS	497	Standard query response 0x913a A incoming.telemetry.mozilla.org CNA
206	7.223293771	192.168.1.1	192.168.1.18	DNS	313	Standard query response 0xdce6 AAAA incoming.telemetry.mozilla.org

Figure 3: "dns" filter on packet trace of *http://apache.org*

b) After applying the "http" filter on the packet trace, it is observed that separate HTTP request is sent for all components of a website. There are various HTTP requests for text, css, bootstrap, javascipt, slideshow and images. Requests of different images are also separate.

   The approximate number of HTTP requests for *http://apache.org* is around 25. This means that complex websites are rendered part-wise by the browser. Essential parts like text, css and javascript are rendered first followed by heavy parts like images and videos.

c) The first DNS request for *http://apache.org* is at 7.044159780 second (see Figure 3 line 176) and the last content object is received at 8.537084834 second (see Figure 4 line 1832).
Thus, the total time taken for rendering the website is 1.492925054 seconds.



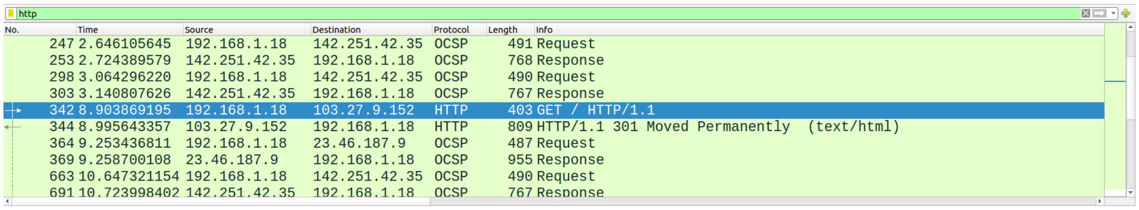
```
http
No.   Time        Source        Destination   Protocol Length Info
1659 8.178104655 192.168.1.18  142.251.42.35 OCSP       490 Request
1675 8.254226405 192.168.1.18  151.101.2.132 HTTP       362 GET /favicons/favicon-194x194.png HTTP/1.1
1676 8.254950747 192.168.1.18  151.101.2.132 HTTP       360 GET /favicons/favicon-16x16.png HTTP/1.1
1677 8.256007309 142.251.42.35 192.168.1.18  OCSP       767 Response
1729 8.424861943 192.168.1.18  142.251.42.35 OCSP       490 Request
1781 8.502711132 142.251.42.35 192.168.1.18  OCSP       767 Response
1791 8.507599706 192.168.1.18  142.251.42.35 OCSP       491 Request
1802 8.518971820 151.101.2.132 192.168.1.18  HTTP       987 HTTP/1.1 200 OK  (PNG)
1832 8.537084834 151.101.2.132 192.168.1.18  HTTP       637 HTTP/1.1 200 OK  (PNG)
1840 8.597482721 142.251.42.35 192.168.1.18  OCSP       768 Response
```

Figure 4: Last Content Object received from *http://apache.org*

d) No HTTP traffic is observed when the "http" filter is applied on packet trace of *http://www.cse.iitd.ac.in*. A single response is present, which shows a 301 Moved Permanently error (see Figure 5 lines 342-344).

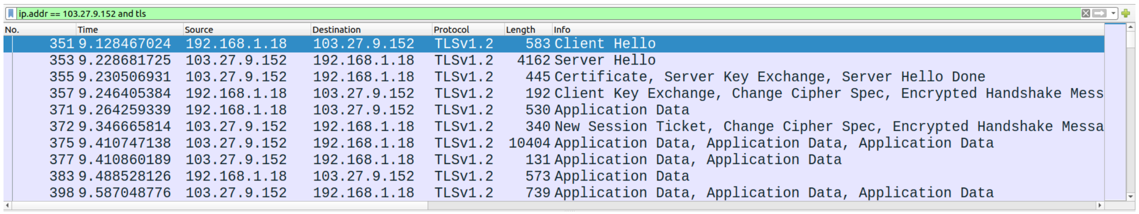
A possible explanation behind this is *http://www.cse.iitd.ac.in* uses HTTPS, whereas *http://apache.org* uses both HTTP and HTTPS. The only difference between HTTP and HTTPS is HTTPS uses encryption in HTTP requests and responses.

Thus, when we try to trace the packet of *http://www.cse.iitd.ac.in*, it is automatically redirected to HTTPS, and we observe no HTTP traffic. The traffic is visible in the TLS (TLSv1.2) protocol, which encrypts the HTTP requests and responses. Figure 6 shows the HTTP requests and responses present in TLS layer in form of 'Application Data'.



```
http
No.   Time         Source        Destination   Protocol Length Info
 247 2.646105645  192.168.1.18  142.251.42.35 OCSP       491 Request
 253 2.724389579  142.251.42.35 192.168.1.18  OCSP       768 Response
 298 3.064296220  192.168.1.18  142.251.42.35 OCSP       490 Request
 303 3.140807626  142.251.42.35 192.168.1.18  OCSP       767 Response
 342 8.903869195  192.168.1.18  103.27.9.152  HTTP       403 GET / HTTP/1.1
 344 8.995643357  103.27.9.152  192.168.1.18  HTTP       809 HTTP/1.1 301 Moved Permanently  (text/html)
 364 9.253436811  192.168.1.18  23.46.187.9   OCSP       487 Request
 369 9.258700108  23.46.187.9   192.168.1.18  OCSP       955 Response
 663 10.647321154 192.168.1.18  142.251.42.35 OCSP       490 Request
 691 10.723998402 142.251.42.35 192.168.1.18  OCSP       767 Response
```

Figure 5: "http" filter on packet trace of *http://www.cse.iitd.ac.in*

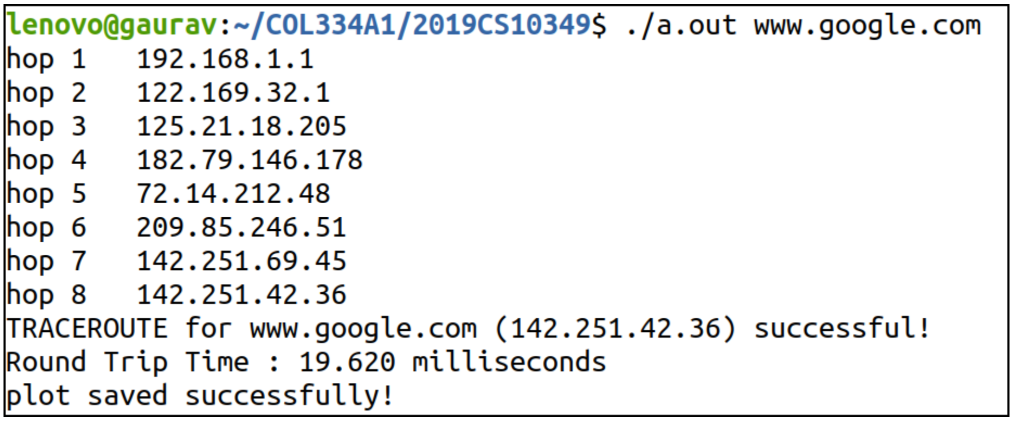


```
ip.addr == 103.27.9.152 and tls
No.   Time        Source        Destination  Protocol Length Info
 351 9.128467024 192.168.1.18  103.27.9.152 TLSv1.2    583 Client Hello
 353 9.228681725 103.27.9.152  192.168.1.18 TLSv1.2   4162 Server Hello
 355 9.230506931 103.27.9.152  192.168.1.18 TLSv1.2    445 Certificate, Server Key Exchange, Server Hello Done
 357 9.246405384 192.168.1.18  103.27.9.152 TLSv1.2    192 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Mess
 371 9.264259339 192.168.1.18  103.27.9.152 TLSv1.2    530 Application Data
 372 9.346665814 103.27.9.152  192.168.1.18 TLSv1.2    340 New Session Ticket, Change Cipher Spec, Encrypted Handshake Messa
 375 9.410747138 103.27.9.152  192.168.1.18 TLSv1.2  10404 Application Data, Application Data, Application Data
 377 9.410860189 103.27.9.152  192.168.1.18 TLSv1.2    131 Application Data, Application Data
 383 9.488528126 192.168.1.18  103.27.9.152 TLSv1.2    573 Application Data
 398 9.587048776 103.27.9.152  192.168.1.18 TLSv1.2    739 Application Data, Application Data, Application Data
```

Figure 6: "tls" filter on packet trace of *http://www.cse.iitd.ac.in*

3 Traceroute using Ping

Traceroute is implemented using the *Ping* command in C++ language. Calls to the *Ping* command are made using *system()* function of C++. Sample output is present below where the input destination domain is *www.google.com*. The RTT vs Hops plot is also present below.



```
lenovo@gaurav:~/COL334A1/2019CS10349$ ./a.out www.google.com
hop 1   192.168.1.1
hop 2   122.169.32.1
hop 3   125.21.18.205
hop 4   182.79.146.178
hop 5   72.14.212.48
hop 6   209.85.246.51
hop 7   142.251.69.45
hop 8   142.251.42.36
TRACEROUTE for www.google.com (142.251.42.36) successful!
Round Trip Time : 19.620 milliseconds
plot saved successfully!
```

Figure 7: A sample output for *www.google.com*

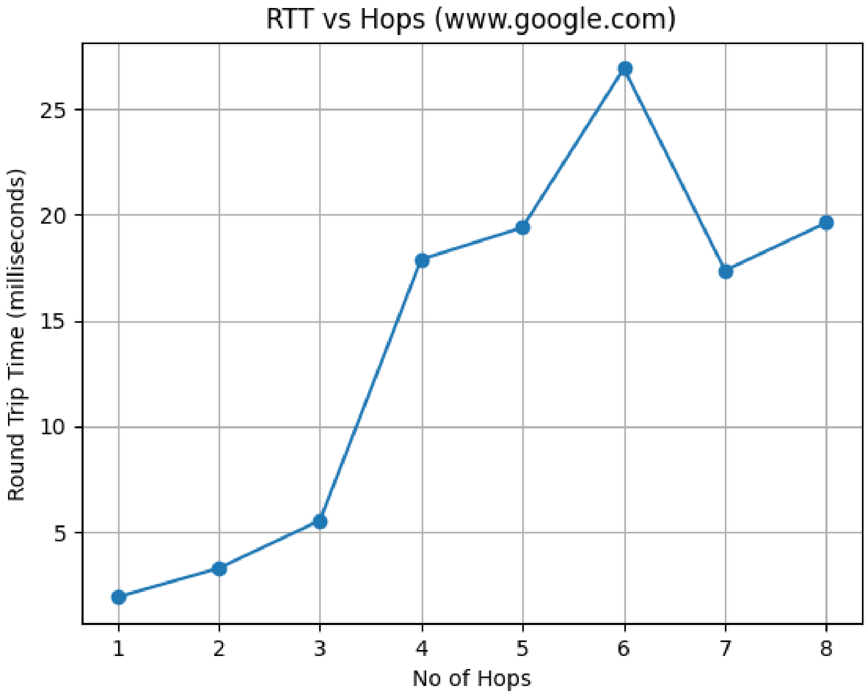


Figure 8: RTT vs Hops plot for *www.google.com*