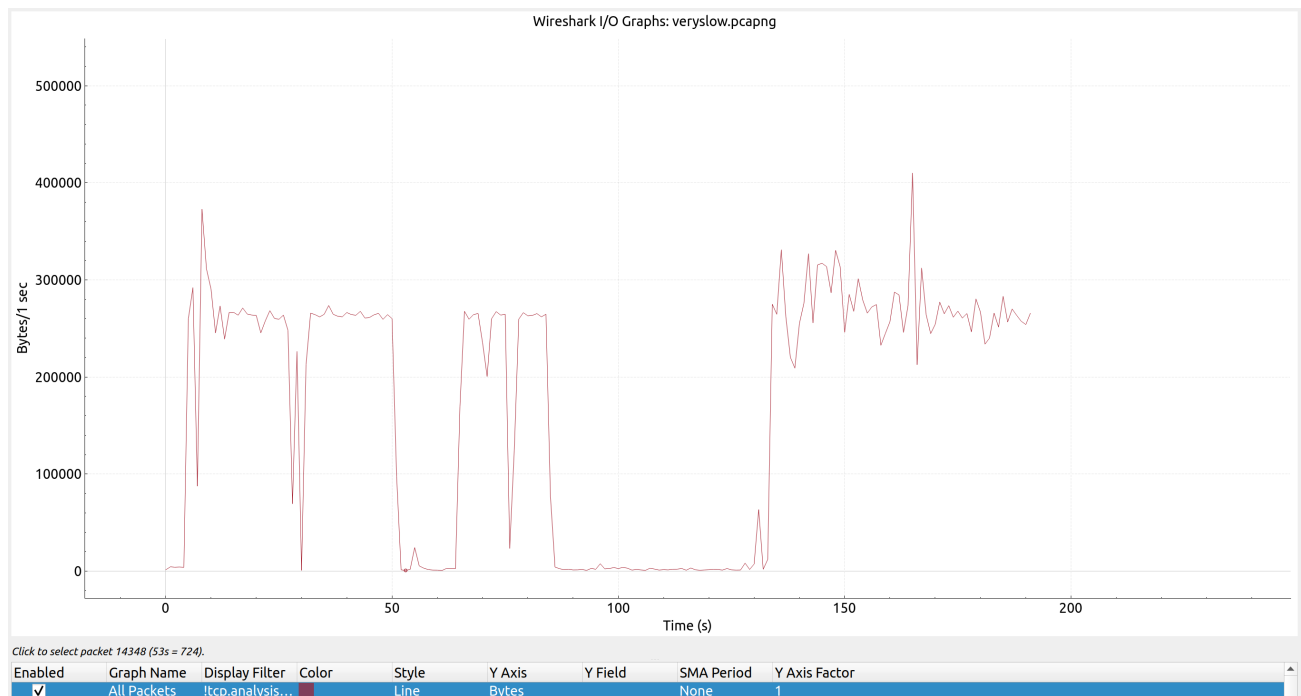


CN HW1

1.

1. Go to `Statistics -> I/O Graphs`

After adding a filter to remove TCP error packets, the transmit rate limit is 260 KBps \approx 2080 Kbps



2. `140.112.28.151`

Observe the packets using TCP and DNS protocol. We can find that `140.112.28.151` is an IP address that is frequently seen as either source or destination.

3. [Reference](#)

Apply filter `ip.src == 140.112.28.151 && tcp.srcport == 5555` and follow the stream to get the secret message sent via TCP

Apply filter `ip.dst == 140.112.28.151 && udp.dstport == 48763` and follow the stream to get the secret message sent via UDP

- secret message via TCP : 418 I'm a teapot
- secret message via UDP : Baby shark, doo doo doo doo doo doo

The main difference between TCP and UDP is that TCP is a connection-oriented protocol, meaning that client and server should establish a connection before transmitting data and should close the connection after transmitting the data. Therefore, more packets

(overheads) are needed to transfer data using TCP. On the otherhand, UDP does not require connection establishment and is able to transfer data directly.

4. [Reference](#)

- Network Layer Header

1. Check the IP Version field: In an IPv4 header, the IP version field has a value of '4' (or '0100' in binary), whereas in an IPv6 header, the IP version field has a value of '6' (or '0110' in binary).
2. IPv4 header has a Protocol field that indicates the protocol used by the upper-layer, while IPv6 header has a Next Header field that specifies the type of the next header, including both extension headers and upper-layer protocols.

- IPv4 Header

```
▼ Internet Protocol Version 4, Src: 152.199.43.83, Dst: 140.112.28.151
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 2948
    Identification: 0x104f (4175)
  ▶ Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 55
    Protocol: TCP (6)
```

- IPv6 Header

```
▼ Internet Protocol Version 6, Src: fe80::dbf2:a116:8126:23f5, Dst: ff02::fb
  0110 .... = Version: 6
  ▶ .... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
  .... 0010 0011 1110 1001 1011 = Flow Label: 0x23e9b
  Payload Length: 38
  Next Header: UDP (17)
  Hop Limit: 1
  Source Address: fe80::dbf2:a116:8126:23f5
  Destination Address: ff02::fb
```

- Link Layer Header

1. Type field indicates the IP version that is used for the packet

- IPv4

```
▼ Ethernet II, Src: Cisco_ff:fc:a8 (00:08:e3:ff:fc:a8), Dst: PcsCompu_a7:ec:5c (08:00:27:a7:ec:5c)
  ▶ Destination: PcsCompu_a7:ec:5c (08:00:27:a7:ec:5c)
  ▶ Source: Cisco_ff:fc:a8 (00:08:e3:ff:fc:a8)
  Type: IPv4 (0x0800)
```

- IPv6

```
▼ Ethernet II, Src: EliteGro_6a:a4:37 (1c:69:7a:6a:a4:37), Dst: IPv6mcast_0c (33:33:00:00:00:0c)
  ▶ Destination: IPv6mcast_0c (33:33:00:00:00:0c)
  ▶ Source: EliteGro_6a:a4:37 (1c:69:7a:6a:a4:37)
  Type: IPv6 (0x86dd)
```

5. Apply filter `dns contains wikipedia` and locate the packet that contains

```
Standard query response 0x50e7 A zh.wikipedia.org
```

1. The packet contains DNS query to find the IPv4 address of `zh.wikipedia.org` and the response sent from DNS server.
2.
 - type A : IPv4 address that a domain name corresponds to
 - type CNAME : Canonical name functions as an alias for a machine.
 - type NS : Name servers. Servers that store the correspondence between domain name and IP address
3. According to the response, `zh.wikipedia.org` has the same IPv4 address as `dyna.wikimedia.org`. Therefore, the IP address of `zh.wikipedia.org` is `103.102.166.224`

```
▼ Queries
  ▶ zh.wikipedia.org: type A, class IN
▼ Answers
  ▶ zh.wikipedia.org: type CNAME, class IN, cname dyna.wikimedia.org
  ▶ dyna.wikimedia.org: type A, class IN, addr 103.102.166.224
▼ Authoritative nameservers
  ▶ wikimedia.org: type NS, class IN, ns ns2.wikimedia.org
  ▶ wikimedia.org: type NS, class IN, ns ns0.wikimedia.org
  ▶ wikimedia.org: type NS, class IN, ns ns1.wikimedia.org
▼ Additional records
  ▶ ns0.wikimedia.org: type A, class IN, addr 208.80.154.238
  ▶ ns1.wikimedia.org: type A, class IN, addr 208.80.153.231
  ▶ ns2.wikimedia.org: type A, class IN, addr 198.35.27.27
  ▶ <Root>: type OPT
```

2.

1. Follow the TCP stream and get the result :

- UserName : cnta
- Password : ji32k7au4a83

```

220----- Welcome to Pure-FTPd [privsep] [TLS] -----
220-You are user number 1 of 5 allowed.
220-Local time is now 02:51. Server port: 21.
220-This is a private system - No anonymous login
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
USER cnta
331 User cnta OK. Password required
PASS ji32k7au4a83
230 OK. Current directory is /
SYST
215 UNIX Type: L8
FEAT
211-Extensions supported:
EPRT
IDLE
MDTM
SIZE
MFMT
REST STREAM
MLST type*;size*;sizd*;modify*;UNIX.mode*;UNIX.uid*;UNIX.gid*;unique*;
MLSD
AUTH TLS
PBSZ
PROT
UTF8
TVFS
ESTA
PASV
EPSV
SPSV

211 End.
TYPE I
200 TYPE is now 8-bit binary
EPSV
229 Extended Passive mode OK (|||30012|)
STOR lorem.txt
150 Accepted data connection
226-File successfully transferred
226 0.002 seconds (measured here), 0.93 Mbytes per second

```

- Based on the TCP stream found in problem 1, we can observe that packets containing requests to the server have a common destination port.

Based on the TCP stream found in problem 1, mark the packet that contains `STOR lorem.txt`, `STOR link.txt`, `STOR midterm.txt`, and `STOR puppy.png`. Packets containing file contents can be found near them. We can find the ports on the server that we send file contents with according to the destination ports of those packets.

- Server is listening to port 5000 for FTP requests
- The ports on the server we send file contents with are :

FileName	Port
lorem.txt	30012
links.txt	30013
midterm.txt	30010
puppy.png	30016

- Based on the TCP stream found in problem 1, mark the packet that contains `STOR midterm.txt`. A packet with length of 3127 bytes can be found near the packet. Follow the packet and found the content of `midterm.txt`

- FileName : midterm.txt
- Number of Questions : 10

C0MPU73r Ne7w0Rk\$
 M!d73rM Ex@M
 2023/10/25
 14:20~17:20

1. (10%)

Consider sending a large file of F bits from Host A to Host B. There are three links (and two switches) between A and B, and the links are uncongested (that is, no queuing delays). Host A segments the file into segments of S bits each and adds 80 bits of header to each segment, forming packets of $L = 80 + S$ bits. Each link has a transmission rate of R bps. Find the value of S that minimizes the delay of moving the file from Host A to Host B. Disregard propagation delay.

2. (10%)

Describe the different wireless technologies you use during the day and their characteristics. If you have a choice between multiple technologies, why do you prefer one over another?

3. (10%)

What advantage does a circuit-switched network have over a packet-switched network? What advantages does TDM have over FDM in a circuit-switched network?

4. (10%)

Consider sending a packet from a source host to a destination host over a fixed route. List the delay components in the end-to-end delay. Which of these delays are constant and which are variable?

5. (10%)

What are the five layers in the Internet protocol stack? What are the principal responsibilities of each of these layers?

6. (10%)

Consider a short, 30-meter link, over which a sender can transmit at a rate of 300 bits/sec in both directions. Suppose that packets containing data are 100,000 bits long, and packets containing only control (e.g., ACK or handshaking) are 200 bits long. Assume that N parallel connections each get $1/N$ of the link bandwidth. Now consider the HTTP protocol, and suppose that each downloaded object is 100 Kbits long, and that the initial downloaded object contains 10 referenced objects from the same sender. Would parallel downloads via parallel instances of non-persistent HTTP make sense in this case? Now consider persistent HTTP. Do you expect significant gains over the non-persistent case? Justify and explain your answer.

7. (10%)

Consider distributing a file of $F = 10$ Gbits to N peers. The server has an upload rate of $u_s = 1$ Gbps, and each peer has a download rate of $d_i = 200$ Mbps and an upload rate of u_i . For $N = 10, 100$, and $1,000$ and $u_i = 2$ Mbps, 10 Mbps, and 100 Mbps, prepare a chart giving the minimum distribution

3.

```

250-8BITMIME
250-SMTPUTF8
250-AUTH LOGIN PLAIN
250 STARTTLS
MAIL FROM:<prof.devil@notearuniv.edu>
250 Accepted
RCPT TO:<wanna.cry@notearuniv.edu>
250 Accepted
DATA
354 End data with <CR><LF>.<CR><LF>
Content-Disposition: inline
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain
MIME-Version: 1.0
Date: Mon, 4 Sep 2023 14:18:05 +0800
From: prof.devil@notearuniv.edu
To: wanna.cry@notearuniv.edu
Subject: Signing up for the course
X-Mailer: smtp-cli 3.10, see http://smtp-cli.logix.cz
Message-Id: <1693808285-381726@smtp-cli>

Hi Wanna Cry,=0D
=0D
Thank you for signing up for "Assembly Languages from Beginner to Quitter."=
=0D
Due to the capacity of the classroom, only some of the students can get the=
permission code.=0D
We are glad to tell you that you have successfully passed the course qualif=
ication check.=0D
Here's your course permission code: "toRt0R-53d-@CcuM54n-bibenduM"=0D
Please join the course before the 3rd week of this semester. Otherwise, thi=
s permission code will no longer be valid.=0D
On the other hand, DO NOT distribute your permission code to other students=
..=0D
Enjoy this course and see you next week!=0D
=0D
Best,=0D
Prof Devil=
.
250 OK: message queued
QUIT

```

1. [Reference](#)

According to the TCP stream, ESMTP protocol is used.

- Server Port : 4000
- Application Protocol : MIME
- Port that the server typically uses : 25

2. Follow the TCP stream

- Sender : prof.devil@notearuniv.edu
- Receiver : wanna.cry@notearuniv.edu
- Subject : Signing up for the course

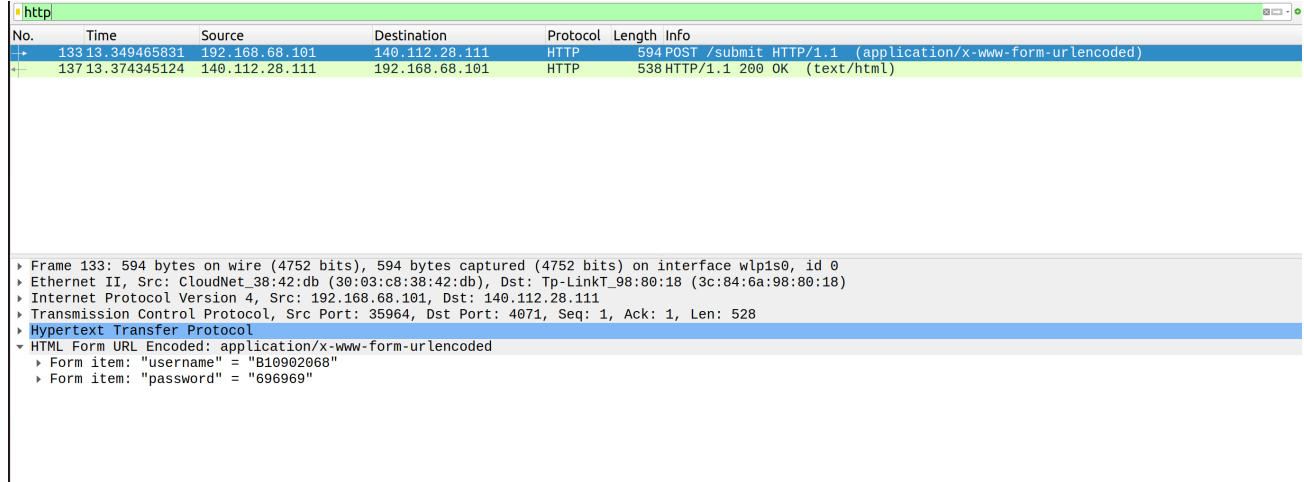
3. Yes

- Course : Assembly Languages from Beginner to Quitter.
- Permission Code : toRt0R-53d-@CcuM54n-bibenduM

4. No, the client is using TCP. The disadvantage without using TLS is that the transferring data are not encrypted. Therefore, it can be seen by others under the same internet.

4.

1. Find the packet that uses HTTP protocol to send a POST request to 140.112.28.111
Username and password can be found in the content of the packet.



The screenshot shows a Wireshark packet capture with the filter 'http'. Two packets are listed:

No.	Time	Source	Destination	Protocol	Length	Info
133	13.349465831	192.168.68.101	140.112.28.111	HTTP	594	POST /submit HTTP/1.1 (application/x-www-form-urlencoded)
137	13.374345124	140.112.28.111	192.168.68.101	HTTP	538	HTTP/1.1 200 OK (text/html)

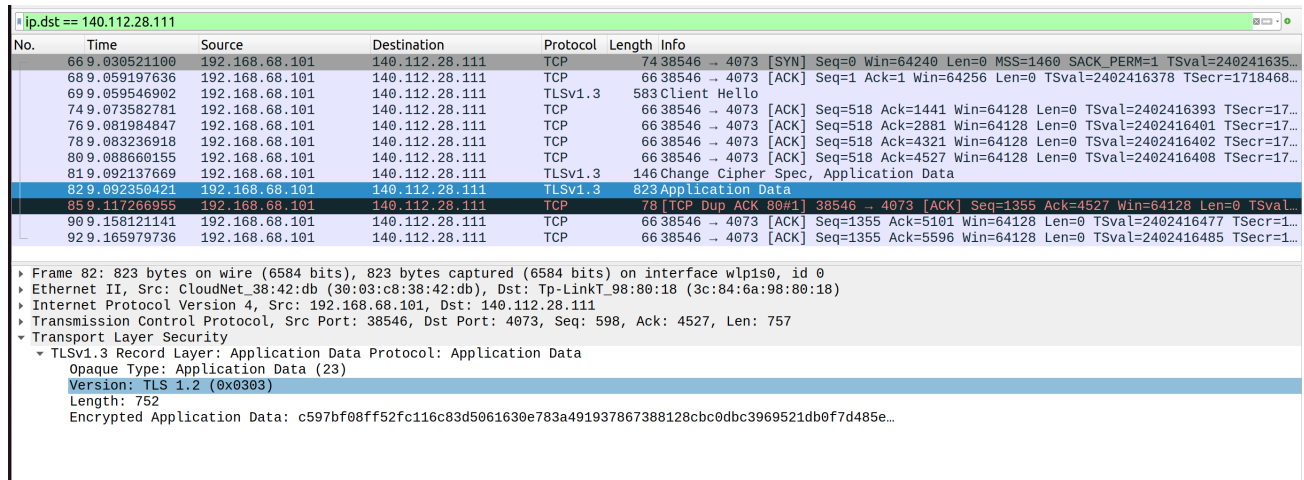
The details pane for the selected packet (No. 133) shows:

- Frame 133: 594 bytes on wire (4752 bits), 594 bytes captured (4752 bits) on interface wlp1s0, id 0
- Ethernet II, Src: CloudNet_38:42:db (30:03:c8:38:42:db), Dst: Tp-LinkT_98:80:18 (3c:84:6a:98:80:18)
- Internet Protocol Version 4, Src: 192.168.68.101, Dst: 140.112.28.111
- Transmission Control Protocol, Src Port: 35964, Dst Port: 4071, Seq: 1, Ack: 1, Len: 528
- Hypertext Transfer Protocol
- HTML Form URL Encoded: application/x-www-form-urlencoded
 - Form item: "username" = "B10902068"
 - Form item: "password" = "696969"

2. Reference

We cannot find any packet that contains the password; instead, we can only find a packet containing encrypted data using TLS protocol.

The result is caused by the use of HTTPS protocol. Data in packets are encrypted if HTTPS is used.



The screenshot shows a Wireshark packet capture with the filter 'ip.dst == 140.112.28.111'. Multiple packets are listed, including TCP SYN, ACK, and TLS records. The details pane for the selected packet (No. 82) shows:

- Frame 82: 823 bytes on wire (6584 bits), 823 bytes captured (6584 bits) on interface wlp1s0, id 0
- Ethernet II, Src: CloudNet_38:42:db (30:03:c8:38:42:db), Dst: Tp-LinkT_98:80:18 (3c:84:6a:98:80:18)
- Internet Protocol Version 4, Src: 192.168.68.101, Dst: 140.112.28.111
- Transmission Control Protocol, Src Port: 38546, Dst Port: 4073, Seq: 598, Ack: 4527, Len: 757
- Transport Layer Security
 - TLSv1.3 Record Layer: Application Data Protocol: Application Data
 - Opaque Type: Application Data (23)
 - Version: TLS 1.2 (0x0303)
 - Length: 752
 - Encrypted Application Data: c597bf08ff52fc116c83d5061630e783a491937867388128cbc0dbc3969521db0f7d485e...

5.

1. The web server sends a POST request to http://voip.csie.org:4071/submit when we press the send button
 - Command :

```
curl -i -X POST http://voip.csie.org:4071/submit\  
-H "Content-Type: application/x-www-form-urlencoded" \
```

```
-d "username=B10902068&password=51"
```

```
POST /submit HTTP/1.1
Host: voip.csie.org:4071
Connection: keep-alive
Content-Length: 29
Accept: text/html, */*; q=0.01
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://voip.csie.org:4070
Referer: http://voip.csie.org:4070/
Accept-Encoding: gzip, deflate
Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7,eo;q=0.6,zh-CN;q=0.5,th;q=0.4
```

```
username=B10902068&password=xHTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: http://voip.csie.org:4070
Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, PATCH, DELETE
Access-Control-Allow-Headers: X-Requested-With,content-type
Access-Control-Allow-Credentials: true
Content-Type: text/html; charset=utf-8
Content-Length: 17
ETag: W/"11-r1/CHWaPL6hGmuxB/35BPX6Dqbg"
Date: Sun, 01 Oct 2023 07:37:03 GMT
Connection: keep-alive
Keep-Alive: timeout=5
```

Hello B10902068!

```
(base) enzhouang@Zenbook:~$ curl -i -X POST http://voip.csie.org:4071/submit -H "Content-Type: application/x-www-form-urlencoded" -d "username=B10902068&password=51"
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: null
Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, PATCH, DELETE
Access-Control-Allow-Headers: X-Requested-With,content-type
Access-Control-Allow-Credentials: true
Content-Type: text/html; charset=utf-8
Content-Length: 17
ETag: W/"11-r1/CHWaPL6hGmuxB/35BPX6Dqbg"
Date: Mon, 25 Sep 2023 08:52:29 GMT
Connection: keep-alive
Keep-Alive: timeout=5

Hello B10902068!
```

2. • Command :

```
curl -i -X POST http://voip.csie.org:4071/submit \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "username=B10902068&password=51&secret=CN"
```

• Response :

Hello B10902068. Your secret is "THE_dUe_Da7e_0F_@Ss19NmeNT_1_Is_Oct_4Th"


```
(base) enzohuang@Zenbook:~$ curl -i -X POST http://voip.csie.org:4071/submit -H "Content-Type: application/x-www-form-urlencoded" -d "username=B10902068&password=51&secret=CN"
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: null
Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, PATCH, DELETE
Access-Control-Allow-Headers: X-Requested-With,content-type
Access-Control-Allow-Credentials: true
Content-Type: text/html; charset=utf-8
Content-Length: 74
ETag: W/"4a++G7luFUpH43xNGi7jsgpudaElnA"
Date: Mon, 25 Sep 2023 08:59:10 GMT
Connection: keep-alive
Keep-Alive: timeout=5

Hello B10902068. Your secret is "THE_dUe_Da7e_0F_@Ss19NmeNT_1_Is_Oct_4Th"
```

6.

1. [Reference](#)

The traceroute command works by sending ICMP packets to routers that would be reached on the path towards destination. The routers would then return the packets to sender. After receiving the returned packets, the command would output the path to 8.8.8.8 and the time delay of each routers.

```
b10902068@ws1 [~] traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  140.112.30.254 (140.112.30.254)  9.103 ms  9.103 ms  9.235 ms
 2  140.112.149.121 (140.112.149.121)  0.404 ms  0.381 ms  0.364 ms
 3  140.112.0.238 (140.112.0.238)  0.302 ms  140.112.0.218 (140.112.0.218)  0.403 ms  0.380 ms
 4  140.112.0.206 (140.112.0.206)  1.164 ms  1.148 ms  1.481 ms
 5  140.112.0.34 (140.112.0.34)  1.217 ms  1.202 ms  1.186 ms
 6  72.14.196.229 (72.14.196.229)  4.663 ms  1.268 ms  6.382 ms
 7  108.170.244.33 (108.170.244.33)  1.665 ms  1.645 ms  108.170.244.65 (108.170.244.65)  2.215 ms
 8  142.251.77.87 (142.251.77.87)  1.682 ms  209.85.242.125 (209.85.242.125)  1.299 ms  209.85.245.65 (209.85.245.65)  1.999 ms
 9  dns.google (8.8.8.8)  1.345 ms  1.327 ms  1.312 ms
```

2. [Reference](#)

Instead of providing the information of the next router, only three star characters are presented in each hop after the fourth one.

The reason that cause `traceroute` to behave like this is that no router beyond 140.112.xxx.xxx has responded. A sequence of three-star hops indicates that the routers don't respond because their owners forbade them to.

```

b10902068@ws1 [~] traceroute 198.51.100.23
traceroute to 198.51.100.23 (198.51.100.23), 30 hops max, 60 byte packets
 1  140.112.30.254 (140.112.30.254)  1.882 ms  1.913 ms  2.051 ms
 2  140.112.149.121 (140.112.149.121)  0.418 ms  0.396 ms  0.375 ms
 3  140.112.0.238 (140.112.0.238)  0.271 ms  140.112.0.218 (140.112.0.218)  0.384 ms  140.112.0.238 (140.112.0.238)  0.277 ms
 4  140.112.0.206 (140.112.0.206)  1.071 ms  0.866 ms  1.026 ms
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *

```

7.

1. The IP address is 140.112.30.26

```

(base) enzohuang@Zenbook:~$ dig csie.ntu.edu.tw

; <<>> DiG 9.18.12-0ubuntu0.22.04.3-Ubuntu <<>> csie.ntu.edu.tw
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21843
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;csie.ntu.edu.tw.                IN      A

;; ANSWER SECTION:
csie.ntu.edu.tw.                600     IN      A      140.112.30.26

;; Query time: 31 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Sep 25 17:07:17 CST 2023
;; MSG SIZE rcvd: 60

```

2. [Reference](#)

Two of the IPs are 205.251.242.103 , 54.239.28.85 and 52.94.236.248

Using several different IP addresses for a single domain name is a technique to achieve

load balancing.

```
(base) enzo@huang@Zenbook:~$ dig amazon.com

; <<>> DiG 9.18.12-0ubuntu0.22.04.3-Ubuntu <<>> amazon.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58397
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;amazon.com.                IN      A

;; ANSWER SECTION:
amazon.com.                512     IN      A      205.251.242.103
amazon.com.                512     IN      A      52.94.236.248
amazon.com.                512     IN      A      54.239.28.85

;; Query time: 15 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Sep 25 17:07:47 CST 2023
;; MSG SIZE  rcvd: 87
```