Aakash Chaudhary (2023MCS2483) & Anjali Singh (2023JCS2565)

1. Network analysis

   (a) At your home (not hostel room which is on the IITD network) or from a 4G or other cellular connection, run traceroute via your Ethernet and WiFi networks for www.iitd.ac.in, and note the IP addresses seen on the path. If your ISP seems to be blocking packets on the path to the www.iitd.ac.in network then try with different destinations like www.google.com or www.facebook.com or www.nytimes.com or www.indianexpress.com, etc.

   (b) Report any curious things you notice, like some paths that default to IPv6 and how you can force traceroute to use IPv4, any private IP address spaces you notice like 10.0.0.0 to 10.255.255.255, or 172.16.0.0 to 172.31.255.255, or 192.168.0.0 to 192.168.255.255, missing routers along the path that do not seem to reply to the traceroute requests, etc.

   (c) Ping allows you to specify the size of packets to send. What is the maximum size of ping packets that you are able to send?

Solution:

(a) The table below shows the IP addresses seen on the path while reaching www.google.com. We have used the command `sudo traceroute www.iitd.ac.in -I --max-hops=255`. IP address for www.iitd.ac.in was `103.27.9.24` .

Hop Count	PACKET 1 IP Address	PACKET 2 IP Address	PACKET 3 IP Address)
1	192.168.43.1	192.168.43.1	192.168.43.1
2	*	*	*
3	10.71.83.35	10.71.83.51	10.71.83.51
4	172.26.100.118	172.26.100.118	172.26.100.118
5	172.26.100.102	172.26.100.102	172.26.100.102
6	192.168.44.26	192.168.44.28	192.168.44.26
7	*	*	*
8	*	*	*
9	*	*	*
10	*	*	*
11	136.232.148.178	136.232.148.178	136.232.148.178
12	*	*	*
13	*	*	*
14	*	*	*
15	103.27.9.24	103.27.9.24	103.27.9.24
16	103.27.9.24	103.27.9.24	103.27.9.24
17	103.27.9.24	103.27.9.24	103.27.9.24

(b)
1. None of the paths have defaulted to IPv6 so we do not need to force traceroute to IPv4.
2. For some hops we have written * that means that router has blocked ICMP packet and thus we could not get the IP Address.
3. The IP Addresses mentioned in the 1, 3, 4, 5, and 6 hops are all private IP's.

(c) The maximum size of Ping Packets that we were able to send was 65527 as it was responding `message too long` for 65528 bytes of data.

```
anjali@anjali-Vostro-15-3568:~$ ping -s 65528 www.iitd.ac.in
PING www.iitd.ac.in(2001:df4:e000:29::212) 65528 data bytes
ping: sendmsg: Message too long
ping: sendmsg: Message too long
ping: sendmsg: Message too long
ping: sendmsg: Message too long
^C
--- www.iitd.ac.in ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3078ms

anjali@anjali-Vostro-15-3568:~$ ping -s 65527 www.iitd.ac.in
PING www.iitd.ac.in(2001:df4:e000:29::212) 65527 data bytes
65535 bytes from 2001:df4:e000:29::212: icmp_seq=3 ttl=61 time=116 ms
65535 bytes from 2001:df4:e000:29::212: icmp_seq=7 ttl=61 time=304 ms
65535 bytes from 2001:df4:e000:29::212: icmp_seq=8 ttl=61 time=378 ms
65535 bytes from 2001:df4:e000:29::212: icmp_seq=14 ttl=61 time=93.4 ms
65535 bytes from 2001:df4:e000:29::212: icmp_seq=15 ttl=61 time=56.2 ms
65535 bytes from 2001:df4:e000:29::212: icmp_seq=16 ttl=61 time=47.0 ms
65535 bytes from 2001:df4:e000:29::212: icmp_seq=17 ttl=61 time=71.2 ms
65535 bytes from 2001:df4:e000:29::212: icmp_seq=18 ttl=61 time=45.8 ms
65535 bytes from 2001:df4:e000:29::212: icmp_seq=19 ttl=61 time=53.6 ms
65535 bytes from 2001:df4:e000:29::212: icmp_seq=20 ttl=61 time=53.8 ms
```

Fig.1 Maximum size of Ping Packet delivered was of 65527 bytes.

2. Can you replicate the traceroute functionality using ping? Ping allows you to initialize a TTL. Write a simple script in which you use ping to replicate traceroute. You can write this script in bash or perl or python or any of your favourite scripting languages.

Solution:

```python
#!/usr/bin/env python3

import subprocess


def Traceroute(dest_ip):
    hops = 30
    nslookup = ["nslookup", dest_ip]
    res = subprocess.run(
        nslookup, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True
    )
    target = res.stdout.split("\n")[5].split()[1]
    print(f"Traceroute to {dest_ip} ({target})")

    for ttl in range(1, hops + 1):
        traceroute = ["ping", "-4", "-c", "3", "-t", str(ttl), "-w", "3",
            dest_ip]
        result = subprocess.run(
            traceroute, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text
                =True
        )
        lat = ["*", "*", "*"]

        ip = None
```

```
        count = 0
        for eachline in result.stdout.split("\n"):
            if "From" in eachline:
                ip = eachline.split()[1:3]
            elif "from" in eachline:
                ip = eachline.split()[3:5]

            if "Time to live exceeded" in eachline:
                break
            elif "time=" in eachline:
                lat[count] = str(eachline.split()[-2][5:]) + " ms"
                count += 1

        if ip is not None and ip != target:
            current_ip = ip[1][1:-1]

            traceroute = ["ping", "-4", "-c", "3", "-w", "3", current_ip]
            result = subprocess.run(
                traceroute, stdout=subprocess.PIPE, stderr=subprocess.PIPE,
                    text=True
            )
            for eachline in result.stdout.split("\n"):
                if "time=" in eachline:
                    lat[count] = str(eachline.split()[-2][5:]) + " ms"
                    count += 1

        if ip is None:
            print(f"{ttl}: * * *")
        else:
            if ("icmp" in ip[1]):
                ip[1] = "(" + ip[0] + ")"
            print(f"{ttl}: {ip[0]} {ip[1]} {lat[0]} {lat[1]} {lat[2]}")
            if ip[1][1:-2] == target:
                break

        if "0 received" in result.stderr:
            break


if __name__ == "__main__":
    dest_ip = input("Enter the Destination IP Address or Host Name ")
    Traceroute(dest_ip)
```

3. Internet architecture

   (a) In a neat tabular format, report the number of hops from the (3) traceroute sources to the above (5) destinations. If the pair of (traceroute source, destination) are geographically close to each other, does it roughly translate into fewer hops? Do Google and Facebook differ from the others in the number of hops required to reach them, irrespective of which traceroute source is used? Why would this be so?

   (b) Also report the latencies between the traceroute sources and the web-servers. Does the latency seem to be related to the number of hops, being higher when there are more hops? Why is this the case?

(c) Which of the destination web-servers are resolved to the same IP address irrespective of from where you do a traceroute to them? Why do you think some web-servers are resolved to different IP addresses when queried from different parts of the world? You can also use nslookup to change the DNS server that you want to use. You can also use this dig web interface which may help speed up things for you: https://www.digwebinterface.com

(d) If you do traceroutes from the same starting point to different IP addresses you found for the same web-server, do the paths appear different? Which ones are longer?

(e) Try tracerouting to Google and Facebook from different countries of traceroute servers around the world. Are you able to find any countries that do not seem to have their local ISPs directly peered with Google and Faceboook?

Solution:

(a) The table below represents the number of hops from the (3) traceroute sources to the (5) destinations.

Table 1. Number of Hops for Different Sources and Destinations.

Source / **Destination**	**Server 1** Austin, US 216.218.252.175	**Server 2** Berlin, DE 216.218.253.231	**Local Machine**
www.utah.edu	14	18	31
www.uct.ac.za	Max Limit Exceeded	Max Limit Exceeded	Max Limit Exceeded
www.iitd.ac.in	15	17	11
www.google.com	9	12	15
www.facebook.com	9	11	16

`traceroute` did not terminate for www.uct.ac.za even after setting up `max-hops` to 255 which is the maximum hop limit for `traceroute` command. Hence, we can conclude that www.uct.ac.za has blocked `traceroute` command from their firewall.

A work around to get the number of hops for www.uct.ac.za is to use `sudo traceroute -T www.uct.ac.za --max-hops=255` which results in 30 hops.

The Austin,US server had fewer hops as compared to Berlin,DE and Delhi,IN for www.utah.edu because it was closer geographically. Similarly can be said about www.iitd.ac.in due to geographical proximity, the number of hops was less for the local machine situated in Delhi, India.

As in the case of Google and Facebook, they have comparable number of hops for all the 3 traceroute sources, because they have large number of servers across the globe, hich results in having a server in proximity of each country. Hence comparable number of hops can be achieved.

(b) The table below represents the latencies (in ms) from the (3) traceroute sources to the (5) destinations.

Table 2. Latencies (in ms) for Different Sources and Destinations.

Source / Destination	Server 1 Austin, US 216.218.252.175	Server 2 Berlin, DE 216.218.253.231	Local Machine
www.utah.edu	37.669	130.924	274.325
www.uct.ac.za	-	-	-
www.iitd.ac.in	277.112	158.034	9.490
www.google.com	32.597	109.726	7.879
www.facebook.com	39.372	133.136	9.116

There seem to be no relation between the number of hops and latency from different sources and destinations. But we can see a direct correlation between the sources and www.utah.edu, as the number of hops increase the latency also increases. This happens due to no direction connection between the source and destination, hence the packets travel from the traceroute path.

(c) The destination web-servers - www.iitd.ac.in, www.utah.edu, www.uct.ac.za have same IP address irrespective of from where you do a traceroute to them. The web-servers which resolved to different IP addresses from different source locations are Google and Facebook, and this is due to having multiple servers around the globe and hence DNS resolution gives the IP address of the server closest to the source geographically.

(d) Using two different DNS servers for resolving www.google.com :
1) 202.164.44.246 (Amritsar) = 142.250.77.196
2) 202.53.95.14 (Hyderabad) = 142.250.195.36

The paths do have some routers common in their path, and the path using Amritsar DNS server takes 10 hops and Hyderbad DNS server takes 12 hops.

(e) The terabyte traceroute server (216.234.191.172) which is located in Edmonton, Alberta, Canada (AS13911) does not have a peer connection with Google (AS15169) and Facebook (AS32934). The same can be verified by checking if AS13911 exists in the peer group of Facebook and Google on the respective links. These links tell us about the peers of the given ASN.

4. Packet analysis

(a) Apply a "dns" filter on the packet trace, and see if you can find DNS queries and responses for www.iitd.ac.in. How long did it take for the Domain Name Server request-response to complete?

(b) Apply "http" filter on the packet trace and report the approximate number of HTTP requests that were generated. What can you tell from this observation about how web-pages are structured ad how browsers render complex pages with multiple images and files?

(c) Apply a filter such as "((ip.src==192.168.1.3 && ip.dst==10.7.174.111) || (ip.src==10.7.174.111 && ip.dst==192.168.1.3)) && tcp". As would be self-explanatory, this will filter for TCP packets moving between your browser and web-server. Recall that the source and destination IP addresses are a part of the network layer header, also called the IP layer since IP (Internet Protocol) is the most common network layer protocol. Find the number of TCP connections that were opened between your browser and the web server. The signature for a new TCP connection is a 3-way handshake: The client sends a SYN message to the server, the server replies with a SYN-ACK message, and the client then sends an ACK. You will find that several TCP connections were opened between your browser and the web-server. Is this the same as the number of HTTP requests for content objects that you found in the previous part? Do you find that some content objects are

fetched over the same TCP connection? Note that TCP connections are distinguished from one another based on the source port and destination port.

(d) Now try doing a trace for http://www.indianexpress.com and filter for "http". What do you find, is there any HTTP traffic? Browse through the entire trace without any filters, are you able to see the contents of any HTML and Javascript files being transferred? What just happened?

Solution:

(a) It took 0.01795 seconds for the DNS request-response to complete.

(b) The number of HTTP requests that were generated for act4d.iitd.ac.in was 10. We have observed that the website made ten separate HTTP requests to get different parts of the website, like the HTML code, the CSS files, the Logo, and Images, and the structure follows an order in how the website is rendered. First, the base HTML code is rendered then the CSS files are rendered simultaneously. And lastly, the images and logo were rendered. This tells us that the media is lazy-loaded only after the skeleton of the HTML code is rendered.

There is only 1 HTTP request generated for iitd.ac.in because the website loads a temporary page which redirects to home.iitd.ac.in. The GET request fetches the HTML code for the redirection page. The contents of home.iitd.ac.in is loaded using TLS, which is encrypted.

ip.addr == 103.27.9.5 && http

No.	Time	Source	Destination	Protocol	Length	Info
39	2.404901771	192.168.29.17	103.27.9.5	HTTP	487	GET / HTTP/1.1
45	2.750175125	103.27.9.5	192.168.29.17	HTTP/XML	1326	HTTP/1.1 200 OK
50	3.053922410	192.168.29.17	103.27.9.5	HTTP	462	GET /act4d/templates/beez/css/template.css HTTP/1.1
57	3.076092773	192.168.29.17	103.27.9.5	HTTP	462	GET /act4d/templates/beez/css/position.css HTTP/1.1
58	3.077686856	103.27.9.5	192.168.29.17	HTTP	1182	HTTP/1.1 200 OK (text/css)
62	3.079785237	192.168.29.17	103.27.9.5	HTTP	461	GET /act4d/templates/beez/css/general.css HTTP/1.1
65	3.085053394	192.168.29.17	103.27.9.5	HTTP	460	GET /act4d/templates/beez/css/layout.css HTTP/1.1
66	3.086590204	192.168.29.17	103.27.9.5	HTTP	437	GET /wiki1-bak/wiki1/statf0e.php HTTP/1.1
70	3.096985308	103.27.9.5	192.168.29.17	HTTP	1407	HTTP/1.1 200 OK (text/css)
79	3.101712708	103.27.9.5	192.168.29.17	HTTP	291	HTTP/1.1 200 OK (text/css)
82	3.102838458	103.27.9.5	192.168.29.17	HTTP	616	HTTP/1.1 404 Not Found (text/html)
83	3.137168058	103.27.9.5	192.168.29.17	HTTP	1290	HTTP/1.1 200 OK (text/css)
85	3.137806256	192.168.29.17	103.27.9.5	HTTP	469	GET /act4d/templates/beez/images/act4d.png HTTP/1.1
86	3.137977863	192.168.29.17	103.27.9.5	HTTP	458	GET /act4d/images/balazahir.jpg HTTP/1.1
149	3.231508345	192.168.29.17	103.27.9.5	HTTP	459	GET /act4d/templates/beez/css/print.css HTTP/1.1
153	3.347911982	103.27.9.5	192.168.29.17	HTTP	1338	HTTP/1.1 200 OK (text/css)
203	3.447173443	103.27.9.5	192.168.29.17	HTTP	917	HTTP/1.1 200 OK (PNG)
259	3.572333208	192.168.29.17	103.27.9.5	HTTP	464	GET /act4d/templates/beez/favicon.ico HTTP/1.1
291	3.635385867	103.27.9.5	192.168.29.17	HTTP	2717	HTTP/1.1 200 OK (JPEG JFIF image)

Fig.1 Filtered using HTTP filter on act4d.iitd.ac.in

((ipv6.src == 2001:df4:e000:29::212 && ipv6.dst == 2405:201:4007:b111:c521:a4a8:7b72:a996) || (ipv6.src == 2405:201:4007:b111:c521:a4a8:7b72:a996 && ipv6.dst == 2001:df4:e000:29::212)) && http

No.	Time	Source	Destination	Protocol	Length	Info
505	8.446384981	2405:201:4007:b111:…	2001:df4:e000:29::2…	HTTP	432	GET / HTTP/1.1
507	8.456303644	2001:df4:e000:29::2…	2405:201:4007:b111:…	HTTP	527	HTTP/1.1 302 Found (text/html)

Fig.2 Filtered using HTTP filter on iitd.ac.in

(c) The number of TCP connections opened between my browser and the web-server are 4. It is not the same as the number of HTTP requests for content objects. Some content objects are fetched over the same TCP connection, so the number of TCP connections is less than the number of HTTP requests.

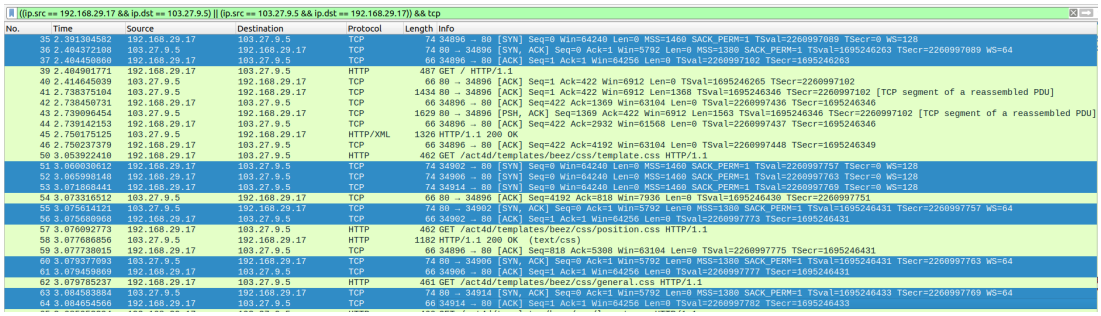

Fig.3 Filtered using TCP and Source-Destination IPs

(d) After applying the 'HTTP' filter, we do not see any HTTP traffic on Wireshark. Most of the traffic is TCP and TLS. And the application Data is transferred over TLS. After browsing the entire trace without filters, we could not see any HTML or Javascript files. All the Application data is transferred using TLS, and all the Application data is encrypted. So nothing can be viewed using Wireshark.