

# Assignment 3

- **Name:** Arnab Sen
- **Roll:** 510519006
- **Gsuite:** [510519006.arnab@students.iests.ac.in](mailto:510519006.arnab@students.iests.ac.in)
- **Subject:** Computer Networks Lab (CS 3272)

## Question 1

Analyse the packets (across all layers) exchanged with your computer while executing the following commands:

1. ping
2. traceroute
3. dig
4. arp
5. wget

## Answer 1

### 1. ping

ping uses the ICMP protocol to send packets to a destination. Running the command `ping www.github.com` in a terminal and then capturing the packets we see something like this:

No.	Time	Source	Destination	Protocol	Length	Info
→ 42	0.349575...	192.168.1.4	13.234.210.38	ICMP	98	Echo (ping) request id=0xd51a, seq=9/2304, ttl=64 (reply in 44)
← 44	0.395857...	13.234.210.38	192.168.1.4	ICMP	98	Echo (ping) reply id=0xd51a, seq=9/2304, ttl=47 (request in 42)
142	1.351130...	192.168.1.4	13.234.210.38	ICMP	98	Echo (ping) request id=0xd51a, seq=10/2560, ttl=64 (reply in 145)
145	1.396458...	13.234.210.38	192.168.1.4	ICMP	98	Echo (ping) reply id=0xd51a, seq=10/2560, ttl=47 (request in 142)
281	2.352636...	192.168.1.4	13.234.210.38	ICMP	98	Echo (ping) request id=0xd51a, seq=11/2816, ttl=64 (reply in 288)
288	2.401311...	13.234.210.38	192.168.1.4	ICMP	98	Echo (ping) reply id=0xd51a, seq=11/2816, ttl=47 (request in 281)
408	3.354225...	192.168.1.4	13.234.210.38	ICMP	98	Echo (ping) request id=0xd51a, seq=12/3072, ttl=64 (reply in 424)
424	3.399421...	13.234.210.38	192.168.1.4	ICMP	98	Echo (ping) reply id=0xd51a, seq=12/3072, ttl=47 (request in 408)

→ Ethernet II, Src: IntelCor\_81:09:25 (24:ee:9a:81:09:25), Dst: Syrotech\_33:c3:d6 (7c:a9:6b:33:c3:d6)  
- Internet Protocol Version 4, Src: 192.168.1.4, Dst: 13.234.210.38  
    0100 .... = Version: 4  
    .... 0101 = Header Length: 20 bytes (5)  
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)  
    Total Length: 84  
    Identification: 0x7dff (32255)  
    Flags: 0x40, Don't fragment  
    Fragment Offset: 0  
    Time to Live: 64  
    Protocol: ICMP (1) TTL and Protocol  
    Header Checksum: 0xaed [validation disabled]  
    [Header checksum status: Unverified]  
    Source Address: 192.168.1.4 Source and Destination IPs  
    Destination Address: 13.234.210.38  
- Internet Control Message Protocol  
    Type: 8 (Echo (ping) request) Request Type  
    Code: 0  
    Checksum: 0xea0f [correct]  
    [Checksum Status: Good]  
    Identifier (BE): 54554 (0xd51a)  
    Identifier (LE): 6869 (0x1ad5)  
    Sequence Number (BE): 9 (0x0009)  
    Sequence Number (LE): 2304 (0x0900)  
    [Response frame: 44]  
    Timestamp from icmp data: Feb 1, 2022 15:29:31.000000000 IST  
    [Timestamp from icmp data (relative): 0.889616974 seconds]  
- Data (48 bytes)

- this is a request packet sent by the host to the destination.
- the Time To Live (TTL) is set to 64 .
- Protocol is ICMP .
- the packet is of ICMP type 8 (Echo Request) .

icmp						
No.	Time	Source	Destination	Protocol	Length	Info
42	0.349575...	192.168.1.4	13.234.210.38	ICMP	98	Echo (ping) request id=0xd51a, seq=9/2304, ttl=64 (reply in 44)
44	0.395857...	13.234.210.38	192.168.1.4	ICMP	98	Echo (ping) reply id=0xd51a, seq=9/2304, ttl=47 (request in 42)
142	1.351130...	192.168.1.4	13.234.210.38	ICMP	98	Echo (ping) request id=0xd51a, seq=10/2560, ttl=64 (reply in 145)
145	1.396458...	13.234.210.38	192.168.1.4	ICMP	98	Echo (ping) reply id=0xd51a, seq=10/2560, ttl=47 (request in 142)
281	2.352636...	192.168.1.4	13.234.210.38	ICMP	98	Echo (ping) request id=0xd51a, seq=11/2816, ttl=64 (reply in 288)
288	2.401311...	13.234.210.38	192.168.1.4	ICMP	98	Echo (ping) reply id=0xd51a, seq=11/2816, ttl=47 (request in 281)
408	3.354225...	192.168.1.4	13.234.210.38	ICMP	98	Echo (ping) request id=0xd51a, seq=12/3072, ttl=64 (reply in 424)
424	3.399421...	13.234.210.38	192.168.1.4	ICMP	98	Echo (ping) reply id=0xd51a, seq=12/3072, ttl=47 (request in 408)

```

> Ethernet II, Src: Syrotech_33:c3:d6 (7c:a9:6b:33:c3:d6), Dst: IntelCor_81:09:25 (24:ee:9a:81:09:25)
-> Internet Protocol Version 4, Src: 13.234.210.38, Dst: 192.168.1.4
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
-> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 84
  Identification: 0x7dff (32255)
-> Flags: 0x40, Don't fragment
  Fragment Offset: 0
  Time to Live: 47
  Protocol: ICMP (1)
  Header Checksum: 0x2bed [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 13.234.210.38
  Destination Address: 192.168.1.4
-> Internet Control Message Protocol
  Type: 0 (Echo (ping) reply)
  Code: 0
  Checksum: 0xf20f [correct]
  [Checksum Status: Good]
  Identifier (BE): 54554 (0xd51a)
  Identifier (LE): 6869 (0x1ad5)
  Sequence Number (BE): 9 (0x0009)
  Sequence Number (LE): 2304 (0x0900)
  [Request frame: 42]
  [Response time: 46.282 ms]
  Timestamp from icmp data: Feb 1, 2022 15:29:31.000000000 IST
  [Timestamp from icmp data (relative): 0.935898846 seconds]
```

- this is a reply packet sent by the destination to the host.
- the Time To Live (TTL) is set to 47 .
- Protocol is also ICMP .
- the packet is of ICMP type 0 (Echo Reply) .

Layer	Protocol Used
Internet	ICMP

**Note :** ICMP is not associated with a transport layer protocol such as TCP or UDP.

## 2. traceroute

The way traceroute works is by sending packets with increasing values of the Time To Live (TTL) to the destination.

No.	Time	Source	Destination	Protocol	Length	Info
345	3.359312...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=1/256, ttl=1 (no response found!)
346	3.359361...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=2/512, ttl=1 (no response found!)
347	3.359379...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=3/768, ttl=1 (no response found!)
348	3.359396...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=4/1024, ttl=2 (no response found!)
349	3.359412...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=5/1280, ttl=2 (no response found!)
350	3.359426...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=6/1536, ttl=2 (no response found!)
351	3.359443...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=7/1792, ttl=3 (no response found!)
352	3.359457...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=8/2048, ttl=3 (no response found!)
353	3.359473...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=9/2304, ttl=3 (no response found!)
354	3.359491...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=10/2560, ttl=4 (no response found!)
355	3.359505...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=11/2816, ttl=4 (no response found!)
356	3.359519...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=12/3072, ttl=4 (no response found!)
357	3.359535...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=13/3328, ttl=5 (no response found!)
358	3.359549...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=14/3584, ttl=5 (no response found!)
359	3.359563...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=15/3840, ttl=5 (no response found!)
360	3.359579...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=16/4096, ttl=6 (no response found!)
361	3.362227...	192.168.1.1	192.168.1.4	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
362	3.362346...	192.168.1.1	192.168.1.4	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
363	3.362456...	192.168.1.1	192.168.1.4	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
364	3.362558...	10.10.0.5	192.168.1.4	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
368	3.398762...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=17/4352, ttl=6 (no response found!)
369	3.398865...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=18/4668, ttl=6 (no response found!)
370	3.398889...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=19/4864, ttl=7 (no response found!)
371	3.398904...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=20/5120, ttl=7 (no response found!)
373	3.403501...	103.27.170.190	192.168.1.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
376	3.436715...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=21/5376, ttl=7 (no response found!)
377	3.436768...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=22/5632, ttl=8 (no response found!)
378	3.436785...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=23/5888, ttl=8 (no response found!)
379	3.436800...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=24/6144, ttl=8 (no response found!)
380	3.436817...	192.168.1.4	13.234.210.38	ICMP	74	Echo (ping) request id=0x640a, seq=25/6400, ttl=9 (no response found!)

Now, ideally we should see some Time to Live exceeded responses. So if we add that filter in wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
361	3.362227...	192.168.1.1	192.168.1.4	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
362	3.362346...	192.168.1.1	192.168.1.4	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
363	3.362456...	192.168.1.1	192.168.1.4	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
364	3.362558...	10.10.0.5	192.168.1.4	ICMP	102	Time-to-live exceeded (Time to live exceeded in transit)
373	3.403501...	103.27.170.190	192.168.1.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
382	3.439530...	52.95.66.156	192.168.1.4	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
384	3.441837...	103.27.170.190	192.168.1.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
385	3.441837...	103.27.170.190	192.168.1.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
394	3.468444...	103.10.208.13	192.168.1.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
398	3.476208...	52.95.66.156	192.168.1.4	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
399	3.478666...	52.95.64.191	192.168.1.4	ICMP	186	Time-to-live exceeded (Time to live exceeded in transit)
400	3.481110...	52.95.64.191	192.168.1.4	ICMP	186	Time-to-live exceeded (Time to live exceeded in transit)
401	3.481161...	99.83.76.135	192.168.1.4	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
773	7.574582...	99.83.76.135	192.168.1.4	ICMP	182	Time-to-live exceeded (Time to live exceeded in transit)
807	7.733398...	99.83.76.142	192.168.1.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
13...	12.71859...	13.234.210.38	192.168.1.4	ICMP	74	Echo (ping) reply id=0x640a, seq=52/13312, ttl=47 (request in 1324)

The IPs we see in the Source field exactly match with the IPs we see in the traceroute command.

Since I used the traceroute with -I flag, it uses ICMP protocol.

Layer	Protocol Used
Internet	ICMP

### 3. dig

We know dig command uses the DNS protocol, so filtering the packets in wireshark we see:

No.	Time	Source	Destination	Protocol	Length	Info
69	0.811073...	192.168.1.4	10.10.0.1	DNS	97	Standard query 0xa851 A www.github.com OPT
74	0.846241...	10.10.0.1	192.168.1.4	DNS	115	Standard query response 0xa851 A www.github.com CNAME github.com A 13.234.210.38 OPT
184	1.654786...	192.168.1.4	10.10.0.1	DNS	82	Standard query 0x8580 A az764295.vo.msecnd.net
189	1.716548...	10.10.0.1	192.168.1.4	DNS	127	Standard query response 0x8580 A az764295.vo.msecnd.net CNAME cs22.wpc.v0cdn.net A 117.18.232.200

Analysing the request:

dns						
No.	Time	Source	Destination	Protocol	Length	Info
69	0.811073...	192.168.1.4	10.10.0.1	DNS	97	Standard query 0xa851 A www.github.com OPT
74	0.846241...	10.10.0.1	192.168.1.4	DNS	115	Standard query response 0xa851 A www.github.com CNAME github.com A 13.234.210.38 OPT
184	1.654780...	192.168.1.4	10.10.0.1	DNS	82	Standard query 0x8580 A az764295.vo.msecnd.net
189	1.716548...	10.10.0.1	192.168.1.4	DNS	127	Standard query response 0x8580 A az764295.vo.msecnd.net CNAME cs22.wpc.v0cdn.net A 117.18.232.200
Ethernet II, Src: IntelCor_81:09:25 (24:ee:9a:81:09:25), Dst: Syrotech_33:c3:d6 (7c:a9:6b:33:c3:d6)						
Internet Protocol Version 4, Src: 192.168.1.4, Dst: 10.10.0.1						
0100 .... = Version: 4 .... 0101 = Header Length: 20 bytes (5) Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) Total Length: 83 Identification: 0x68cb (26827) Flags: 0x00 Fragment Offset: 0 Time to Live: 64 Protocol: UDP (17) Header Checksum: 0x4618 [validation disabled] [Header checksum status: Unverified] Source Address: 192.168.1.4 Destination Address: 10.10.0.1						
User Datagram Protocol, Src Port: 43713, Dst Port: 53 Source Port: 43713 Destination Port: 53						
Length: 63 Checksum: 0xcc07 [unverified] [Checksum Status: Unverified] [Stream index: 3] [Timestamps] UDP payload (55 bytes)						
Domain Name System (query) Transaction ID: 0xa851 Flags: 0x0120 Standard query Questions: 1 Answer RRs: 0 Authority RRs: 0 Additional RRs: 1 Queries www.github.com: type A, class IN						
[Response In: 74]						

- the query was made to the DNS server 10.10.0.1 .
- Query was for www.github.com for A record.

Analysing the response:

dns						
No.	Time	Source	Destination	Protocol	Length	Info
69	0.811073...	192.168.1.4	10.10.0.1	DNS	97	Standard query 0xa851 A www.github.com OPT
74	0.846241...	10.10.0.1	192.168.1.4	DNS	115	Standard query response 0xa851 A www.github.com CNAME github.com A 13.234.210.38 OPT
184	1.654780...	192.168.1.4	10.10.0.1	DNS	82	Standard query 0x8580 A az764295.vo.msecnd.net
189	1.716548...	10.10.0.1	192.168.1.4	DNS	127	Standard query response 0x8580 A az764295.vo.msecnd.net CNAME cs22.wpc.v0cdn.net A 117.18.232.200
[Stream index: 3] [Timestamps] UDP payload (73 bytes)						
Domain Name System (response) Transaction ID: 0xa851 Flags: 0x8180 Standard query response, No error Questions: 1 Answer RRs: 2 Authority RRs: 0 Additional RRs: 1 Queries www.github.com: type A, class IN Name: www.github.com [Name Length: 14] [Label Count: 3] Type: A (Host Address) (1) Class: IN (0x0001) Answers www.github.com: type CNAME, class IN, cname github.com Name: www.github.com Type: CNAME (Canonical NAME for an alias) (5) Class: IN (0x0001) Time to live: 3600 (1 hour) Data length: 2 CNAME: github.com						
github.com: type A, class IN, addr 13.234.210.38 Name: github.com Type: A (Host Address) (1) Class: IN (0x0001) Time to live: 60 (1 minute) Data length: 4 Address: 13.234.210.38						
Additional records [Request In: 69] [Time: 0.035167700 seconds]						

- In the answers we received a CNAME record for www.github.com . Basically saying both www.github.com and github.com are the same.

- Then received a A record for `github.com` with address `13.234.210.38`.
- Protocol used is `DNS`.

Layer	Protocol Used
Application	DNS
Transport	UDP
Internet	IP

## 4. arp

`arp` maintains a table of mac address and their corresponding IP address. So, when we run the `arp` command, it sends a broadcast packet to all the hosts in the network. Analyzing the packets we see:

```

No. Time           Source          Destination        Protocol Length Info
170 18.50732...  Syrotech_33:c... Broadcast      ARP       60 ARP Announcement for 192.168.1.1

Frame 170: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface wlan0, id 0
Ethernet II, Src: Syrotech_33:c3:d6 (7c:a9:6b:33:c3:d6), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Source: Syrotech_33:c3:d6 (7c:a9:6b:33:c3:d6)
  Type: ARP (0x0806)
  Padding: 0000000000000000000000000000000000000000000000000000000000000000
Address Resolution Protocol (ARP Announcement)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  [Is gratuitous: True]
  [Is announcement: True]
  Sender MAC address: Syrotech_33:c3:d6 (7c:a9:6b:33:c3:d6)
  Sender IP address: 192.168.1.1
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.1.1

```

- Destination: Broadcast (`ff:ff:ff:ff:ff:ff`)
- Protocol is `ARP`

Layer	Protocol Used
Internet	ARP

## 5. wget

`wget` makes a GET HTTP(S) request to the server and then downloads the file. So, we can see the packets sent by the host to the server:

No.	Time	Source	Destination	Protocol	Length	Info
19	0.838392...	192.168.1.4	13.234.176.102	TCP	74	41972 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2710269459 TSecr=0 WS=128
20	0.884957...	13.234.176.102	192.168.1.4	TCP	74	443 → 41972 [SYN, ACK] Seq=1 Ack=1 Win=65535 Len=0 MSS=1436 SACK_PERM=1 TSval=1126753271 TSecr=2710269459 WS=1024
21	0.885025...	192.168.1.4	13.234.176.102	TCP	66	41972 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2710269506 TSecr=1126753271
22	0.885870...	192.168.1.4	13.234.176.102	TLSV...	583	Client Hello
23	0.952184...	13.234.176.102	192.168.1.4	TLSV...	1490	Server Hello, Change Cipher Spec, Application Data
24	0.952245...	192.168.1.4	13.234.176.102	TCP	66	41972 → 443 [ACK] Seq=518 Ack=1425 Win=63104 Len=0 TSval=2710269573 TSecr=1126753323
25	0.953096...	192.168.1.4	13.234.176.102	TLSV...	72	Change Cipher Spec
26	0.955850...	13.234.176.102	192.168.1.4	TLSV...	1395	Application Data, Application Data, Application Data
27	0.955889...	192.168.1.4	13.234.176.102	TCP	66	41972 → 443 [ACK] Seq=524 Ack=2754 Win=63104 Len=0 TSval=2710269577 TSecr=1126753323
28	1.049476...	13.234.176.102	192.168.1.4	TCP	66	443 → 41972 [ACK] Seq=2754 Ack=524 Win=67584 Len=0 TSval=1126753434 TSecr=2710269574
29	1.049526...	192.168.1.4	13.234.176.102	TLSV...	124	Application Data
30	1.097643...	13.234.176.102	192.168.1.4	TCP	66	443 → 41972 [ACK] Seq=2754 Ack=582 Win=67584 Len=0 TSval=1126753482 TSecr=2710269670
31	1.097696...	192.168.1.4	13.234.176.102	TLSV...	211	Application Data
32	1.097643...	13.234.176.102	192.168.1.4	TLSV...	145	Application Data
33	1.097743...	192.168.1.4	13.234.176.102	TCP	66	41972 → 443 [ACK] Seq=727 Ack=2833 Win=64128 Len=0 TSval=2710269719 TSecr=1126753482
34	1.097643...	13.234.176.102	192.168.1.4	TLSV...	145	Application Data
35	1.097775...	192.168.1.4	13.234.176.102	TCP	66	41972 → 443 [ACK] Seq=727 Ack=2912 Win=64128 Len=0 TSval=2710269719 TSecr=1126753482
36	1.170435...	13.234.176.102	192.168.1.4	TLSV...	145	Application Data
37	1.170495...	192.168.1.4	13.234.176.102	TCP	66	41972 → 443 [ACK] Seq=727 Ack=4304 Win=63104 Len=0 TSval=2710269791 TSecr=1126753533
38	1.170435...	13.234.176.102	192.168.1.4	TLSV...	1458	Application Data
39	1.170537...	192.168.1.4	13.234.176.102	TCP	66	41972 → 443 [ACK] Seq=727 Ack=5696 Win=61824 Len=0 TSval=2710269791 TSecr=1126753533
40	1.170436...	13.234.176.102	192.168.1.4	TLSV...	234	Application Data
41	1.170556...	192.168.1.4	13.234.176.102	TCP	66	41972 → 443 [ACK] Seq=727 Ack=5864 Win=61696 Len=0 TSval=2710269791 TSecr=1126753533
42	1.171703...	13.234.176.102	192.168.1.4	TLSV...	1458	Application Data
43	1.171741...	192.168.1.4	13.234.176.102	TCP	66	41972 → 443 [ACK] Seq=727 Ack=7256 Win=63104 Len=0 TSval=2710269793 TSecr=1126753533
44	1.179712...	13.234.176.102	192.168.1.4	TLSV...	1490	Application Data

```

Frame 19: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface wlan0, id 0
Ethernet II, Src: IntelCor_81:09:25 (24:ee:9a:81:09:25), Dst: Syrotech_33:c3:d6 (7c:a9:6b:33:c3:d6)
  Destination: Syrotech_33:c3:d6 (7c:a9:6b:33:c3:d6)
    Address: Syrotech_33:c3:d6 (7c:a9:6b:33:c3:d6)
      ... .0. .... .... .... = LG bit: Globally unique address (factory default)
      ... .0. .... .... .... = IG bit: Individual address (unicast)
  Source: IntelCor_81:09:25 (24:ee:9a:81:09:25)
    Address: IntelCor_81:09:25 (24:ee:9a:81:09:25)
      ... .0. .... .... .... = LG bit: Globally unique address (factory default)
      ... .0. .... .... .... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.1.4, Dst: 13.234.176.102
Transmission Control Protocol, Src Port: 41972, Dst Port: 443, Seq: 0, Len: 0

```

- First packet was sent with `SYN` flag set.
- Second packet received with `SYN-ACK` flag set.
- Finally, a third packet was sent with `ACK` flag set.
- Now, once the connection was setup, data transfer took place from the server to my machine.

Looking at a packet of Data transfer we see:

Connection Setup

Data Transfer

- most of the data is all gibberish, because it has been end-to-end encrypted. The request was made through an `HTTPS` protocol which makes it impossible for sniffers like wireshark to decode the data.
  - Application Layer protocol: `HTTPS` .
  - Transport Layer protocol: `TCP` .

Layer	Protocol Used
Application	HTTPS
Transport	TCP
Internet	IP

## Question 2

Capture the packets while sending/receiving telnet request/response between your computer and a custom server running the telnet daemon. What is your observation while analysing the application layer data?

## Answer 2

I have set up a telnet daemon in a virtual machine which is in bridged network mode. My VM's IP is 192.168.1.3 . Since I don't have any service running on port 23 (telnet), started a netcat server on port 23 with nc -lvp 23 .

Making a telnet request to the server by the command:

telnet 192.168.1.3 23

No.	Time	Source	Destination	Protocol	Length	Info
5	0.613522...	192.168.1.4	192.168.1.3	TCP	74	56534 -> 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TStamp=912293413 TSectr=0 WS=128
6	0.614060...	192.168.1.3	192.168.1.4	TCP	74	23 -> 56534 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TStamp=3677714773 TSectr=912293413 WS=128
7	0.614136...	192.168.1.4	192.168.1.3	TCP	66	56534 -> 23 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TStamp=912293414 TSectr=3677714773
46	3.183675...	192.168.1.4	192.168.1.3	TELNET	73	Telnet Data ...
47	3.184285...	192.168.1.3	192.168.1.4	TCP	66	23 -> 56534 [ACK] Seq=1 Ack=8 Win=65280 Len=0 TStamp=3677717345 TSectr=912295983
92	5.103719...	192.168.1.4	192.168.1.3	TELNET	73	Telnet Data ...
93	5.104475...	192.168.1.3	192.168.1.4	TCP	66	23 -> 56534 [ACK] Seq=1 Ack=15 Win=65280 Len=0 TStamp=3677719266 TSectr=912297903
136	9.943716...	192.168.1.4	192.168.1.3	TELNET	77	Telnet Data ...
137	9.944287...	192.168.1.3	192.168.1.4	TCP	66	23 -> 56534 [ACK] Seq=1 Ack=26 Win=65280 Len=0 TStamp=3677724108 TSectr=912307243
200	13.67179...	192.168.1.4	192.168.1.3	TELNET	82	Telnet Data ...
201	13.67236...	192.168.1.3	192.168.1.4	TCP	66	23 -> 56534 [ACK] Seq=1 Ack=42 Win=65280 Len=0 TStamp=3677735803 TSectr=912306471
269	21.63385...	192.168.1.3	192.168.1.4	TCP	66	56534 -> 23 [FIN, ACK] Seq=42 Ack=2 Win=64256 Len=0 TStamp=912314433 TSectr=3677735803
270	21.63405...	192.168.1.4	192.168.1.3	TCP	66	23 -> 56534 [ACK] Seq=42 Ack=2 Win=64256 Len=0 TStamp=912314433 TSectr=3677735803
271	21.63444...	192.168.1.3	192.168.1.4	TCP	66	23 -> 56534 [ACK] Seq=2 Ack=43 Win=65280 Len=0 TStamp=3677735804 TSectr=912314433

Frame 5: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface wlan0, id 0  
Ethernet II, Src: 24:ee:9a:81:09:25, Dst: 08:00:27:be:20:60  
Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.3  
Transmission Control Protocol, Src Port: 56534, Dst Port: 23, Seq: 0, Len: 0

Connection Established

```
0000  08 00 27 be 20 60 24 ee 9a 81 09 25 08 00 45 10  .`$%E
0010  00 3c 5a 01 40 00 40 06 5d 53 c0 a8 01 04 c0 a8  -<@]s...
0020  01 03 dc d6 00 17 07 25 fe cc 00 00 00 00 a0 02  ....%
0030  fa f0 83 86 00 02 04 05 b4 04 02 08 0a 36 60  ..6`...
0040  7e 25 00 00 00 00 01 03 03 07  ~%.....
```

It first uses TCP protocol to establish connection. My machine first sends a packet with SYN flag to which the server responds with SYN-ACK. And then finally my machine sends a packet with ACK flag to the server, establishing the connection.

Now, after that when the data transfer happens it uses the TELNET protocol. So, I can see the following packets:

No.	Time	Source	Destination	Protocol	Length	Info
5	0.613522...	192.168.1.4	192.168.1.3	TCP	74	56534 -> 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TStamp=912293413 TSectr=0 WS=128
6	0.614060...	192.168.1.3	192.168.1.4	TCP	74	23 -> 56534 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TStamp=3677714773 TSectr=912293413 WS=128
7	0.614136...	192.168.1.4	192.168.1.3	TCP	66	56534 -> 23 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TStamp=912293414 TSectr=3677714773
46	3.183675...	192.168.1.4	192.168.1.3	TELNET	73	Telnet Data ...
47	3.184285...	192.168.1.3	192.168.1.4	TCP	66	23 -> 56534 [ACK] Seq=1 Ack=8 Win=65280 Len=0 TStamp=3677717345 TSectr=912295983
92	5.103719...	192.168.1.4	192.168.1.3	TELNET	73	Telnet Data ...
93	5.104475...	192.168.1.3	192.168.1.4	TCP	66	23 -> 56534 [ACK] Seq=1 Ack=15 Win=65280 Len=0 TStamp=3677719266 TSectr=912297903
136	9.943716...	192.168.1.4	192.168.1.3	TELNET	77	Telnet Data ...
137	9.944287...	192.168.1.3	192.168.1.4	TCP	66	23 -> 56534 [ACK] Seq=1 Ack=26 Win=65280 Len=0 TStamp=3677724108 TSectr=912307243
200	13.67179...	192.168.1.4	192.168.1.3	TELNET	82	Telnet Data ...
201	13.67236...	192.168.1.3	192.168.1.4	TCP	66	23 -> 56534 [ACK] Seq=1 Ack=42 Win=65280 Len=0 TStamp=3677727838 TSectr=912306471
269	21.63385...	192.168.1.3	192.168.1.4	TCP	66	23 -> 56534 [FIN, ACK] Seq=42 Ack=2 Win=65280 Len=0 TStamp=912314433 TSectr=3677735803
270	21.63405...	192.168.1.4	192.168.1.3	TCP	66	56534 -> 23 [FIN, ACK] Seq=42 Ack=2 Win=64256 Len=0 TStamp=912314433 TSectr=3677735803
271	21.63444...	192.168.1.3	192.168.1.4	TCP	66	23 -> 56534 [ACK] Seq=2 Ack=43 Win=65280 Len=0 TStamp=3677735804 TSectr=912314433

Frame 200: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface wlan0, id 0  
Ethernet II, Src: 24:ee:9a:81:09:25, Dst: 08:00:27:be:20:60  
Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.3  
Transmission Control Protocol, Src Port: 56534, Dst Port: 23, Seq: 26, Ack: 1, Len: 16  
Telnet

telnet protocol used to send data

```
0000  08 00 27 be 20 60 24 ee 9a 81 09 25 08 00 45 10  .`$%E
0010  00 44 5a 06 40 00 40 06 5d 46 c0 a8 01 04 c0 a8  -DZ@]F...
0020  01 03 dc d6 00 17 07 25 fe c6 3c a5 ba 82 80 18  ....%
0030  01 f6 83 8e 00 00 01 01 08 a0 36 60 b1 27 db 35  ..6'5
0040  9d cc 73 65 63 72 65 74 20 6d 65 73 73 61 67 65  secret message
0050  0d 0a
```

Data is not encrypted

One thing to note is that the data transfer is not encrypted, and anyone sniffing the network can easily decode the data.

## Question 3

Capture the packets while sending/receiving ssh request/response between your computer and one of the department servers. What is your observation while analysing the application layer data?

## Answer 3

SSH uses asymmetric cryptography to establish a shared secret key and then symmetric cryptography for bulk encryption with that key.

So, the first step after initial connection is to exchange the public key. Here we can see two packets that are involved in the key exchange, we can also see the respective algorithms used.

ip.addr==192.168.1.4 && tcp.port==50314 && ip.addr==192.168.1.2 && tcp.port==22

No.	Time	Source	Destination	Protocol	Length	Info
21	0.805291...	192.168.1.4	192.168.1.2	TCP	76	50314 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TStamp=2989130834 TSecr=0 WS=128
22	0.805554...	192.168.1.4	192.168.1.4	TCP	76	22 → 50314 [SYN, ACK] Seq=1 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TStamp=3885192828 TSecr=2989130834 WS=128
23	0.805570...	192.168.1.4	192.168.1.2	TCP	68	50314 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TStamp=2989130834 TSecr=3885192828
24	0.805836...	192.168.1.4	192.168.1.2	SSHv2	100	Client: Protocol (SSH-2.0-OpenSSH_8.4p1 Debian-6)
25	0.805981...	192.168.1.4	192.168.1.2	TCP	68	22 → 50314 [ACK] Seq=1 Ack=33 Win=65152 Len=0 TStamp=3885192828 TSecr=2989130835
29	0.816168...	192.168.1.2	192.168.1.4	SSHv2	100	Server: Protocol (SSH-2.0-OpenSSH_8.7p1 Debian-2)
30	0.816192...	192.168.1.4	192.168.1.2	TCP	68	50314 → 22 [ACK] Seq=33 Ack=33 Win=64256 Len=0 TStamp=2989130845 TSecr=3885192839
31	0.816688...	192.168.1.4	192.168.1.2	SSHv2	1580	<b>Client: Key Exchange Init</b>
32	0.817040...	192.168.1.2	192.168.1.4	TCP	68	22 → 50314 [ACK] Seq=33 Ack=1545 Win=63872 Len=0 TStamp=3885192840 TSecr=2989130846
33	0.817188...	192.168.1.2	192.168.1.4	SSHv2	1124	<b>Server: Key Exchange Init</b>
34	0.817195...	192.168.1.4	192.168.1.2	TCP	68	50314 → 22 [ACK] Seq=1545 Ack=1089 Win=64128 Len=0 TStamp=2989130846 TSecr=3885192840
35	0.819037...	192.168.1.4	192.168.1.2	SSHv2	116	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
36	0.819182...	192.168.1.2	192.168.1.4	TCP	68	22 → 50314 [ACK] Seq=1089 Ack=1593 Win=64128 Len=0 TStamp=3885192842 TSecr=2989130848
37	0.823745...	192.168.1.2	192.168.1.4	SSHv2	624	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=276)
38	0.823760...	192.168.1.4	192.168.1.2	TCP	68	50314 → 22 [ACK] Seq=1593 Ack=1645 Win=64128 Len=0 TStamp=2989130853 TSecr=3885192846
39	0.826652...	192.168.1.4	192.168.1.2	SSHv2	84	Client: New Keys
40	0.826805...	192.168.1.2	192.168.1.4	TCP	68	22 → 50314 [ACK] Seq=1645 Ack=1609 Win=64128 Len=0 TStamp=3885192849 TSecr=2989130855

Frame 31: 1580 bytes on wire (12640 bits), 1580 bytes captured (12640 bits) on interface any, id 0

- Linux cooked capture v1
- Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.2
- Transmission Control Protocol, Src Port: 50314, Dst Port: 22, Seq: 33, Ack: 33, Len: 1512
- SSH Protocol
  - SSH Version 2 (encryption:chacha20-poly1305@openssh.com mac:<implicit> compression:none)
    - Packet Length: 1508
    - Padding Length: 10
  - **Key Exchange (method:curve25519-sha256)**
    - Message Code: Key Exchange Init (20)
    - Algorithms
      - Cookie: 5945f0c0b161eaeeaa1add15921ce7b88
      - **kek\_algorithms** length: 241
      - **kek\_algorithms** string [truncated]: curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,server\_host\_key\_algorithms length: 500
      - **server\_host\_key\_algorithms** string [truncated]: ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,sk-ecdsa-public-v01@openssh.com
      - **encryption\_algorithms\_client\_to\_server** length: 108
      - **encryption\_algorithms\_client\_to\_server** string: chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com
      - **encryption\_algorithms\_server\_to\_client** length: 108
      - **encryption\_algorithms\_server\_to\_client** string: chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com
      - **mac\_algorithms\_client\_to\_server** length: 213
      - **mac\_algorithms\_client\_to\_server** string [truncated]: umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com
      - **mac\_algorithms\_server\_to\_client** length: 213
      - **mac\_algorithms\_server\_to\_client** string [truncated]: umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com

**Key exchanges**

After this, the server is verified and both the parties negotiate a session key using a version of something called the Diffie-Hellman algorithm. This algorithm is designed in such a way that both the parties contribute equally in generation of session key. The generated session key is shared symmetric key i.e. the same key is used for encryption and decryption.

No.	Time	Source	Destination	Protocol	Length	Info
21	0.805291...	192.168.1.4	192.168.1.2	TCP	76	50314 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TStamp=2989130834 TSecr=0 WS=128
22	0.805554...	192.168.1.2	192.168.1.4	TCP	76	22 → 50314 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TStamp=3885192828 TSecr=2989130834 WS=128
23	0.805570...	192.168.1.4	192.168.1.2	TCP	68	50314 → 22 [ACK] Seq=1 Ack=1 Win=64242 Len=0 TStamp=2989130834 TSecr=3885192828
24	0.805836...	192.168.1.4	192.168.1.2	SSHv2	100	Client: Protocol (SSH-2.0-OpenSSH_8.4p1 Debian-6)
25	0.805981...	192.168.1.2	192.168.1.4	TCP	68	62 → 50314 [ACK] Seq=1 Ack=33 Win=65152 Len=0 TStamp=2989130834 TSecr=3885192828 TSecr=2989130835
29	0.816186...	192.168.1.2	192.168.1.4	SSHv2	100	Server: Protocol (SSH-2.0-OpenSSH_8.7p1 Debian-2)
30	0.816202...	192.168.1.4	192.168.1.2	TCP	68	50314 → 22 [ACK] Seq=33 Ack=33 Win=64256 Len=0 TStamp=2989130845 TSecr=3885192839
31	0.816888...	192.168.1.4	192.168.1.2	SSHv2	1580	Client: Key Exchange Init
32	0.817040...	192.168.1.2	192.168.1.4	TCP	68	62 → 50314 [ACK] Seq=1545 Ack=1545 Win=63872 Len=0 TStamp=3885192840 TSecr=2989130846
33	0.817188...	192.168.1.2	192.168.1.4	SSHv2	1124	Server: Key Exchange Init
34	0.817195...	192.168.1.4	192.168.1.2	TCP	68	50314 → 22 [ACK] Seq=1545 Ack=1089 Win=64128 Len=0 TStamp=2989130846 TSecr=3885192840
35	0.819037...	192.168.1.4	192.168.1.2	SSHv2	116	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
36	0.819182...	192.168.1.2	192.168.1.4	TCP	68	22 → 50314 [ACK] Seq=1089 Ack=1593 Win=64128 Len=0 TStamp=3885192842 TSecr=2989130848
37	0.823745...	192.168.1.2	192.168.1.4	SSHv2	624	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=276)
38	0.823760...	192.168.1.4	192.168.1.2	TCP	68	50314 → 22 [ACK] Seq=1593 Ack=1645 Win=64128 Len=0 TStamp=2989130853 TSecr=3885192846
39	0.826652...	192.168.1.4	192.168.1.2	SSHv2	84	Client: New Keys
40	0.826805...	192.168.1.2	192.168.1.4	TCP	68	62 → 50314 [ACK] Seq=1645 Ack=1609 Win=64128 Len=0 TStamp=3885192849 TSecr=2989130855
41	0.827008...	192.168.1.4	192.168.1.2	SSHv2	112	Client: Encrypted packet (len=44)
42	0.827122...	192.168.1.2	192.168.1.4	TCP	68	22 → 50314 [ACK] Seq=1645 Ack=1653 Win=64128 Len=0 TStamp=3885192850 TSecr=2989130856
43	0.827239...	192.168.1.2	192.168.1.4	SSHv2	112	Server: Encrypted packet (len=44)
44	0.827245...	192.168.1.4	192.168.1.2	TCP	68	50314 → 22 [ACK] Seq=1653 Ack=1689 Win=64128 Len=0 TStamp=2989130856 TSecr=3885192850
45	0.827303...	192.168.1.4	192.168.1.2	SSHv2	128	Client: Encrypted packet (len=60)

No.	Time	Source	Destination	Protocol	Length	Info
21	0.805291...	192.168.1.4	192.168.1.2	TCP	76	50314 -> 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2989130834 TSecr=0 WS=128
22	0.805554...	192.168.1.2	192.168.1.4	TCP	76	22 -> 50314 [SYN, ACK] Seq=1 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=3885192828 TSecr=2989130834 WS=128
23	0.695570...	192.168.1.4	192.168.1.2	TCP	68	50314 -> 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2989130834 TSecr=3885192828
24	0.805836...	192.168.1.4	192.168.1.2	SSHv2	100	Client: Protocol [SSH-2.0-OpenSSH_8.4pi Debian-6]
25	0.805981...	192.168.1.2	192.168.1.4	TCP	68	22 -> 50314 [ACK] Seq=1 Ack=3 Win=65152 Len=0 TSval=3885192828 TSecr=2989130835
26	0.816186...	192.168.1.2	192.168.1.4	SSHv2	100	Server: Protocol [SSH-2.0-OpenSSH_8.7pi Debian-2]
30	0.816202...	192.168.1.4	192.168.1.2	TCP	68	50314 -> 22 [ACK] Seq=33 Ack=33 Win=64256 Len=0 TSval=2989130845 TSecr=3885192839
31	0.816888...	192.168.1.4	192.168.1.2	SSHv2	1580	Client: Key Exchange Init
32	0.817040...	192.168.1.2	192.168.1.4	TCP	68	22 -> 50314 [ACK] Seq=33 Ack=1545 Win=63872 Len=0 TSval=3885192840 TSecr=2989130846
33	0.817188...	192.168.1.2	192.168.1.4	SSHv2	1124	Server: Key Exchange Init
34	0.817195...	192.168.1.4	192.168.1.2	TCP	68	50314 -> 22 [ACK] Seq=1545 Ack=1089 Win=64128 Len=0 TSval=2989130846 TSecr=3885192840
35	0.819637...	192.168.1.4	192.168.1.2	SSHv2	116	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
36	0.819182...	192.168.1.2	192.168.1.4	TCP	68	22 -> 50314 [ACK] Seq=1089 Ack=1593 Win=64128 Len=0 TSval=3885192842 TSecr=2989130848
37	0.823745...	192.168.1.2	192.168.1.4	SSHv2	624	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=276)
38	0.823760...	192.168.1.4	192.168.1.2	TCP	68	50314 -> 22 [ACK] Seq=1593 Ack=1645 Win=64128 Len=0 TSval=2989130853 TSecr=3885192846
39	0.826652...	192.168.1.4	192.168.1.2	SSHv2	84	Client: New Keys
40	0.826805...	192.168.1.2	192.168.1.4	TCP	68	22 -> 50314 [ACK] Seq=1645 Ack=1609 Win=64128 Len=0 TSval=3885192849 TSecr=2989130855

```

> Frame 35: 116 bytes on wire (928 bits), 116 bytes captured (928 bits) on interface any, id 0
> Linux cooked capture v1
> Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.2
> Transmission Control Protocol, Src Port: 50314, Dst Port: 22, Seq: 1545, Ack: 1089, Len: 48
> SSH Protocol
  > SSH Version 2 (encryption:chacha20-poly1305@openssh.com mac:<implicit> compression:none)
    Packet Length: 44
    Padding Length: 6
    > Key Exchange (method:curve25519-sha256)
      Message Code: Elliptic Curve Diffie-Hellman Key Exchange Init (30)
      ECDH client's ephemeral public key length: 32
      ECDH client's ephemeral public key (Q_C): 1f85cee244681f9b4b9a62b7aec6b6991a76db5e15965c6c0fa0aa79ab25aa2c
      Padding String: 000000000000
      [Direction: client-to-server]

```

Encryption algorithm

Once, that is done the rest of the packets are then sent encrypted.

No.	Time	Source	Destination	Protocol	Length	Info
45	0.827303...	192.168.1.4	192.168.1.2	SSHv2	128	Client: Encrypted packet (len=60)
46	0.827365...	192.168.1.2	192.168.1.4	TCP	68	22 -> 50314 [ACK] Seq=1689 Ack=1713 Win=64128 Len=0 TSval=3885192850 TSecr=2989130856
47	0.837083...	192.168.1.2	192.168.1.4	SSHv2	120	Server: Encrypted packet (len=52)
48	0.837124...	192.168.1.4	192.168.1.2	TCP	68	50314 -> 22 [ACK] Seq=1713 Ack=1741 Win=64128 Len=0 TSval=2989130866 TSecr=3885192859
49	0.837359...	192.168.1.4	192.168.1.2	SSHv2	696	Client: Encrypted packet (len=628)
50	0.837704...	192.168.1.2	192.168.1.4	TCP	68	22 -> 50314 [ACK] Seq=1741 Ack=2341 Win=64128 Len=0 TSval=3885192860 TSecr=2989130866
52	0.843357...	192.168.1.2	192.168.1.4	SSHv2	120	Server: Encrypted packet (len=52)
53	0.843583...	192.168.1.4	192.168.1.2	SSHv2	208	Client: Encrypted packet (len=140)
54	0.843942...	192.168.1.2	192.168.1.4	TCP	68	22 -> 50314 [ACK] Seq=1793 Ack=2481 Win=64128 Len=0 TSval=3885192866 TSecr=2989130872
55	0.851386...	192.168.1.2	192.168.1.4	SSHv2	120	Server: Encrypted packet (len=52)
56	0.851655...	192.168.1.4	192.168.1.2	SSHv2	440	Client: Encrypted packet (len=372)
57	0.852014...	192.168.1.2	192.168.1.4	TCP	68	22 -> 50314 [ACK] Seq=1845 Ack=2853 Win=64128 Len=0 TSval=3885192874 TSecr=2989130880
58	0.857871...	192.168.1.2	192.168.1.4	SSHv2	120	Server: Encrypted packet (len=52)
60	0.900209...	192.168.1.4	192.168.1.2	TCP	68	50314 -> 22 [ACK] Seq=2853 Ack=1897 Win=64128 Len=0 TSval=2989130929 TSecr=3885192880
86	2.885536...	192.168.1.4	192.168.1.2	SSHv2	152	Client: Encrypted packet (len=84)
87	2.886045...	192.168.1.2	192.168.1.4	TCP	68	22 -> 50314 [ACK] Seq=1897 Ack=2937 Win=64128 Len=0 TSval=3885194908 TSecr=2989132914
89	2.979247...	192.168.1.2	192.168.1.4	SSHv2	96	Server: Encrypted packet (len=28)
90	2.979270...	192.168.1.4	192.168.1.2	TCP	68	50314 -> 22 [ACK] Seq=2937 Ack=1925 Win=64128 Len=0 TSval=2989133008 TSecr=3885195002
91	2.979439...	192.168.1.4	192.168.1.2	SSHv2	180	Client: Encrypted packet (len=112)
92	2.979574...	192.168.1.2	192.168.1.4	TCP	68	22 -> 50314 [ACK] Seq=1925 Ack=3049 Win=64128 Len=0 TSval=3885195002 TSecr=2989133008
94	3.0006459...	192.168.1.2	192.168.1.4	SSHv2	696	Server: Encrypted packet (len=628)
95	3.0006480...	192.168.1.4	192.168.1.2	TCP	68	50314 -> 22 [ACK] Seq=3049 Ack=2553 Win=64128 Len=0 TSval=2989133035 TSecr=3885195029

```

> Frame 39: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface any, id 0
> Linux cooked capture v1
> Internet Protocol Version 4, Src: 192.168.1.4, Dst: 192.168.1.2
> Transmission Control Protocol, Src Port: 50314, Dst Port: 22, Seq: 1593, Ack: 1645, Len: 16
> SSH Protocol
  > SSH Version 2 (encryption:chacha20-poly1305@openssh.com mac:<implicit> compression:none)
    Packet Length: 12
    Padding Length: 10
    > Key Exchange (method:curve25519-sha256)
      Message Code: New Keys (21)
      Padding String: 00000000000000000000000000000000
      [Direction: client-to-server]

```

All packets are encrypted

## Question 4

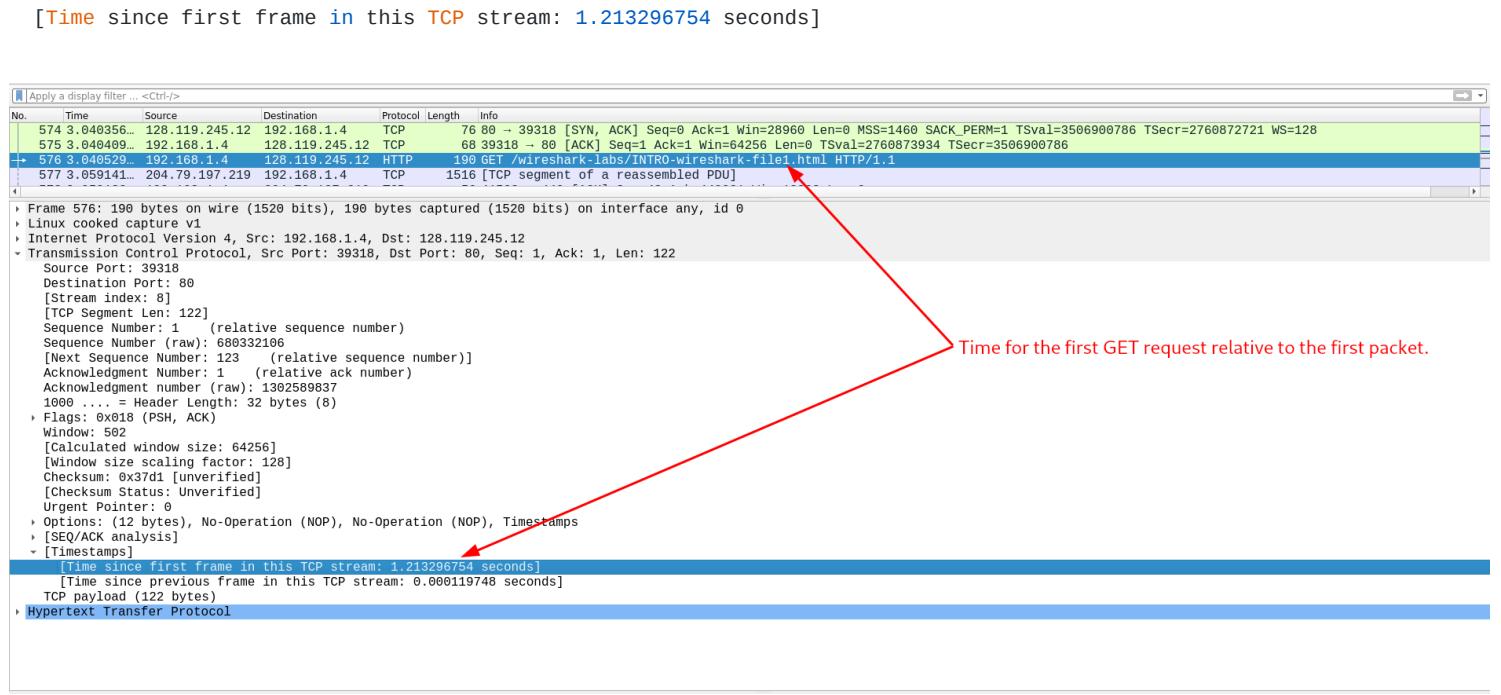
Enter the URL: <http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html> and capture packets using Wireshark. After your browser has displayed the INTRO-wireshark-file1.html page (it is a simple one line of congratulations), stop Wireshark packet capture. Answer the following from the packets captured:

- a. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received?
- b. What is the Internet address of the gaia.cs.umass.edu? What is the Internet address of your computer? Support your answer with an appropriate screenshot from your computer.

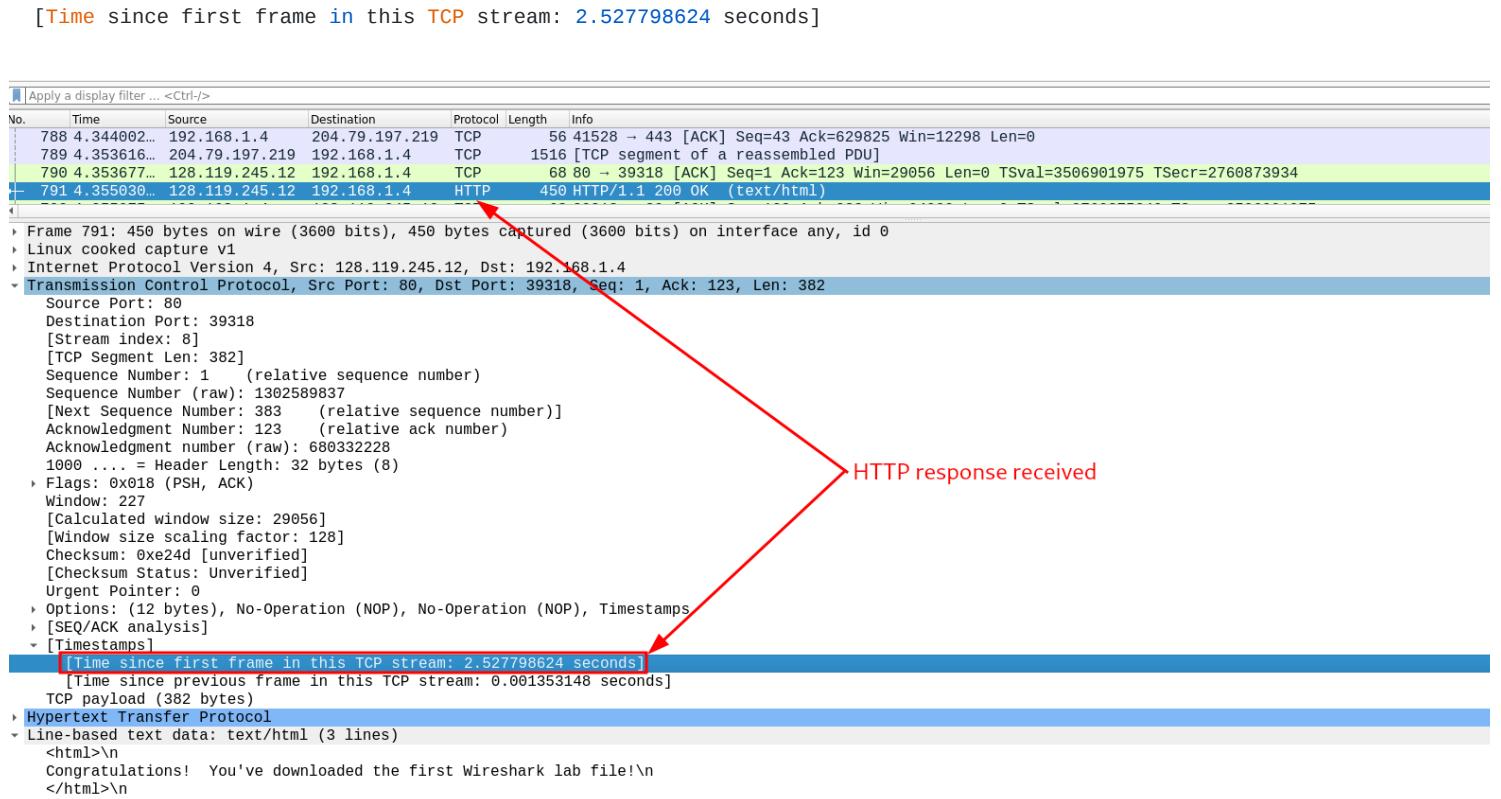
## Answer 4

## a. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received?

The first GET request was made in 1.213296754 s .



The response was received in 2.527798624 s .



So, time taken is 2.527798624 - 1.213296754 = 1.31450187 s .

**b. What is the Internet address of the gaia.cs.umass.edu? What is the Internet address of your computer? Support your answer with an appropriate screenshot from your computer**

- Internet address of the gaia.cs.umass.edu is Destination Address: 128.119.245.12
- Internet address of my computer is Source Address: 192.168.1.4

Wireshark screenshot showing network traffic. A red arrow points from the 'My machine' host (192.168.1.4) to the 'Source Address' field in the packet details. A blue arrow points from the 'Server' host (128.119.245.12) to the 'Destination Address' field. The packet list shows several TCP segments, with the highlighted row showing a GET request for '/wireshark-labs/INTRO-wireshark-file1.html'.

## Question 5

Start the Wireshark packet capturing service. Enter the URL: <https://www.gmail.com> on your browser and sign-in to your gmail account by providing credentials (Username/Password). Answer the following from the captured packets:

- a. Is there any difference in the application layer protocol?
- b. How it is different from the HTTP data you analysed in the above problem

## Answer 5

**a. Is there any difference in the application layer protocol?**

Yes, the current application layer protocol is **TLS**. A primary use case of TLS is encrypting the communication between web applications and servers, such as web browsers loading a website.

But, in the previous case it was just **HTTP**.

**b. How it is different from the HTTP data you analysed in the above problem?**

Previously, the protocol used was simple HTTP. As a result sniffers like wireshark could decode the data. Here is an example:

Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
785 4. 328527...	192.168.1.4	204.79.197.219	TCP	56	41528 - 443 [ACK] Seq=43 Ack=626905 Win=12298 Len=0	
786 4. 334149...	204.79.197.219	192.168.1.4	TCP	1516	[TCP segment of a reassembled PDU]	
787 4. 343959...	204.79.197.219	192.168.1.4	TCP	1516	[TCP segment of a reassembled PDU]	
788 4. 344002...	192.168.1.4	204.79.197.219	TCP	56	41528 - 443 [ACK] Seq=43 Ack=629825 Win=12298 Len=0	
789 4. 353616...	204.79.197.219	192.168.1.4	TCP	1516	[TCP segment of a reassembled PDU]	
790 4. 353677...	128.119.245.12	192.168.1.4	TCP	68	80 -> 39318 [ACK] Seq=1 Ack=123 Win=29056 Len=0 TSval=3506901975 TSecr=2760873934	
791 4. 355030...	128.119.245.12	192.168.1.4	HTTP	450	HTTP/1.1 200 OK (text/html)	

Status Code: 200 [Status Code Description: OK] Response Phrase: OK						
No.	Time	Source	Destination	Protocol	Length	Info
0000	00:00:00.01:00:06:7c:a9	6b:33:c3:d6:7d:83:08:00	..... .k3...}....			
0010	45:00:01:b2:e2:5b:40:00	25:06:3a:ba:80:77:f5:0c	E:...[@%:.;w..			
0020	c0:a0:01:04:00:50:99:96	4d:a3:f1:8d:28:8d:0b:c4	....P:M.....(....			
0030	80:18:00:e3:e2:40:00:00	01:01:08:0a:d1:07:13:d7	....M.....			
0040	a4:8f:97:ce:48:54:50:50	2f:31:21:31:20:32:30:30	....HTTP /1.1 200			
0050	20:4f:4b:0d:0a:44:61:74	65:3a:20:57:65:64:2c:20	OK - Dat e: Wed,			
0060	30:32:20:46:65:62:20:32	39:32:32:28:31:34:3a:35	02 Feb 2 022 14:5			
0070	35:34:32:36:20:47:4d:54	0d:0a:53:65:72:76:65:72	5:26 GMT ..Server			
0080	3a:20:41:70:61:63:68:65	2f:32:23:34:2e:36:20:28	: Apache /2.4.6 (			
0090	43:65:66:74:4f:53:29:20	4f:70:65:66:53:53:4c:2f	CentOS) OpenSSL/			
00a0	31:26:30:2e:32:6b:2d:66	69:70:73:28:58:48:50:2f	1.0.2k-f ips PHP/			
00b0	37:26:34:2e:32:37:20:6d	67:64:57:70:65:72:6c:2f	7.4.27 m od_perl/			
00c0	32:26:30:2e:31:31:26:50	65:72:6c:2f:76:35:26:31	2.0.11 P erl/v5.1			
00d0	36:23:03:0d:0a:4c:61:73	74:2d:4d:67:64:69:66:69	6.3 - Las t-Modifi			
00e0	65:64:3a:20:57:65:64:2c	20:30:32:28:46:65:62:20	ed: Wed, 02 Feb			
00f0	32:30:32:32:20:30:36:3a	35:39:38:30:31:20:47:4d	2022 06: 59:01 GM			
0100	54:00:04:45:54:61:67:3a	20:22:35:31:2d:35:64:37	T - ETag: "51-5d7			
0110	30:33:38:64:36:66:36:33	62:62:22:0d:0a:41:63:63	038d6f63 bb". Acc			
0120	65:70:74:42:52:61:6e:67	65:73:38:29:62:79:74:65	ept-Rang es: byte s. Conte nt-Lengl			
0130	73:00:04:43:6e:74:65:65	6e:74:2d:4c:65:66:67:74	h: 81: C ontent-T			
0140	68:34:28:38:31:0d:0a:43	6f:66:74:65:66:74:2d:54	Content-type: /html;			
0150	79:70:65:3a:20:74:65:78	74:2f:68:74:6d:6c:3b:20	charset: UTF-8 .			
0160	63:68:61:72:73:65:74:3d	55:54:46:2d:38:0d:0d:0d	>.			
0170	0a:3c:68:74:6d:6c:3e:0a	43:ef:66:67:72:61:74:75	<html> Congratul			
0180	6c:61:74:69:6f:6e:73:21	20:20:59:6f:75:27:76:65	ations! You've			
0190	20:64:67:77:6e:6c:61:61	54:85:64:20:74:68:65:20	downloaded the			
01a0	66:69:72:73:74:20:57:69	72:65:73:68:61:72:6b:20	first Wireshark			
01b0	6c:61:62:20:66:69:6c:65	21:0a:3c:2f:68:74:6d:6c	lab file !.</html>			
01c0	3e:0a					

Clearly, we can see the message in plain text.

But, with `gmail.com` the connection is TLS protected. So, the data is encrypted.

ip.addr==192.168.1.4 && tcp.port==56340 && ip.addr==142.250.206.133 && tcp.port==443						
No.	Time	Source	Destination	Protocol	Length	Info
117 5. 783150...	192.168.1.4	142.250.206.133	TLSv...	392	Application Data	
118 5. 783542...	192.168.1.4	142.250.206.133	TLSv...	107	Application Data	
131 5. 850341...	142.250.206.133	192.168.1.4	TLSv...	107	Application Data	
132 5. 850414...	192.168.1.4	142.250.206.133	TCP	68	56340 - 443 [ACK] Seq=364 Ack=40 Win=501 Len=0 TSval=3513555587 TSecr=842035744	
133 5. 857183...	142.250.206.133	192.168.1.4	TCP	68	443 - 56340 [ACK] Seq=1 Ack=325 Win=269 Len=0 TSval=842035744 TSecr=3513555520	
134 5. 857224...	192.168.1.4	142.250.206.133	TCP	68	[TCP Dup ACK 132#1] 56340 - 443 [ACK] Seq=364 Ack=40 Win=501 Len=0 TSval=3513555594 TSecr=842035744	
135 5. 857183...	142.250.206.133	192.168.1.4	TCP	68	443 - 56340 [ACK] Seq=1 Ack=364 Win=269 Len=0 TSval=842035744 TSecr=3513555520	
181 6. 298905...	142.250.206.133	192.168.1.4	TLSv...	750	Application Data	
182 6. 298924...	192.168.1.4	142.250.206.133	TCP	68	56340 - 443 [ACK] Seq=364 Ack=722 Win=496 Len=0 TSval=3513556036 TSecr=842036202	
183 6. 300229...	142.250.206.133	192.168.1.4	TLSv...	99	Application Data	
184 6. 300241...	192.168.1.4	142.250.206.133	TCP	68	56340 - 443 [ACK] Seq=364 Ack=753 Win=501 Len=0 TSval=3513556037 TSecr=842036204	
185 6. 300229...	142.250.206.133	192.168.1.4	TLSv...	107	Application Data	
186 6. 300256...	192.168.1.4	142.250.206.133	TCP	68	56340 - 443 [ACK] Seq=364 Ack=792 Win=501 Len=0 TSval=3513556037 TSecr=842036204	
187 6. 300527...	192.168.1.4	142.250.206.133	TLSv...	107	Application Data	
192 6. 360246...	142.250.206.133	192.168.1.4	TCP	68	443 - 56340 [ACK] Seq=792 Ack=403 Win=269 Len=0 TSval=842036266 TSecr=3513556037	

> Frame 181: 750 bytes on wire (6000 bits), 750 bytes captured (6000 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 142.250.206.133, Dst: 192.168.1.4
Transmission Control Protocol, Src Port: 443, Dst Port: 56340, Seq: 40, Ack: 364, Len: 682
Transport Layer Security
TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
Content Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 677
Encrypted Application Data: df25d96d1177a1cc96a715eea3247b8644bdb5525d57e171ad5ee03a5fb8770c836ba133...
[Application Data Protocol: http-over-tls]

0040 d1:6c:9a:8a:17:03:03:02:a5:df:25:d9:6d:11:77:a1	1.....-.%-m-w-.....\${-D-R]W.
0050 cc:96:a7:15:ee:a3:24:7b:86:44:bd:b5:52:d5:57:e1	q^A:_-w ..k-3Ec.
0060 71:ad:5e:0a:3a:5b:b8:77:0c:83:6b:a1:33:45:63:d9	>2..1..T2...5y.
0070 02:00:3e:32:18:b8:21:85:32:54:1f:a1:9e:35:79:45	.....j ,.I?m-X.
0080 c0:95:ac:f1:b6:6a:c4:bc:2c:f9:49:3f:6d:ac:58:c1	UFV.....#<.....
0090 ea:85:55:66:5c:a1:df:95:9c:ed:23:3c:8c:d9:9b:69	?-q...g ..-}%.3-Y....
00a0 3f:f0:b0:71:aa:d9:14:67:ba:a8:3d:dd:7d:08:25:b1	..O-M.....-
00b0 d9:33:f3:59:1a:aa:13:05:fd:ab:20:e4:4f:83:4d:85	..1%/-= ..??.....
00c0 cf:11:69:25:17:3d:e9:dc:3f:3f:86:9b:ee:d9:13	E-r:4D/S q..R..\\x..lf..Y7:HUe.
00d0 45:bc:72:88:34:44:2f:53:71:b3:b6:52:df:99:5f:5c	..?..2...Y-1]T.5.
00e0 cd:c0:78:10:c5:6c:66:9a:d9:59:37:19:48:55:65:6f	..M..h..I..E..F..
00f0 f7:f1:a8:9a:3f:d7:2e:c4:59:c6:5d:54:02:35:b7	..#..]v..KA.i..\\..Z..#G v."..a..
0100 ad:94:22:4d:ad:86:68:93:f2:2e:a9:t6:49:08:9e:1c	
0110 06:f5:45:fd:f5:cc:46:87:f6:db:cd:9d:fa:ce:cd	
0120 97:f2:23:ed:8d:5d:76:aa:4b:41:05:69:00:96:2d:1a	
0130 5c:88:4f:5a:eb:11:23:47:76:16:22:0e:08:61:dd:a1	

here, we cannot decode the text because it is encrypted. This way anyone sniffing the network cannot find the Username and Password used.